Electric Drives and Controls | Hydraulics | Linear Motion and Assembly Technologies | Pneumatics | Service

Rexroth Bosch Group

Rexroth VisualMotion 11 Multi-Axis Machine Control Volume 1

R911309162
Edition 01

Functional Description

Title Rexroth VisualMotion 11
Multi-Axis Machine Control
Volume 1

Type of Documentation Functional Description

Document Typecode DOK-VISMOT-VM*-11VRS**-FK01-EN-P

Internal File Reference Document Number, 120-2300-B326-01/EN
Part of Box Set, 20-11V-EN (MN R911309160)

Purpose of Documentation This documentation describes the functionality supported in Rexroth VisualMotion 11 using G*P 11 firmware.

Record of Revisions

Description	Release Date	Notes
DOK-VISMOT-VM*-11VRS**-FK01-EN-P	4/2006	Official release

Copyright © 2005 Bosch Rexroth AG

Copying this document, giving it to others and the use or communication of the contents thereof without express authority, are forbidden. Offenders are liable for the payment of damages. All rights are reserved in the event of the grant of a patent or the registration of a utility model or design (DIN 34-1).

Validity The specified data is for product description purposes only and may not be deemed to be guaranteed unless expressly confirmed in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.

Published by Bosch Rexroth AG
Bgm.-Dr.-Nebel-Str. 2 • D-97816 Lohr a. Main
Tel.: +49 (0)93 52/40-0 • Fax: +49 (0)93 52/40-48 85 • Telex: 68 94 21

Bosch Rexroth Corporation • Electric Drives and Controls
5150 Prairie Stone Parkway • Hoffman Estates, IL 60192 • USA
Tel.: 847-645-3600 • Fax: 847-645-6201

http://www.boschrexroth.com/

Dept. ESG4 (DPJ)

Note This document has been printed on chlorine-free bleached paper.

Table of Contents

1 About this Manual

1.1 Introduction

The VisualMotion 11 Functional Description manual describes the VisualMotion software, including the integration of IndraLogic, the tools used for programming both motion and logic, and the system's functionality.

The Functional Description manual is a two volume set. The two books are structured as follows:

- Volume 1 contains information on how to use VisualMotion Toolkit, IndraLogic and all the supporting tools available for programming both motion and logic.
- Volume 2 is structured as a reference guide which presents detailed functionality of the features supporting VisualMotion and IndraLogic.

1.2 Contents of Volume 1

The following table provides a description for each chapter in volume 1:

Chapter	Description
1	**About this Manual -** describes the structure of the two volume functional description set.
2	**VisualMotion Software Installation** - describes the system requirements for installing and operating VisualMotion 11 on your PC. It also describes the installation of VisualMotion, IndraLogic and the CAM Builder softwares. Initial communication between the software and your system, including the control, drives and I/O is also described.
3	**Overview of VisualMotion System** - describes a VisualMotion system, what it includes and how it operates, such as operating modes
4	**VisualMotion Toolkit** - describes the VisualMotion Toolkit environment, the programming modes as well as basic steps required for creating a VisualMotion motion project, the steps for importing a default IO mapping IndraLogic project and the use of the IO Box utility for activating the project.
5	**Programming Capabilities -** describes the basic motion programming capabilities for Events, Position Monitoring Groups, PID, and PLS.
6	**Operating Modes** - describes the basic operating modes available in VisualMotion
7	**Commissioning Tools -** describes the functionality of the commissioning tools in VisualMotion including how to configure, download and use them. IndraLogic's integration into VisualMotion is described in this chapter.
8	**Data Editing and Monitoring Tools –** describes how to use data editing and monitoring tools with a VisualMotion project.
9	**Diagnostic Tools -** describes tools used for diagnosing and debugging a VisualMotion project
10	**Project Management Tools -** describes the tools used to manage a VisualMotion project
11	**Error Reaction -** describes how VisualMotion reacts to errors

Table 1-1: Volume 1 Chapter

2 VisualMotion Software Installation

2.1 Overview

The VisualMotion software package includes the installation of the following editors and tools:

- VisualMotion Toolkit motion programming software and tools
- IndraLogic PLC logic and motion programming software
- New CamBuilder tool
- DriveTop software for commissioning drives

The editors and tools installed as part of the VisualMotion software package support both English and German. Separate help systems and/or documentation for VisualMotion, IndraLogic and the CamBuilder are available as part of the installation.

Note: It is not recommended to have a stand alone version of DriveTop installed on the same PC in addition to the integrated version installed with VisualMotion 11.

2.2 System Requirements

The following system specifications are recommended for running VisualMotion 11 software.

Computer

VisualMotion Toolkit can be installed on any IBM™ PC compatible Pentium computer with the following minimum requirements:

- 500 MHz Pentium 2 Processor
- 128 MB of RAM system memory
- Windows 2000 (SP4) or Windows XP (SP1)
- 95 MB of hard drive space on drive C and 165 MB on the target drive
- MS-IE (Microsoft Internet Explorer) 5.1 or later

Display

A super VGA monitor or better. A color monitor display makes it possible to take full advantage of VisualMotion's graphic interface.

Printer

VMT uses the default printer installed on your computer. For optimal resolution, especially when printing projects, use a high-resolution (300-dpi) laser or ink jet printer.

Mouse

A mouse interface is required to use VisualMotion Toolkit, IndraLogic, and supported tools.

Serial I/O

VMT can be configured to use the PC's serial port for communication between the host PC and the PPC-R. An IKB0005 RS-232 serial cable is required between the host PC and the PPC-R X10 or X16 communication ports. Hardware handshaking is not supported.

2.3 Installing the VisualMotion Software

VisualMotion Toolkit 11 is the initial software installation. Use the following steps to begin the installation of the VisualMotion software package:

1. Insert the VisualMotion CD into the CD-ROM drive. VisualMotion will start automatically.
2. Select the setup language.

The language option can be changed at any time after installation by selecting ***Tools ⇒ Options*** from VisualMotion Toolkit's main menu.

Note: If the language selected in VisualMotion does not match the language of the computer's operating system, some windows in VisualMotion will maintain the operating system's language.

ChooseLang.tif

Fig. 2-1: Choose Setup Language

Note: When installing VisualMotion Toolkit in English, only the English resource Dynamically Linked Libraries (DLL) are installed. The German language will not be available with an English only installation. To install the German resource DLLs, repeat the installation process using German as the setup language. However, English resource DLLs are supported with a German installation.

The VisualMotion installation launches an InstallShield Wizard that guides the user through the installation process.

Overwrite an Existing Release

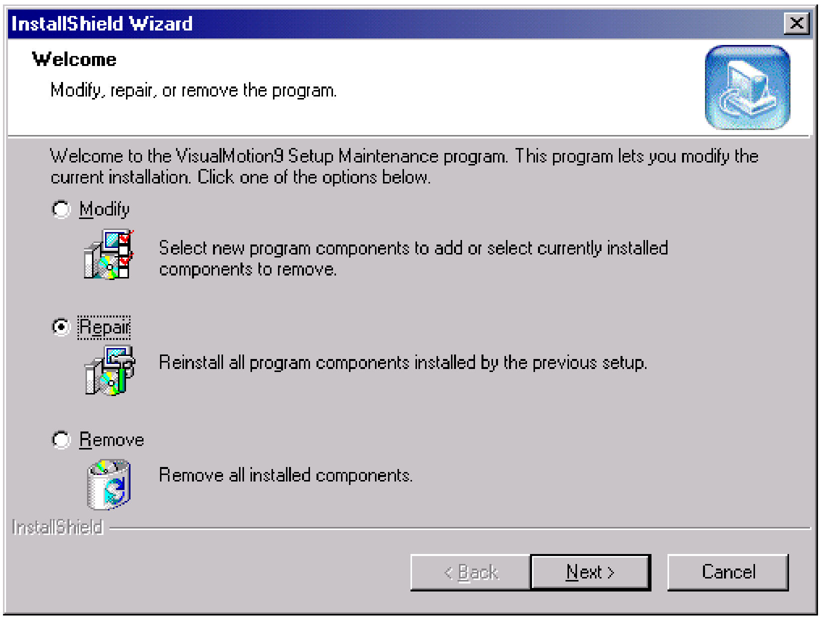
If a previous release of VisualMotion Toolkit, IndraLogic or the CamBuilder are found on the PC, the setup program will find the software and launch a *Modify, Repair, and Remove* wizard.

From this wizard, the following options are available:

Option	Description
Modify	This option allows new program components to be added or installed components can be removed.
Repair	This option overwrites existing VisualMotion components with the files in the new release. This will not affect the programs that have been created with an earlier release. Programs saved on the computer's hard drive can be downloaded to the new release of VisualMotion.
Remove	This option removes all installed components

Table 2-1: Options for Overwriting a VisualMotion Installation

Note: The *Modify, Repair, and Remove* wizard opens if previous installations of VisualMotion Toolkit, CamBuilder, and/or IndraLogic are found on the PC.



Install_Repair.tif

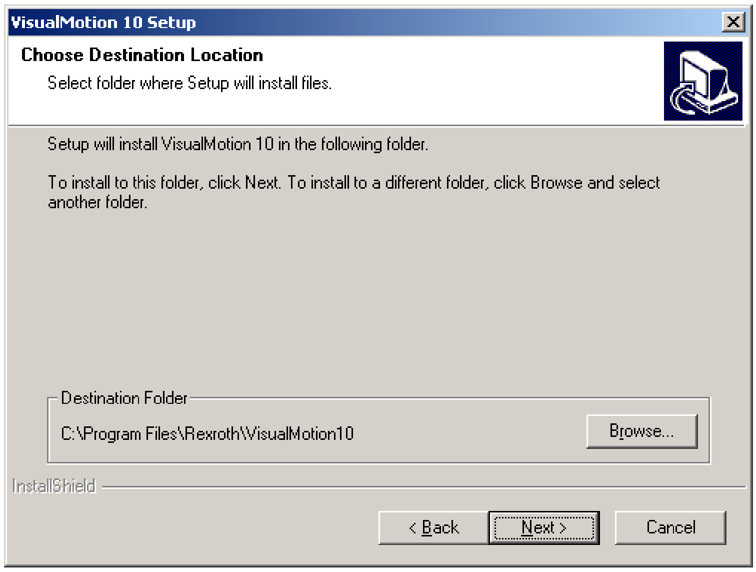
Fig. 2-2: Modify, Repair, Remove VisualMotion Components

3. Click ***Next>*** in the *Welcome* window.
4. Click ***Yes*** in the license window to continue with the installation.
5. Enter a user name, company name, and serial number in the *Customer Information* window.

Note: The serial number is printed on the software packaging material.

6. Accept the default *Destination Folder* by clicking the ***Next*** button, or change the default location by clicking the ***Browse…*** button.

Rexroth Bosch Group



Install_Destination.tif

Fig. 2-3: Installation Destination Location

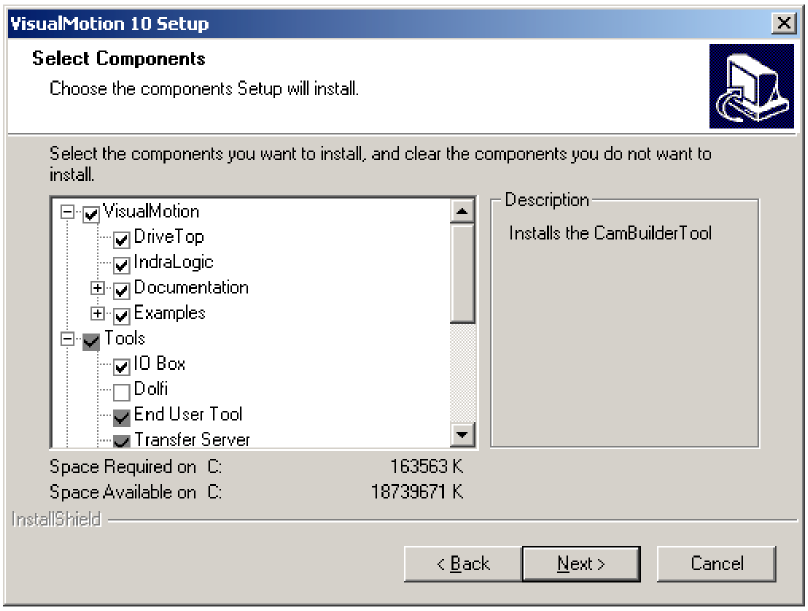
7. From the *Select Components* window, select the VisualMotion components to install and click the ***Next*** button. The following table lists the available components available for installation:

Main Component	Sub-Item	Default	Optional	Description
VisualMotion	Icon Editor	X		VisualMotion Icon Editor, compiler & commissioning tools
	DriveTop	X		Drive commissioning (integrated version)
	IndraLogic	X		Logic Editor Tool (integrated)
	Documentation	X	X	VisualMotion manuals and help system documentation
	Examples	X	X	VisualMotion sample projects, PLC function blocks and HMI OPC Client
Tools	I/O Box	X		VisualMotion IO-Box utility
	Dolfi		X	Software application for downloading firmware to Rexroth controls and drives
	End User Tool	X		VisualMotion end user tool for general status, parameter editing and system archiving
	Transfer Server	X		VisualMotion Dde transfer server for third-party interfaces
	CamBuilder	X		New CamBuilder tool **Refer to CamBuilder note below:**
DriveHelp	DIAX03		X	For SSE03 and ELS05 firmware
	DIAX04		X	For SSE03 and ELS05 firmware
	Ecodr3		X	For SGP01, FGP03, SGP20 and SMT02 firmware
	EcodrCS		X	For MGP01 firmware
	IndraDrive		X	For MPx02 and MPx03 firmwares

Table 2-2: Select Components

CamBuilder Note:

If the new CamBuilder tool is unchecked during the initial installation, the CamBuilder Utility used with VisualMotion 9 will be installed instead. If the new CamBuilder tool is installed, the older CamBuilder Utility can be added to the ***Registered Tools*** menu selection under the ***Tools*** menu. Refer to the *Tools* menu description in section 13.10 in volume 2 of the *VisualMotion 11 Functional Description* manual.



Install_Setup_Type.tif

Fig. 2-4: Installation Setup Type

IndraLogic Installation

8. Select the installation language for IndraLogic
9. An InstallShield wizard will guide the user through the IndraLogic installation.
10. Click the ***Next*** button to begin the IndraLogic installation.

Note: A Microsoft XML Parser must be installed as part of the IndraLogic installation.

CamBuilder Installation

11. If the new CamBuilder tool is selected for installation, select the setup language.
12. An InstallShield wizard will guide the user through the CamBuilder tool installation.
13. Click the ***Next*** button to begin the CamBuilder installation.

Optional Software Installation

14. In addition to the default VisualMotion component installations, selected optional software such as Dolfi and SyCon will follow the IndraLogic installation.
15. Follow the InstallShield instruction until all optional components are installed.
16. After the InstallShield wizard completes, it is recommended that the computer be restarted.

Rexroth Bosch Group

2.4 Establish Communication using VisualMotion Toolkit

VisualMotion 11 supports the VisualMotion Dde server and the Scalable Communication Platform (SCP) for communication with the control.

The VisualMotion Dde server is available for GPP/GMP 11 and earlier firmwares for communication with the control's motion functionality (VisualMotion Toolkit to Control).

The SCP server is available for GPP/GMP 09, 10, and 11 firmware versions and can be used for communication with the control's motion functionality but is required for communication with the control's PLC logic functionality (IndraLogic to Control).

Before establishing communication, determine the communication server to be used and the connection interface to the control.

Note: Control's with a new memory card (PFM), containing no previously downloaded programs, requires a serial connection to establish initial communication.

The following table lists the connection type available by control type:

Control	Connection	Details
PPC-R22.1	Serial (RS232)	Use serial cable IKB0005
	Network	Optional onboard EtherNet
PPC-P11.1	PCI Bus	With PPC-P11.1 installed in host PC
	Serial (RS232)	Use serial cable IKB0005

Table 2-3: Connection Types by Control

SCP Server Configuration

Use the following procedure to configure the SCP server and then establish communication:

1. Open VisualMotion Toolkit in service mode and select ***Tools ⇒ Control Selection***.
2. From the Dde Server section, select SCP (SIS Protocol) and click on the ***Configure…*** button.

Note: Refer to section 18.3, *Configuring the SCP Server,* in volume 2 of the *VisualMotion 11 Functional Description* manual.

3. Once the server is configured, select ***Diagnostics ⇒ System*** from the VisualMotion Toolkit menu.

If proper communication has been established, the Status tab in the Systems Diagnostics window will display the current operating state.

VisualMotion Dde Server Configuration

Use the following procedure to configure the VisualMotion Dde server and then establish communication:

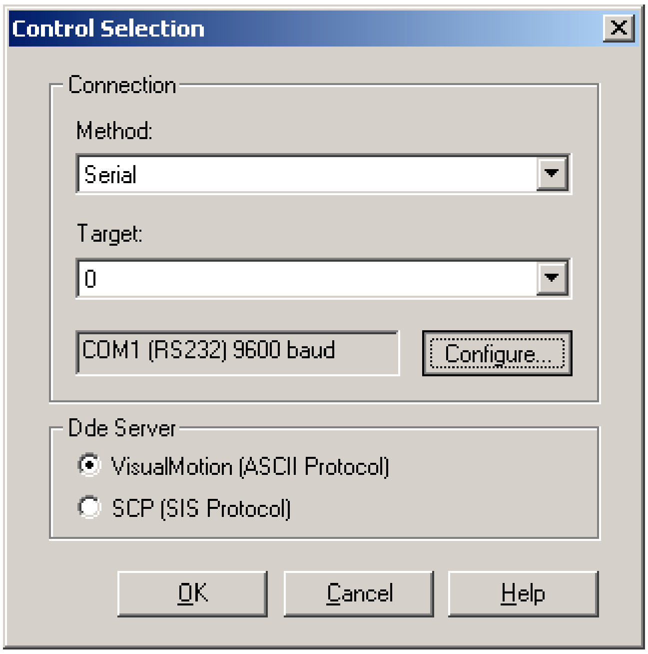
1. Open VisualMotion Toolkit in service mode and select ***Tools ⇒ Control Selection***.
2. From the *Dde Server* section, select ***VisualMotion (ASCII Protocol)***.

3. Select ***Serial*** for the connection method.

Note: Select ***PCI*** for the connection method when establishing communication with a PPC-P11.1 control over the PCI Bus.

---

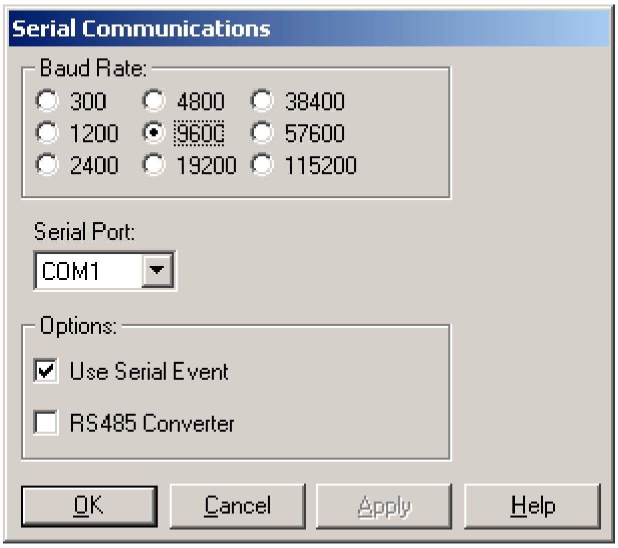
4. Use the default 0 value for the *Target*.



DdeConfigure.tif

Fig. 2-5: Control Selection Window

5. Click the ***Configure…*** button to open the *Serial Communication* window.



Serial_Comm.tif

Fig. 2-6: Serial Communication

6. Select the baud rate setting in the *Serial Communications* window that matches the baud rate setting in the control.

Note: To view the current baud rate settings in the control, press the S1 button on the front of the control 4 times. To display the baud rate of the X16 connector, depress the S1 button 6 times.

7. Once the server is configured, select ***Diagnostics*** ⇒ ***System*** from the VisualMotion Toolkit menu.

If proper communication has been established, the *Status* tab in the Systems Diagnostics window will display the current operating state.

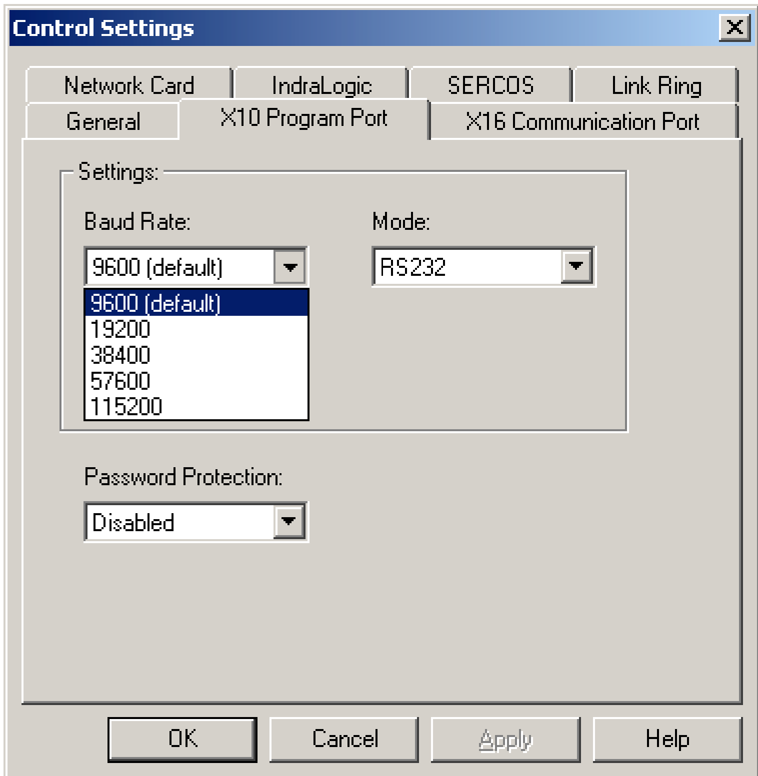
Changing the Baud Rate

After communication has been established between VisualMotion and the control, the baud rate setting can be changed. To change the baud rate:

1. Open the Control Settings window by selecting ***Tools*** ⇒ ***Control Settings…***
2. Select either the X10 Program Port tab or X16 Communication Port tab.

Note: These tabs can only be viewed when VisualMotion is in online or service mode.

3. Select the new baud rate setting from the drop-down menu.



Change_Baud_Rate.tif

Fig. 2-7: Baud Rate Selection

4. Cycle power to the control for the change to take affect.
5. Select ***Tools*** ⇒ ***Control Settings…*** to open the *Control Settings* window in VisualMotion.

Note: A Dde server error may be issued, indicating the baud rate on the control no longer matches the setting in the Dde Server.

6. Click the ***Settings*** button in the error message window. This will open the *Serial Communications* window.
7. Change the baud rate to match the new baud rate in the control.

Optional Ethernet Onboard Interface

The optional Ethernet onboard interface on the PPC-R22.1 control contains it's own TCP/IP (***T****ransmission* ***C****ontrol* ***P****rotocol/****I****nternet* ***P****rotocol*) stack. The TCP/IP stack enables the Ethernet interface to transmit data over the network or Internet and communicate with VisualMotion Toolkit via the SCP or VisualMotion Dde Server.

Ethernet Interface Setup

Before an Ethernet interface can be setup for the PPC-R22.1, initial communication with the control, via a serial connection, must be established. Also, a unique network IP address must be reserved for the control. Contact your local Information Technology (IT) department for details.

EtherNet Setup via Control Parameters

The following steps are used to configure an Ethernet interface via a serial connection (IKB0005) using VisualMotion Toolkit by directly accessing the control parameters.

1. Power up the control and launch VisualMotion Toolkit in service mode.
2. Select ***Data*** ⇒ ***Parameters*** to open the *Parameter Overview tool*.
3. Modify control parameter C-0-0400 and enter the EtherNet's IP Address in dot notation, for example "172.18.11.205".
4. Modify control parameter C-0-0401 and enter the EtherNet's Subnet Mask in dot notation, for example "255.255.0.0".
5. Modify control parameter C-0-0402 and enter the EtherNet's Gateway IP Address in dot notation, for example "172.16.1.1".

Note: The optional onboard EtherNet interface on the PPC-R22.1 supports auto-negotiation and does not require setting the duplexing mode in control parameter C-0-0403.

6. Close VisualMotion Toolkit and cycle power to the control in order for these changes to take affect.

After the control is powered up, the green H3 LEDs on the front of the control will flash continuously.

EtherNet Setup using VisualMotion Tool

The Ethernet interface can also be configured using the *Control Setting* window under the *Tools* menu selection.

1. Power up the control and launch VisualMotion Toolkit in service mode.
2. Select ***Tools*** ⇒ ***Control Settings…*** and select the **Network** tab.
3. Enter the EtherNet's IP Address, Subnet Mask, and Gateway IP Address in dot notation, for example "172.18.11.205".
4. Close VisualMotion Toolkit and cycle power to the control in order for these changes to take affect.
5. After the control is powered up, the green H3 LEDs on the front of the control will flash continuously.

PCI Communication

The GMP 11 firmware for the PPC-P11.1 is designed to work in a complete system solution consisting of a Logic Controller (Soft PLC) and Motion Controller (PPC-PCI) inside an industrial PC with an HMI package.

On the PC, there are three main interfaces with the PPC-P11.1: a soft PLC, VisualMotion, IndraLogic and HMI packages such as WinHMI or WonderWare. The soft PLC will have direct access to the DPR (dual port RAM), while VisualMotion, IndraLogic and the HMI packages will communicate over the DPR via the Scalable Communication Platform (SCP).

GMP firmware supports all the functionality of the GPP firmware with the following exceptions:

- Data Mapper is not supported, thus eliminating the option of ordering the corresponding PC104 interface boards with this system. The parent soft control (softPLC) communicates directly to the PPC-P11.1 via the PCI bus. If additional fieldbus connectivity is required other than Profibus and DeviceNet masters, the soft control (softPLC) should be equipped with the capability to communicate with a PC-resident fieldbus card. The Data Mapper software utility is used to set up PCI cyclic channels for the softPLC. PC104 fieldbus slave cards are not supported.
- The Register and Cyclic channels over the Dual Port RAM use the Sercos cycle time.
- No LocalReco configurations are supported.

3 VisualMotion 11 Overview

3.1 System Overview

The VisualMotion 11 system has a combination of motion and logic programming that provides enhanced speed and versatility for the interface of PLC tasks and real-time motion. This motion/logic combination is provided by VisualMotion 11 programmable multi-axis motion software and IndraLogic PLC programming software. Both software packages reside on the PC and communicate through the SCP (Scalable Communication Platform) or the VisualMotion Dde server (ASCII Protocol) over a serial, EtherNet or PCI connection to the PPC.

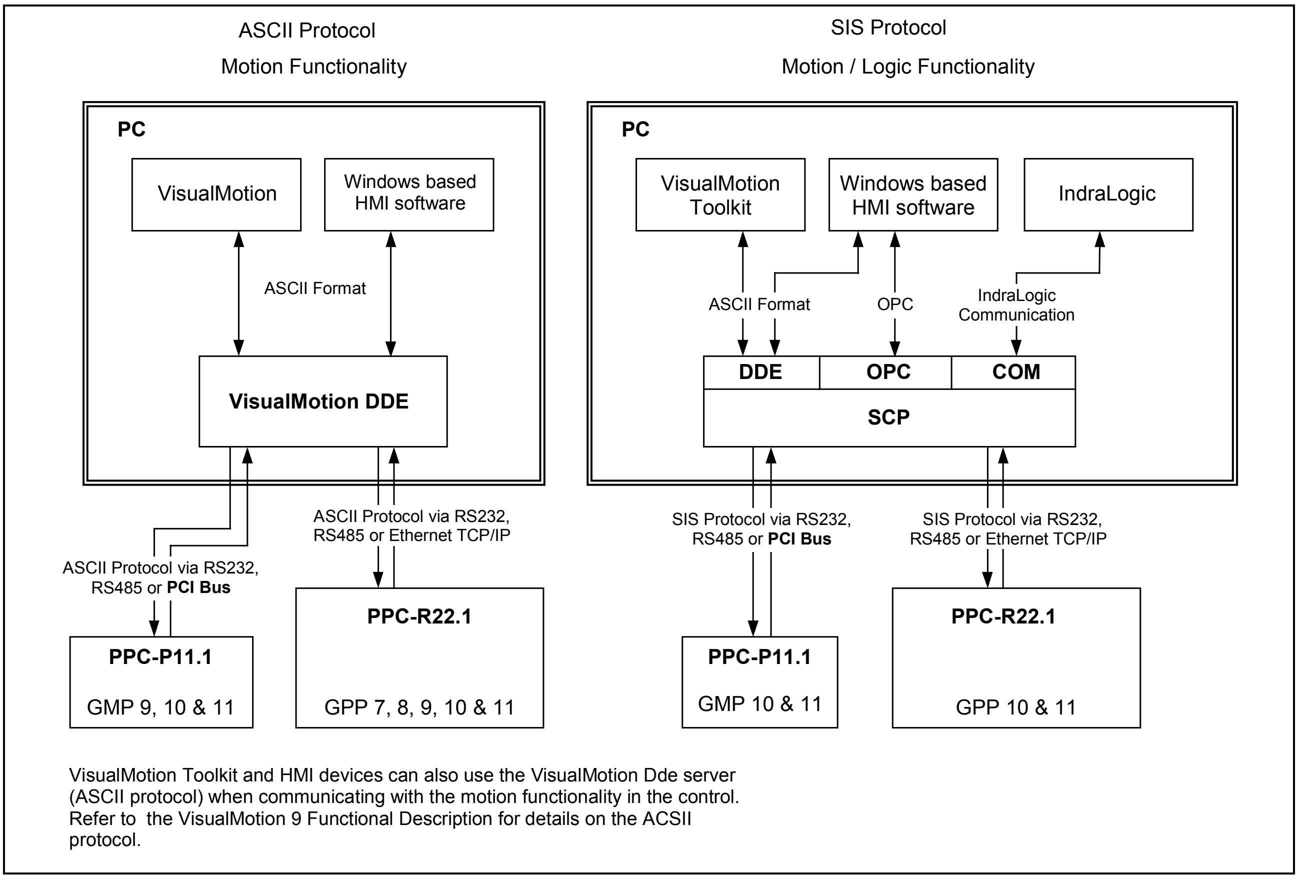


Fig. 3-1: Motion / Logic Overview

Rexroth VisualMotion 11 supports the following hardware form factors and firmware versions:

- PPC-R22 using GPP 11 firmware (Reco-version)
- PPC-P11.1 using GMP 11 firmware (PCI-version)

3.2 Motion Types

Rexroth VisualMotion supports three types of motion:

- Non-Coordinated
- Coordinated
- Electronic Line Shafting

Rexroth
Bosch Group

Non-Coordinated Motion

Non-coordinated motion is primarily used to control a single independent axis. There are two modes of non-coordinated motion:

- Single axis
- Velocity mode

Single axis

Single axis motion commands within a Rexroth VisualMotion project are processed by the control and sent to the digital drive. The project communicates to the drive the target position (travel distance), the velocity and acceleration. This information is used to develop a velocity profile that is maintained and controlled within the intelligent digital drive. As a result, single axis motion does not require continuous calculation by the control and consumes minimum CPU resources.

Velocity Mode

Velocity mode controls the speed of the axis, with no position control loop. Rexroth digital drives maintain torque and velocity loops internally.

A special form of non-coordinated motion called *ratioed axes* permits linking two axes by relating the number of revolutions of a slave axis to a master axis. For example, a ratio might be required when the positioning axis of a gantry robot has a motor on each side of its supporting track.

Coordinated Motion

The Rexroth VisualMotion control defines the following two types of coordinated multi-axis coordinated motion:

- Standard Coordinated Motion
- Coordinated Articulation

Note: Refer to section 6.3, *Coordinated Motion*, for details.

Standard Coordinated Motion

Standard coordinated motion defines motion in terms of a path composed of standard straight line and circular geometry segments. Point positions, (x, y, z), are used to establish the start, middle or end of a geometry segment. Two points define a line; three points define a circle. The path combines these standard geometry segments so that the start of the next segment begins at the end of the previous segment. A path, therefore, is nothing more than a collection of connected segments.

Since each segment has an end point specifying speed, acceleration, deceleration and jerk, each segment can have a unique rate profile curve. A special type of segment, called a blend segment, can be used to join two standard geometry segments. Blend segments provide the capability of continuous smooth motion from one standard segment to another without stopping. They reduce calculation cycle time as well as provide a means of optimal path shaping.

A Rexroth VisualMotion system is capable of calculating a path in any of several different modes:

Constant Speed

Constant Speed mode is always active and tries to maintain a constant speed between any two connecting segments in the path. The system and axes acceleration and deceleration limit this mode. Constant speed is the optimum path motion for applying adhesives or paint, welding and some forms of cutting such as laser or water-jet, etc..

Linear Interpolation

Two points define a coordinated motion straight-line segment. The motion is calculated from the end point of the last segment, or the current position if the system is not in motion, to the new end point.

Circular Interpolation

Three points define a coordinated motion circular segment. Circular motion begins with the end point of the last segment executed, or the current system position if the system is not in motion, moves in a circular arc through an intermediate point, and terminates at the specified endpoint.

Kinematics

In addition to the standard x, y, z kinematics, the control has the capability of executing several forward and inverse kinematic movements by using an application-specific library of kinematic functions.

Kinematics can be developed to customer specifications. Contact Bosch Rexroth to inquire about applications which could benefit from kinematics.

Coordinated Articulation

Coordinated Articulation is an advanced feature in VisualMotion 11 used to move up to six axes in coordinated fashion based on world coordinate inputs from an ELS Group CAM output or manual positions. This feature provides the ability to link cyclic coordinated motion to an ELS master.

Differences from Normal Coordinated Motion

The following outlines the difference in features in Coordinated Articulation from standard Coordinated Motion:

- Points table are not supported
- Motion types supported: Coordinated and ELS Group
- Zones are not supported
- Up to 6 axis supported
- Up to 4 Coordinated Articulation configurations supported (only one configuration allowed per any task A-D)
- No BTC06 support
- Control task register supported
- Coordinated limit parameters supported

Electronic Line Shafting (ELS)

Electronic Line Shafting is used to synchronize one or more slave axes to a master axis. GPP firmware introduces multiple master functionality. Each slave axis can use either velocity, phase or CAM synchronization. ELS has the capability to jog each axis synchronously or independently, and to adjust phase offset and velocity while the project is running.

Note: Refer to section 6.1, *Electronic Line Shafting*, for details.

Velocity synchronization

Velocity synchronization relates slave axes to a master in terms of rotational rate. It is used when axis velocities are most critical, as in paper processing operations in which two or more motors act on a single piece of fragile material.

Phase synchronization

Phase synchronization maintains the same relative position among axes, but adjusts the lead or lag of the slaves to the master in terms of degrees. It is used when the positions of axes are most critical. For example, to achieve proper registration in printing operations, the axis controlling the print head may be programmed for a particular phase offset relative to some locating device, such as a proximity switch.

CAM synchronization

CAM synchronization is used when custom position profiles are needed at a slave axis. A CAM profile can be executed either in the control (control CAM) or in the drives (drive CAMs). The number of control CAMs that can be active at the same time is limited to 4.

A CAM is an (x, y) table of positions that relate a master axis to a slave. CAMs can be stored in the control or in the digital drive. Control CAMs have more adjustment options and can work with Sercos drives that do not support the ELS functionality (e.g., SMT or SSE firmware). The same programming commands and utilities are used for both control and drive CAMs.

4 VisualMotion Programming

VisualMotion 11 provides a complete motion/logic programming solution using both icon-based and PLC-based programming. The icon-based VisualMotion Toolkit programming environment provides the user with an easy to use pick-and-place method of programming that visually shows the program's logical flow using icon instructions. Integrated into VisualMotion Toolkit is an IEC 61131-3 PLC programming tool known as IndraLogic. Refer to section 7.1, *IndraLogic*, for complete details.

Together, these tools allow the user to select their preferred method of programming. However, in order to maintain a synchronized project between the motion side and the logic side, minimal programming is required using both tools. Icon-based programming requires that the I/O communication between motion and logic be programmed in IndraLogic. PLC-based programming requires that all axes, processes (e.g., ELS), and parameter initialization used in the system be programmed in VisualMotion Toolkit. The following diagram illustrates the basic steps required for programming in a VisualMotion system.

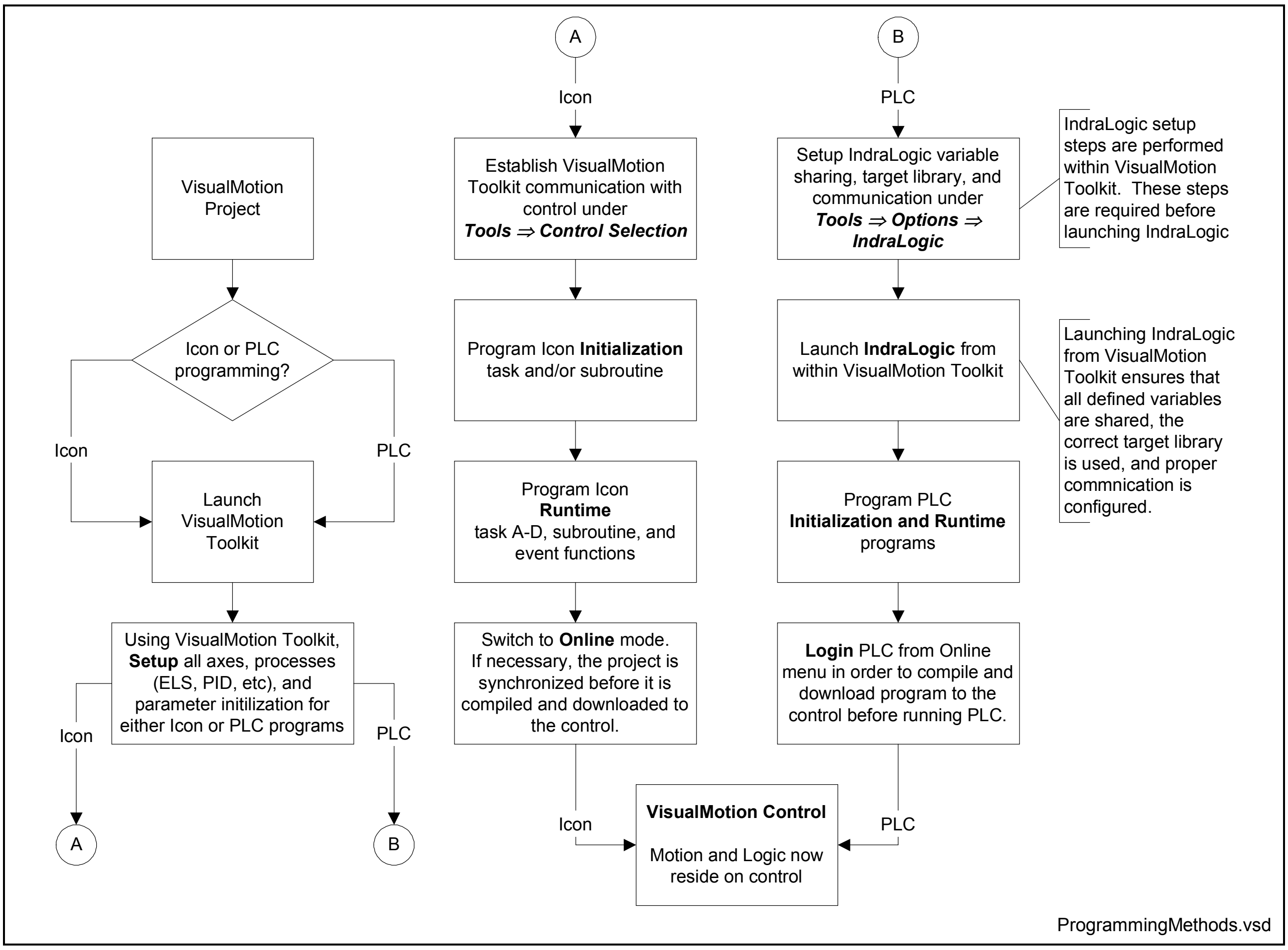


Fig. 4-1: Programming Methods

Rexroth
Bosch Group

4.1 VisualMotion Toolkit Programming Environment

VisualMotion Toolkit is structured to provide ease of navigation and programming. All menu selections, icons, and windows are arranged to make project navigation easier to follow. The VisualMotion Toolkit programming environment is structured as follows:

Area	Description
Main Menus	Pull down menu commands. Refer to chapter 13, *Menu Descriptions*, in volume 2 for details.
Toolbar Icons	Commonly used Windows commands and programming icons. Refer to chapter 14, *Icon Descriptions*, in volume 2 for details.
Project Navigator	The project navigator displays a tree structure for the project folders. The Setup folder is used to setup axes, processes, and parameter initialization. Selecting a folder displays the icon program related to that folder. When a subroutine or event is added to the project, they are indicated as subfolders in the subroutine and event folders.
Icon Palette	Grouping of programming icons based on functionality.
Icon Workspace	Area where icons are placed, configured and connected in a logical flow.
Comment Window	A text window where the user can add special instructions or details about the task.
Status Bar	Visual indications about program progress and programming mode.
Operating State Indicator	Graphical indicator showing the current mode of operation with the control and the project.

Table 4-1: VisualMotion Toolkit Programming Areas

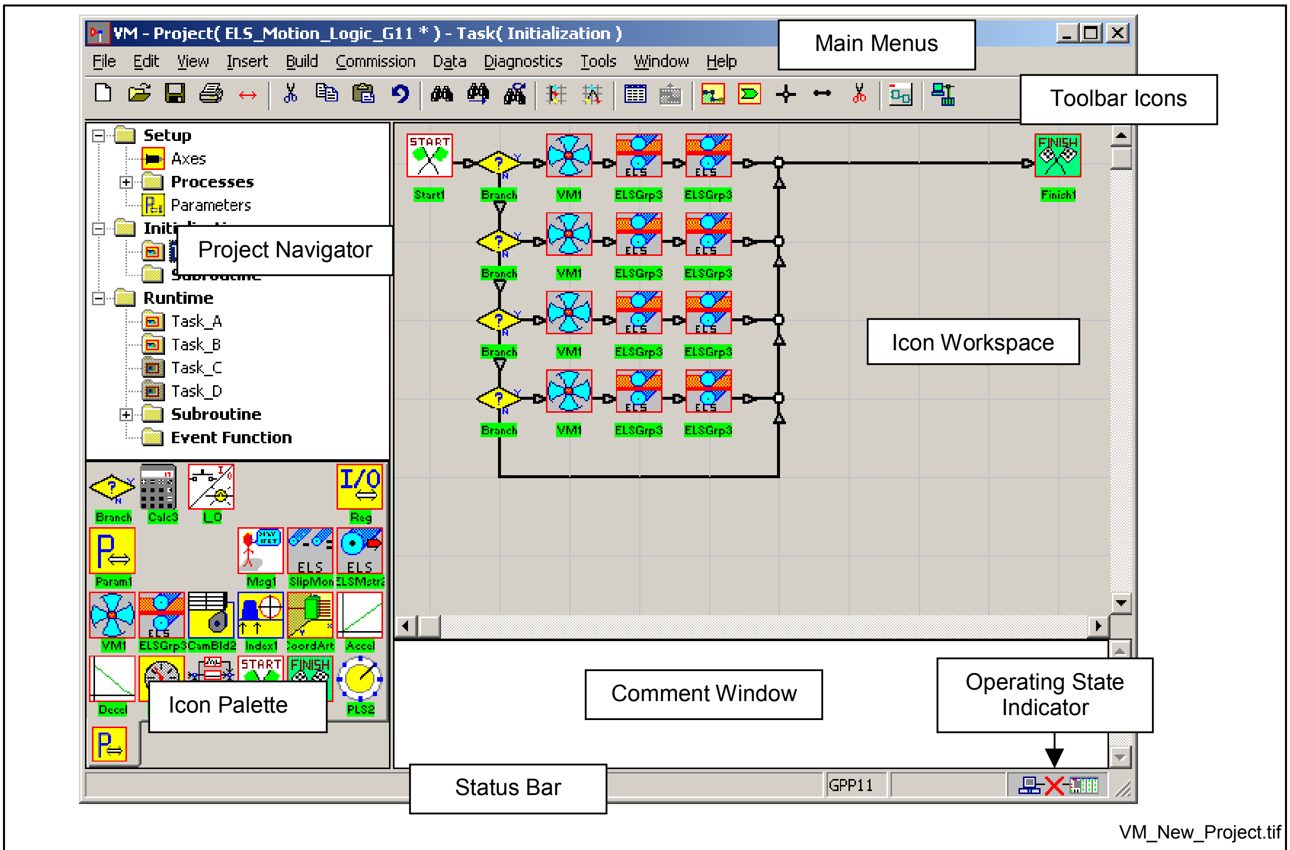


Fig. 4-2: VisualMotion Programming Environment

Main Menus

VisualMotion's main menu selections provide access to the majority of commands; from file selection to establishing communication to the launching of tools used to create and manipulate data in a project and on the control. Refer to chapter 13, *Menu Descriptions*, in volume 2 for complete details.

Toolbar Icons

The toolbar icons that appear just below the main menu are shortcuts to commonly used main menu items. The following toolbar icons are only available from the toolbar:

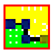
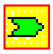
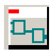
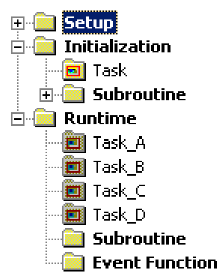
Toolbar Icon	Description
	**Subroutine**: actual icon that is place in the program flow to call an existing subroutine icon program
	**Connector**: allows the path of a program to flow between two points that are not connected by a line
	**Joint**: used to create an intersection of program lines
	Connect: used to connect two objects, normally icons, in a logical program flow
	**Scissors**: used to remove a connecting line between two points
	**IndraLogic**: used to launch IndraLogic PLC software. Also available under Commission menu.

Table 4-2: Toolbar Icons Not Found in Menus

Refer to chapter 14, *Icon Descriptions*, in volume 2 for complete details.

Project Navigator

The project navigator window displays a tree structure containing a *Setup* folder, an *Initialization* folder, and a *Runtime* folder. These folders expand to reveal additional sub folders used to configure, initialize, and program a VisualMotion system.



ProjectNavigator.tif

Fig. 4-3: Project Navigator

Rexroth
Bosch Group

Setup Folder

The Setup folder provides system initialization for all axis motion types, ELS System components, CAM Indexer, and PID Loops. In addition to axes and functionality initialization, parameter transfers can be configured for single or multiple C, A, T, S and P parameter sets.

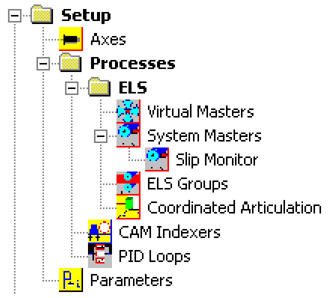


Fig. 4-4: Setup Folder

Refer to Setup Folder Tree Structure on page 4-8 for details.

Note: The Setup folder configuration is required for either Icon or PLC programming.

Initialization Folder

The initialization folder provides navigation to an icon based programming workspace for a task and subroutine that are executed every time the control transitions out of Sercos phase 2 (parameter mode).

Initialization Task

Icon programming begins with the initialization task. The initialization task serves:

- to allow modifications to motion resources such as ELS components by means of conditional programming,
- the initialization of program variables, activation of initialization subroutines, as well as register and parameter transfer functionality.

Note: When creating the initialization task and the initialization subroutine, avoid using the Branch icon to set up functions that run in an infinite loop. The initialization task times out in 5 minutes.

Disabling a Drive in the Initialization Task

To disable a drive from the initialization task that was configured under ***Setup*** ⇒ ***Axes***, use a Param1 icon and transfer a value of 2 to axis parameter A-0-0007 for the relevant axis set.

Runtime Folder

The runtime folder provides navigation to icon based programming workspaces for Tasks A-D, subroutines, and event functions. Runtime icon programs are executed in Sercos phase 4.

Tasks A-D Tasks A-D are used to build the main motion project. These tasks can contain program icons configured with command signals for motion in the drive. These tasks make up the majority of the icon program functionality. VisualMotion can have up to 4 separate processes or tasks running simultaneously in each project.

Subroutine Subroutines are optional sub-programs that are called by a task, other subroutine, or event function when the subroutine icon is executed in the program flow. They are used to improve readability as well as to simplify project organization. Refer to the Subroutine icon

Event Functions Unlike subroutines, event functions are not "called" from a program. Instead, they are "triggered" by conditions (distance, time, etc.) that are specified in an event setup. Refer to the Event icon

Icon Palettes

VisualMotion Toolkit icon palettes are displayed below the *Project Navigator* window. Five standard palettes are provided for Initialization, Single, Coordinated, ELS and Utility icons. Icon palettes can be selected from the ***View*** ⇒ ***Icon Palette*** menu selection or by clicking on an icon tab, just below the icons in the palette. The initialization icon palette is available only when the **Initialization Task** is selected from the Project Navigator. Icons are selected from the palette using a single click of the left mouse button. The selected icon is placed in the VisualMotion workspace by positioning the cross-hair cursor over the grid area where you want the icon to appear and clicking once.

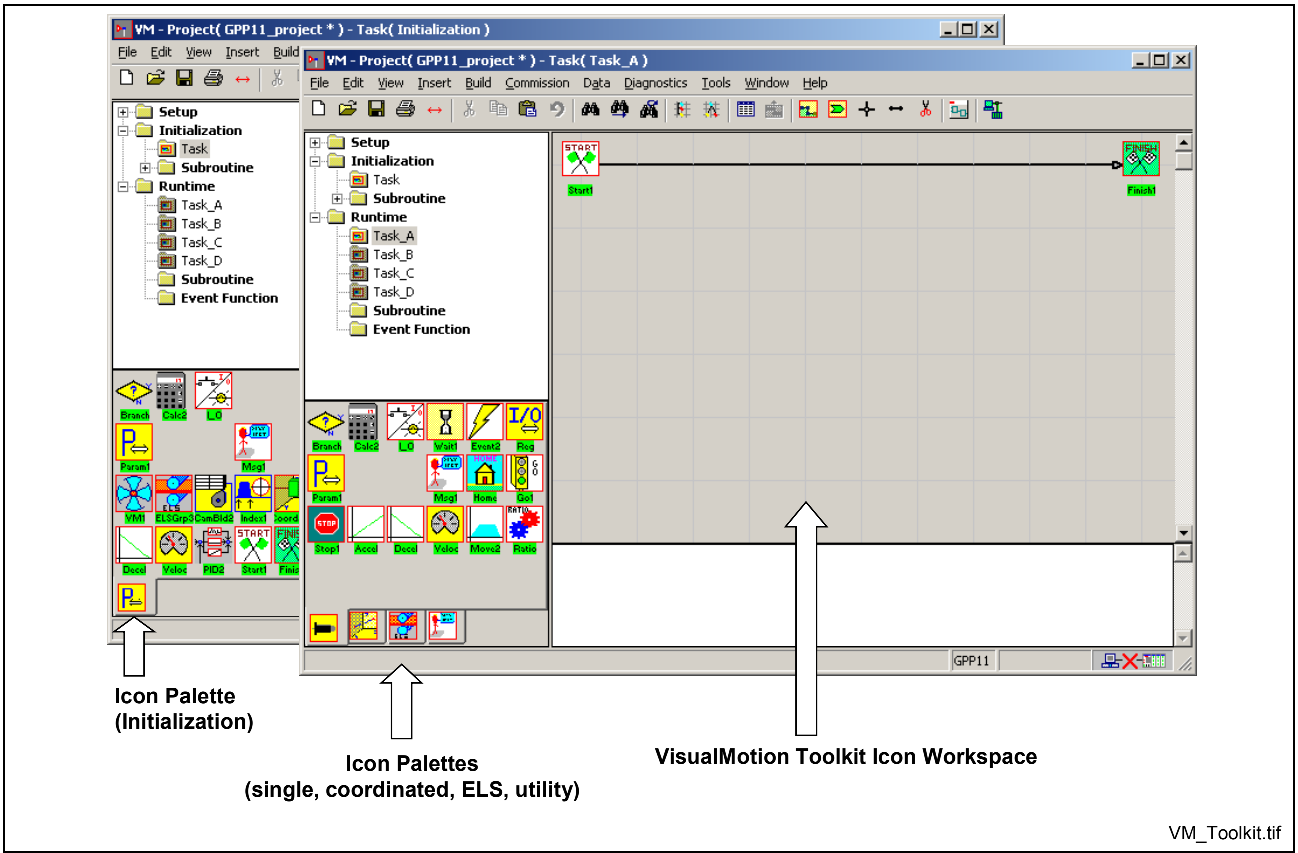


Fig. 4-5: Selecting Icon Palettes

Rexroth Bosch Group

Comment Window

Below the icon workspace is a comment window that is used to document information related to the currently selected task. By default it appears directly below the icon workspace.

The placement options for the comment window as well as other view options are available under **Tools** ⇒ **Options** ⇒ **View**. The comment window can be hidden by selecting ***View*** ⇒ ***Function Comment*** and removing the check next to the ***Function Comment*** menu item.

Status Bar

The status bar provides a visual indication of a process that is currently being executed.

Programming Modes

Two programming modes are supported in VisualMotion Toolkit. These are Project mode and Service mode. The current programming mode is displayed in the status bar of the main window and additional windows such as the Parameter Overview.

Project Mode

Project mode refers to the state of VisualMotion Toolkit when it is possible to interact with data in both the control's memory and in the project file stored on the PC, depending on whether the project is online or offline. Online and offline modes are indicated in the status bar of the VisualMotion window in the lower right corner, see Fig. 4-6.

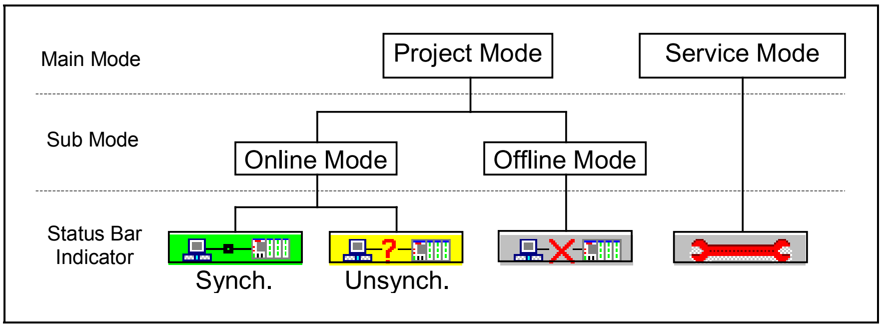


Fig. 4-6: Program Modes

Offline Mode

In offline mode, VisualMotion Toolkit is not communicating with the control. All data management is directed to the project that is stored in a single subdirectory on the computer.

Commissioning Tools in Offline Mode

Commissioning tools, such as the I/O Setup, Fieldbus Mapper and PLS, can be modified or configured while online or offline. If a configuration for a commissioning tool exists in the project folder, it will automatically be loaded when the tool is commissioned.

Online Mode

In online mode, VisualMotion Toolkit is communicating with the control. The viewing and editing of data in the control's memory is possible. The following two modes of synchronization are supported:

Synchronization Mode	Description
Synchronized	Active communication with the control. Modifications to the project are updated on control's memory and project folder.
Unsynchronized	Active communication with the control. Modifications to project data are only updated on control's memory.

Table 4-3: Online Mode Synchronization

Commissioning Tools in Online Mode

Commissioning tools automatically open with their configurations loaded in online mode, if previously configured. Modifications to data while online, immediately take effect in the project and control, depending on the state of the project and the type of data being modified.

Service Mode

Service mode is the state of VisualMotion Toolkit when it is possible to view and edit data in the control's memory only. Making changes while in Service mode only affects the data in the control, not the data stored in the project files on the PC. Edits to a project can be saved to the computer, but in a separate file from the project folder. The file can then be imported into the project folder. Refer to chapter 10, *Project Management Tools*, for details on how to *Import Project Data*.

4.2 Setup Folder Tree Structure

VisualMotion 11 introduces the addition of the **Setup** folder tree structure to the Project Navigator window only for G*P 11 projects. The Setup folder is used to configure all initialization data for Axes and system functionality such as ELS, Coordinated Articulation, CAM Indexer, and PID Loops. In addition to axes and system functionality initialization, the Parameters sub-item allows the initialization of parameters and parameter bit states formerly found in the PrmInt and PrmBit icons.

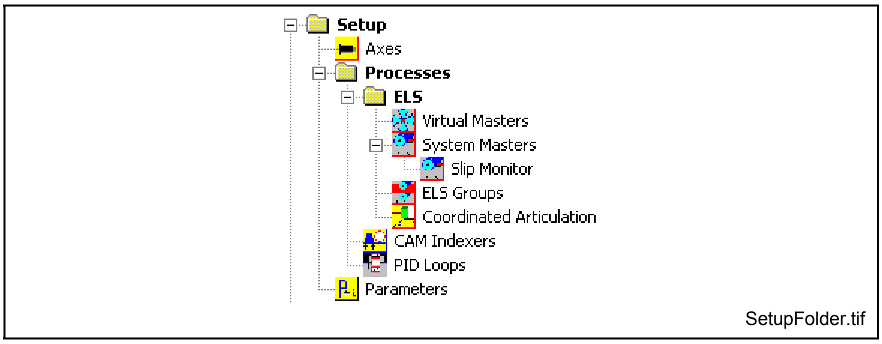


Fig. 4-7: Setup Folder Structure

This new method of initializing axes and system functionality has made it necessary to remove and/or modify existing programming icons in order to separate initialization data from runtime data.

Note: Pre-G*P targets using VisualMotion Toolkit 11 will not display the Setup folder in the Project Navigator window. Icons for pre-G*P 11 firmware targets are not affected unless the program is converted into a G*P 11 project.

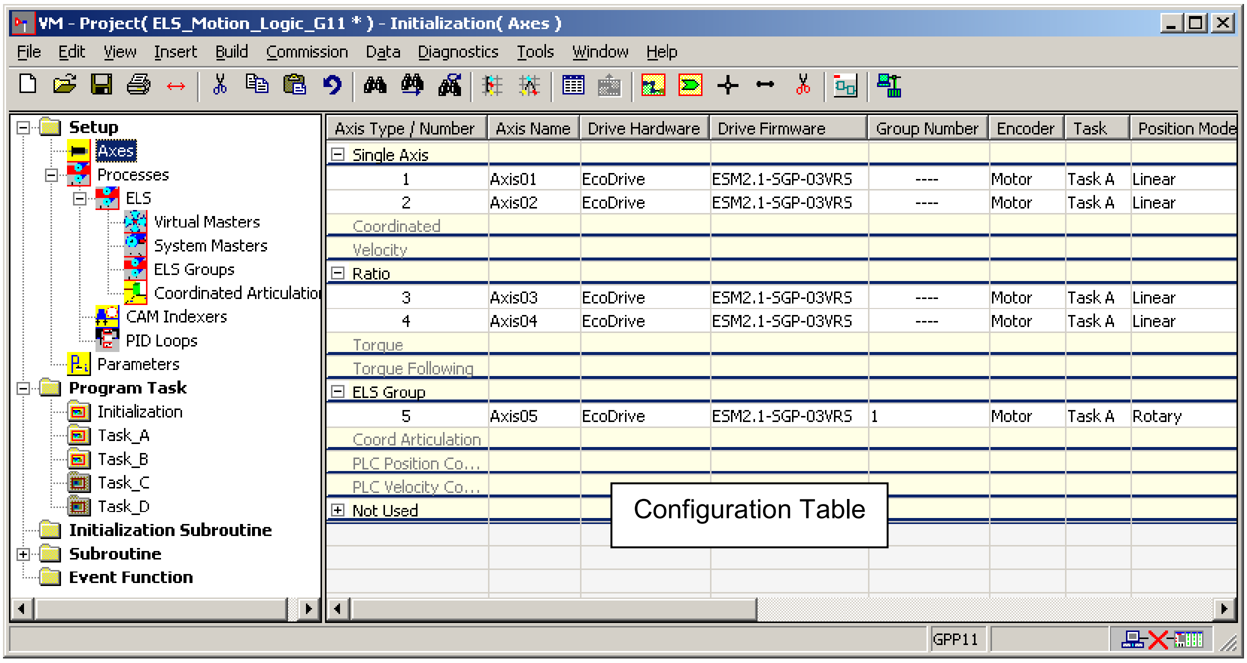
The following table provides an overview of which icons were removed and/or modified for G*P 11 projects:

Setup Sub-item	Icon Affected	Result for G*P 11 Projects
Axes	Axis	Removed from Initialization palette
Virtual Masters	VM	Place in initialization task to initialize program variable values for each Virtual Master configured under Setup
System Masters	ELSMstr	Modifies existing system masters under Setup or adds new system masters, including Slip Monitor
ELS Groups	ELSGrp	Place in initialization task to initialize program variable values for each ELS Group configured under Setup
Coordinated Articulation	CoordArt	Place in initialization task to initialize program variable values for each Coord. Articulation configured under Setup
CAM Indexers	Index	Place in initialization task to initialize program variable values for each CAM Indexer configured under Setup
PID Loops	PID	Place in initialization task to initialize program variable values for each PID Loop configured under Setup
Parameters	PrmInt PrmBit	Removed from all icon palettes For G*P 11 projects. Will still appear in icon palette when opening pre G*P 11 projects

Table 4-4: Icons Affected by Setup Folder Structure

Setup Folder Configuration Tables

When any sub-item under Setup is selected, the right window (typically used for icon programming) displays a table structure listing all of the configurable initialization data for the selected sub-item. The number of columns and heading types depends on the selected sub-item.



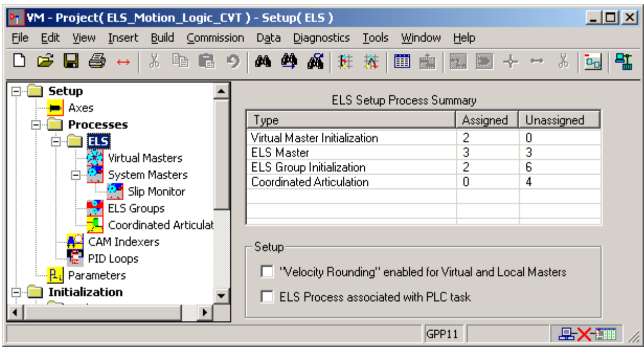
SetupConfigTable.tif

Fig. 4-8: Setup Configuration Table for Axes

Configuration Table Options and Settings

Every sub-item under the Setup folder displays unique column headings representing initialization data for the selected sub-item.

The following sections describe insertion and modification options as well as sorting and column reordering that is common to all sub-item tables. The only exception is the Processes and ELS sub-items. These sub-items display summary information for the related functionality listed below them in the tree structure.



ELS_Summary.tif

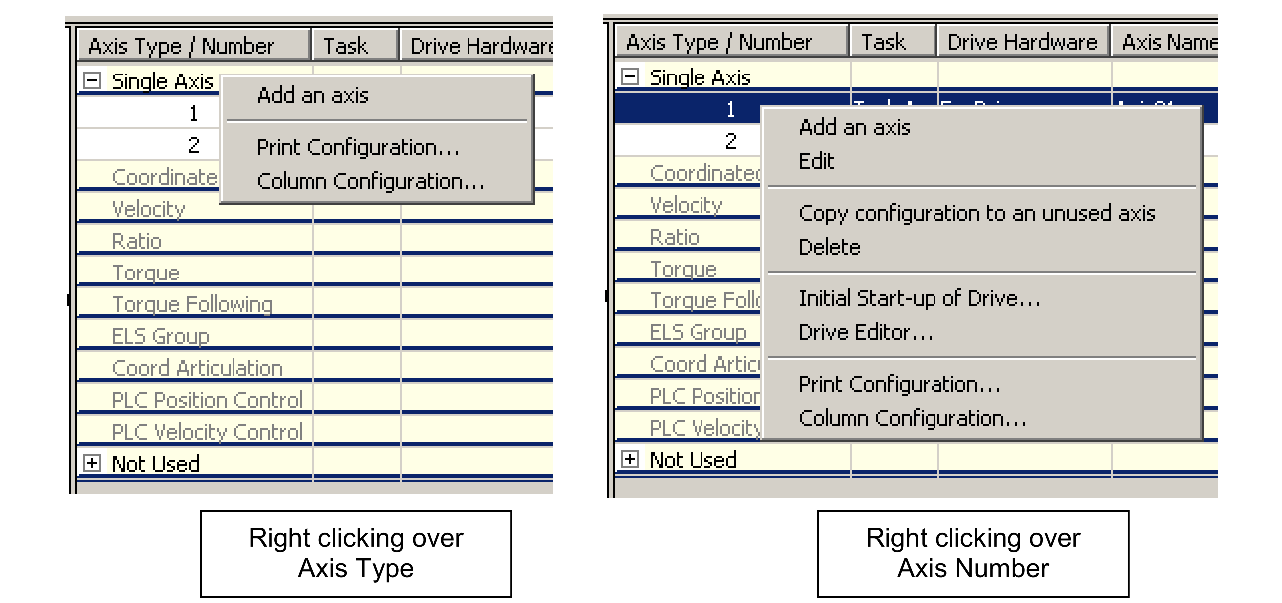
Fig. 4-9: ELS Summary Example

Right Mouse Click Options

The options available for the table configuration window vary between Axes, Processes, and Parameters.

Axes Options

The available options that appear in the popup window vary based on whether the user right clicks over an axis type (motion type) name or over an existing axis number. Refer to the following figure and table for details:



AxisOptions.tif

Fig. 4-10: Setup Axis Options

Option	Description
Add an axis	opens the Motion Selection window
Edit	edits an existing axis configuration
Copy Configuration to an unused axis	copies the same setup a new axis number selected by the user from a drop down list
Delete	deletes the axis from the list
Initial Start-up of Drive…	launches the DriveTop axis configuration wizard
Drive Editor…	launches the DriveTop overview window for the selected axis
Print Configuration	prints a list of the configured axes
Column Configuration	allows user to selected columns to be displayed

Table 4-5: Axes Options

Processes Options

Processes include ELS, CAM Indexer and PID Loops. The available options are the same for all processes with the exception of the type that is being added or modified. Refer to the following figure and table for details:

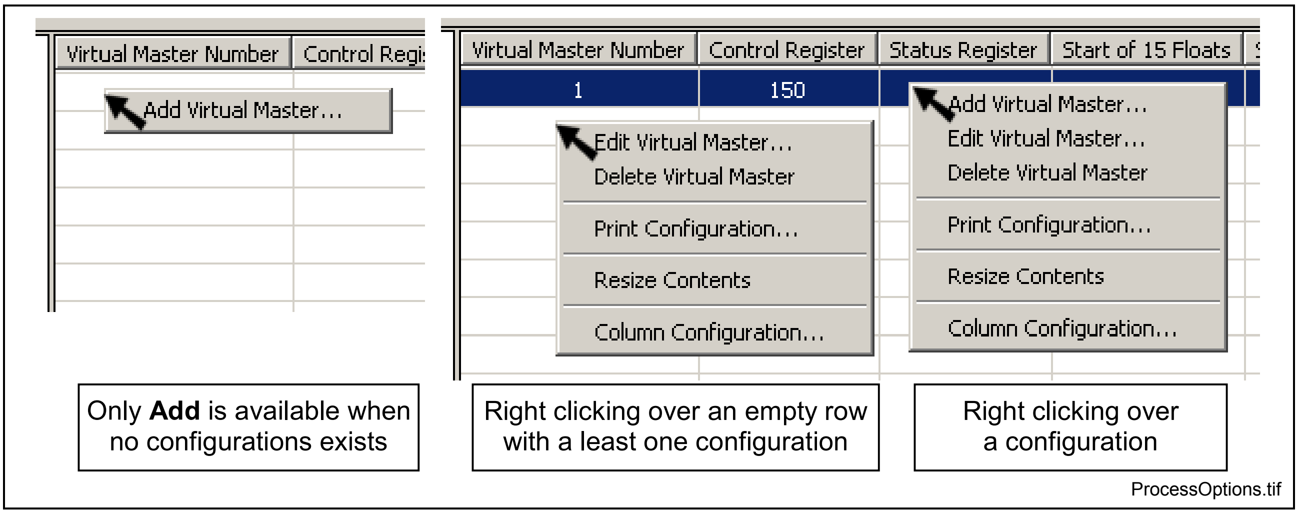


Fig. 4-11: Setup Processes Options

Option	Description
Add	adds a new configuration to the table
Edit	edits an existing configuration
Delete	deletes a configuration from the table
Print Configuration	prints a list of the configured axes
Resize Contents	Restores column width to default size
Column Configuration	allows user to selected columns to be displayed

Table 4-6: Process Options

Parameters Options

The available options are the same for all processes with the exception of the type that is being added or modified. Refer to the following figure and table for details:

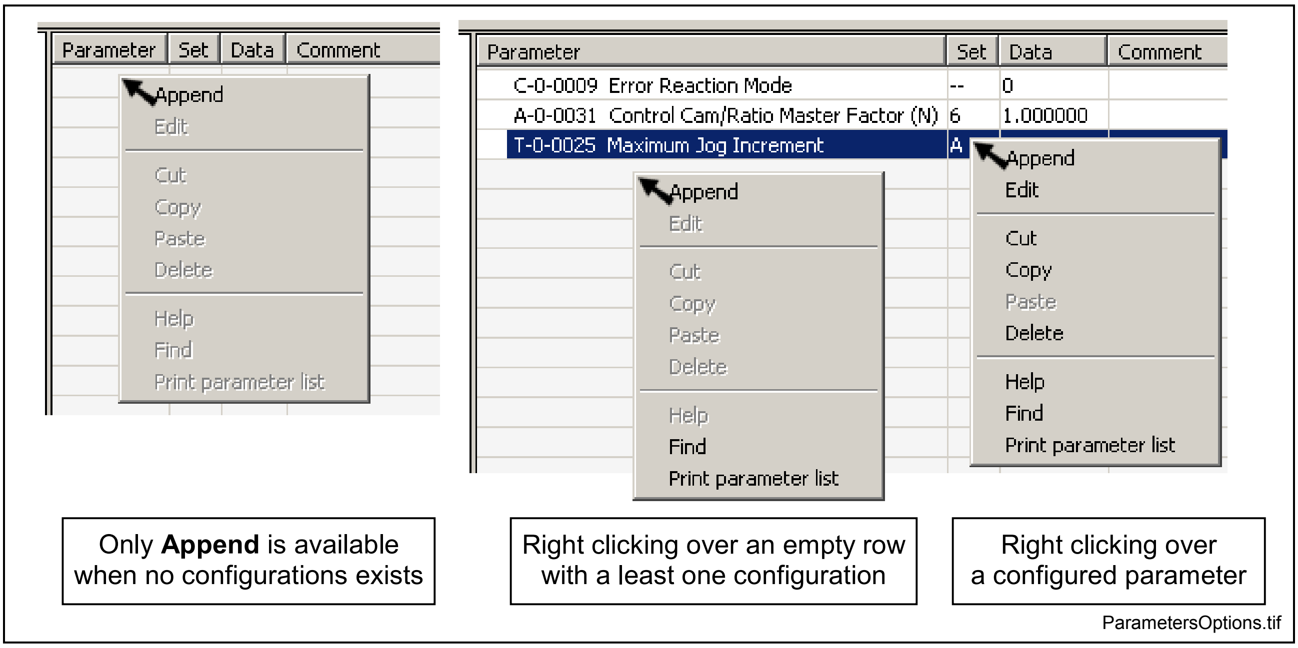


Fig. 4-12: Parameters Options

Rexroth Bosch Group

Option	Description
Append	Adds a parameter transfer configuration
Edit	edits an existing configuration
Cut	Removes and copies configuration to temporary memory
Copy	Copies the configuration to temporary memory
Paste	Adds the contents of the temporary memory
Delete	deletes a configured parameter from the table
Help	Opens context sensitive help for the selected parameter
Find	searches through table for specific entry
Print Parameter List	prints all configured parameters to selected printer

Table 4-7: Process Options

Modifying Data in a Table Configuration

Existing data in a table configuration can be modified by simply double clicking over the data field. Some data can be modified in the actual table cell, while other data opens in a secondary window in order to make modifications. Refer to the following figure for some examples:

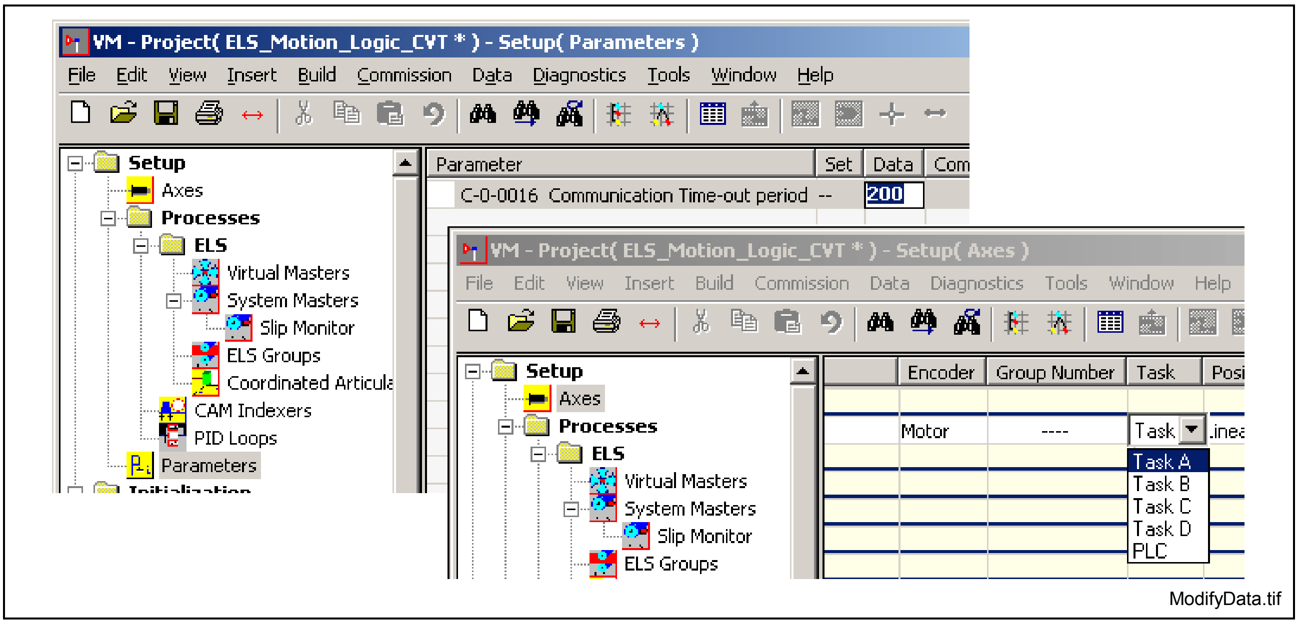


Fig. 4-13: Table Cell Modification Examples

Column Sorting and Re-Ordering

Column sorting is only available within Axes setup. To sort data based on a specific column, click on the column heading. Data is sorted for all configured axes within their respective axis types. For example, axes configured as single axis type will only be sorted within the single axis type section.

To change the order of columns for any axes, process or parameter init, click and hold the column to move and drag it, left or right, to the desired location.

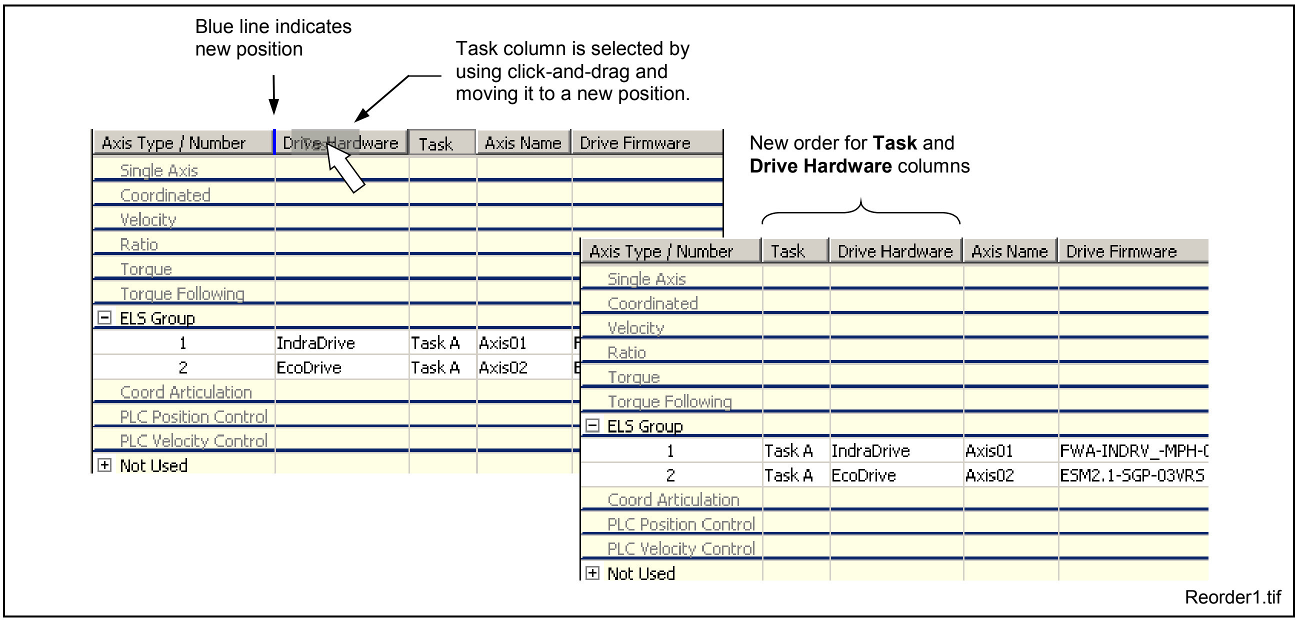


Fig. 4-14: Re-ordering Example

Column Configuration

The user can chose the columns to be displayed in any configuration table by right clicking and selecting Column Configuration. The item names available in the *Choose Columns* window vary based on the current Axes or Process Setup window.

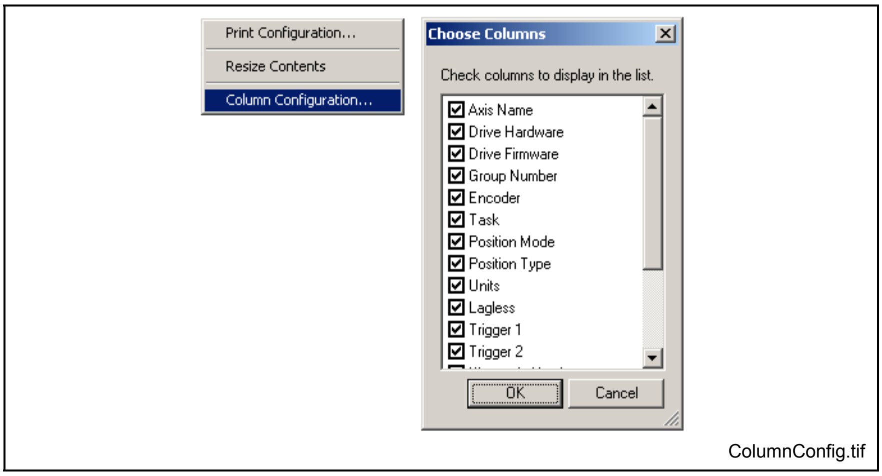


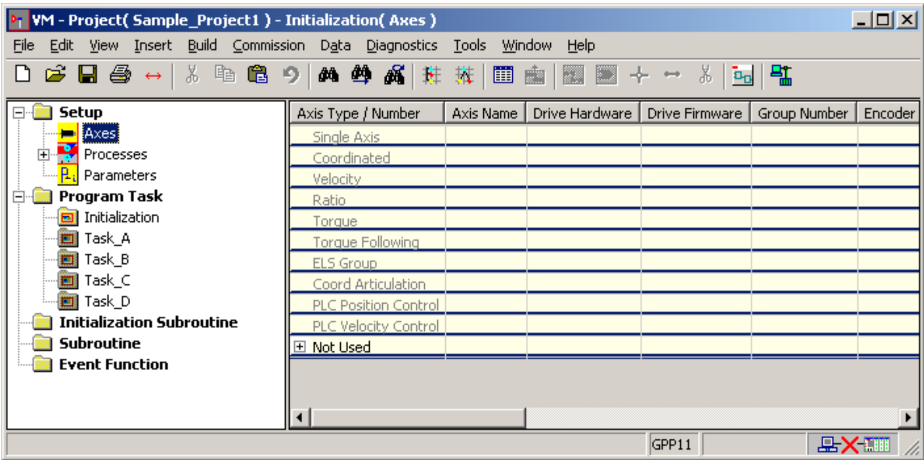
Fig. 4-15: Column Configuration Example

4.3 Setup Axes

VisualMotion 11 introduces a new presentation for configured axes in a G*P 11 project. All axes, regardless of motion type, are configured using the Setup Axes list. The Setup Axes list is the first selection in the *Project Navigator* window. Double click the ***Setup*** folder to expand the sub-items and then click on the **Axes** icon folder.

Note: The **Axis** icon has been removed from the initialization palette for G*P 11 projects. Pre-G*P 11 projects, opened using VisualMotion Toolkit 11 (without converting to a G*P 11 project) will display the Axis icon in the initialization palette. These programs can be compiled and downloaded to the control using VisualMotion Toolkit 11 with complete support for the Axis icon.

Older firmware targets (i.e., G*P 10) that are converted to a G*P 11 project should use ***Setup*** ⇒ ***Axes*** for any new axes configurations. Existing Axis icons in the program will be supported and can be compiled and downloaded to the control. However, support for new axis parameter initialization should be configured using ***Setup*** ⇒ ***Parameters***. Refer to Setup Parameters on page 4-68 for details.



AxesSetup1.tif

Fig. 4-16: Project Navigator showing Axes Setup

Note: The **Not Used** axis type selection at the bottom of the list expands to display the current overview of axes that have not been setup. Double clicking on an unused axis opens the *Motion Selection* window for setting up the axis.

The following table lists the entire column heading that are available by moving the scroll bar along the bottom of the Setup Axes list.

Column Heading	**Description**
Axis Type / Number	displays system motion types and axis Sercos address for configured axes
Axis Name	default or user-defined label assigned to axis
Drive Hardware	selected drive hardware for configured axis
Drive Firmware	selected drive firmware for configured axis
Group Number	valid only for axes configured under ELS Group motion type
Encoder	Motor (primary motor encoder) or External (secondary encoder)
Task	Task assigned to axis: Task A, Task B, Task C, Task D, or PLC
Position Mode	Rotary or Linear
Position Type	Absolute or Modulo
Units	inches or mm
Lagless	Additional velocity feed forward gain added to the position command
Trigger 1	triggering state for probe functionality
Trigger 2	triggering state for probe functionality
Kinematic Number	Valid only for Coordinated and Coord. Articulation motion types
Kinematic Axis	valid only for Coordinated and Coord Articulation motion types
ELS Sync Type	Velocity, Phase, Control CAM or Drive CAM
Phase Sync	Absolute or Relative synchronization method
Master Axis	valid only for Ratio motion type
Comment	text entered by the user identifying the axis

Table 4-8: Setup Axis Column Heading

Note: Any column right of the *Axis Type / Number* column can be reordered by selecting and dragging the column heading and moving it left or right.

Specifying Axis Hardware and Firmware

During the axis setup procedure, the user must specify the drive hardware and firmware types that will be used for each axis. This procedure can be performed in offline or online mode.

Offline Mode

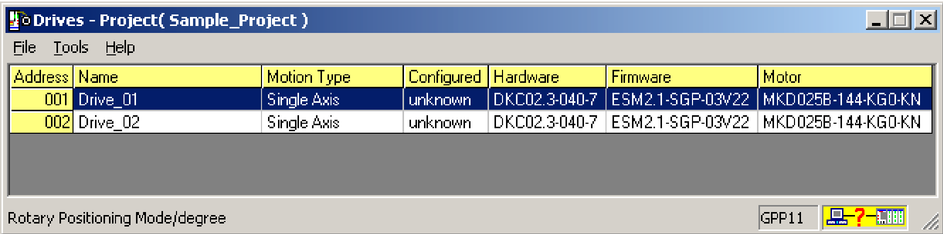
In offline mode, the user must know ahead of time what drive hardware and related firmware will be used for each axis.

Online Mode

In online mode, with a connected control using G*P 11 firmware, the user can retrieve the hardware and firmware types for all drives connected on the Sercos ring. To access the Sercos ring, switch VisualMotion Toolkit to online unsynchronized mode and selecting ***Commission ⇒ Drive Overview…***

Note: For details on how to switch to online unsynchronized mode, refer to menu selection *File ⇒ Online/Offline…F9* in chapter 13 of volume 2.

These steps open the *Drive Overview* window displaying all connected drives as shown in the figure below.



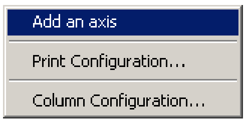
DriveOverview.tif

Fig. 4-17: Commission Drive Overview in Online Unsynchronized Mode

From this window, the user can view the current hardware and firmware types for each axis in the Sercos ring. This window can remain open during the axis setup procedure.

Add an Axis

To add an axis, right click over the desired motion type, for example **Single Axis**, and select ***Add an axis*** to open the *Motion Selection* window.



AxesSetup.tif

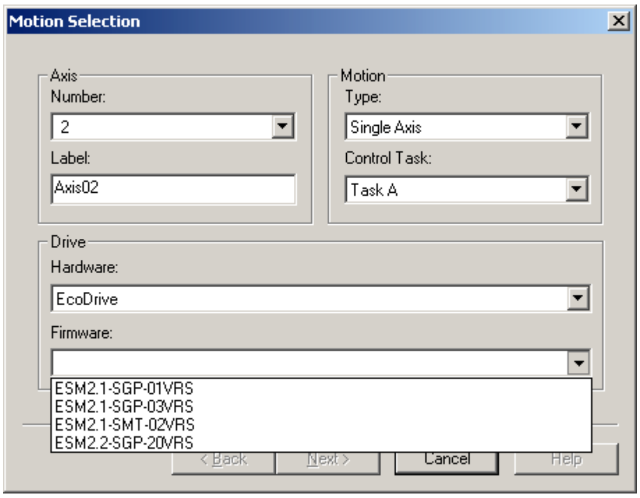
Fig. 4-18: Add an Axis Popup

Motion Selection

The *Motion Selection* window is used by all axis types to setup the basic axis information. The following table details the available selections:

Selection	Details
Axis Number	List of available axis numbers representing the Sercos address.
Label	Default or user-defined variable label assigned to axis.
Motion Type	List of supported system motion types. Configured axis appears under the assigned motion type. Selection: Refer to Fig. 4-16 for available types.
Control Task	Task assigned for the control and status of the axis: Selection: Task A-D or PLC
Hardware	List of supported drive hardware. Selection: Refer to Specifying Axis Hardware and Firmware on page 4-16 for details.
Firmware	List of supported drive firmware based on selected hardware.

Table 4-9: Motion Selection Settings



MotionSelection.tif

Fig. 4-19: Motion Selection

Axis Properties

The *Axis Properties* window is used by all axis types to setup to **Positioning Mode**, **Encoder** type, **Positioning Type**, and **Units**.

The following table details the available selections:

Selection	Details
Lagless	Additional velocity feed forward gain added to the position to reduce any positioning lag in the system. Improved reaction in the system drive settings
Positioning Mode	When set to **Linear**, the units and scaling are specified in parameter A-0-0005 Linear Position Units. Absolute positioning is enabled in the drive. Refer to A-0-0005 for details. When set to **Rotary**, the position, velocity, and acceleration units and scaling are fixed at the drive. Modulo positioning is enabled by default, with a rollover value specified in drive parameter S-0-0103.
Positioning Type	For Rotary positioning mode, refer to A-0-0004, bit 15 For Liner positioning mode, refer to A-0-0004, bit 13
Encoder	**Motor** (primary motor encoder) or **External** (secondary encoder). Refer to A-0-0004, bit 11 **Redundant** (available when configuring an IndraDrive using MPx 04 firmware) A-0-0004, bit 7
Units	Available units are based on selected positioning mode: For Rotary mode, units are in **degrees**. For Linear mode, units are in **inches** or **mm**. Refer to A-0-0005

Table 4-10: Axis Properties Settings

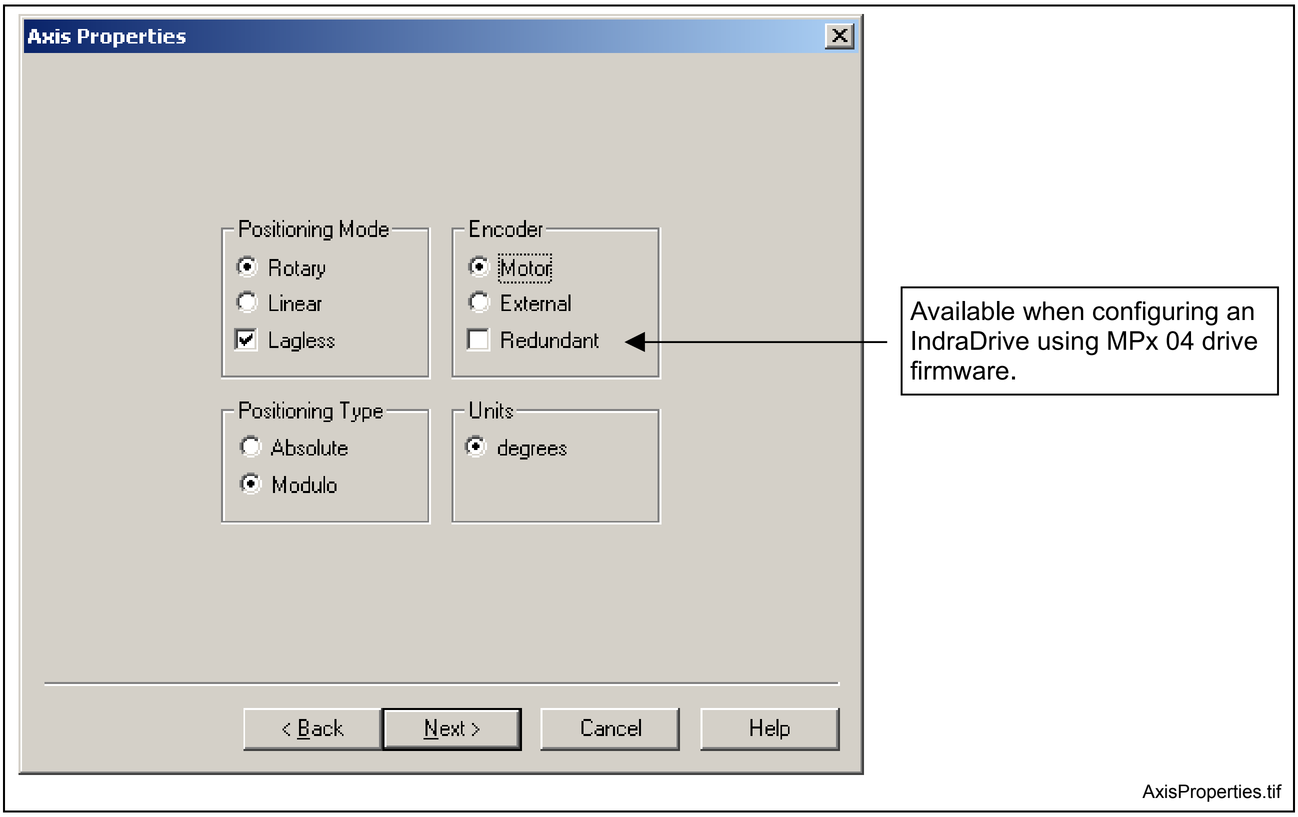


Fig. 4-20: Axis Properties

Note: The redundant motor encoder option maintains machine functionality and provides a controlled stop in the event that the primary motor encoder is lost.

Configure Axis Probe

The *Configure Axis Probe* window is used by all axis types to enable the **Trigger 1** and **Trigger 2** drive based position capture for selected drive I/O input, probe 1 or probe 2. Capture can be on 0->1 or 1->0 transition. Refer to *Probe Events* in section 5.2 for a detailed explanation of the functionality.

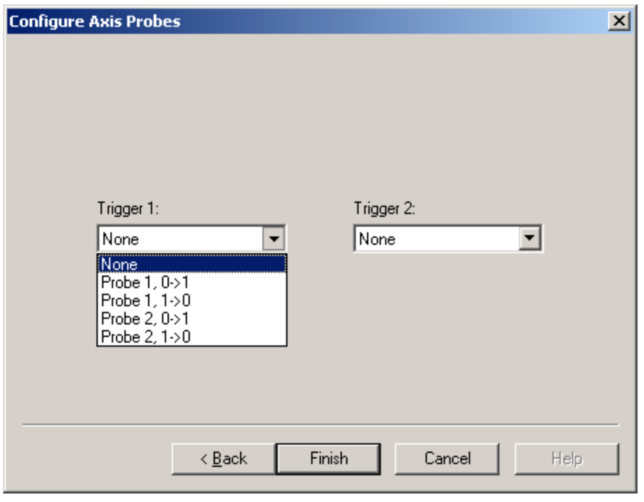


Fig. 4-21: Configure Axis Probes

Single Axis Setup

To setup a single axis, right click over the **Single Axis** name and select ***Add an axis***.

The following windows are used to configure a Single Axis:

- Motion Selection on page 4-17
- Axis Properties on page 4-18
- Configure Axis Probe on page 4-19

Note: For Single Axis mode, an axis is configured as Initially Halted. This disables the axis at the start of the task and must be enabled using a **GO** icon in the user program.

Coordinated Axis Setup

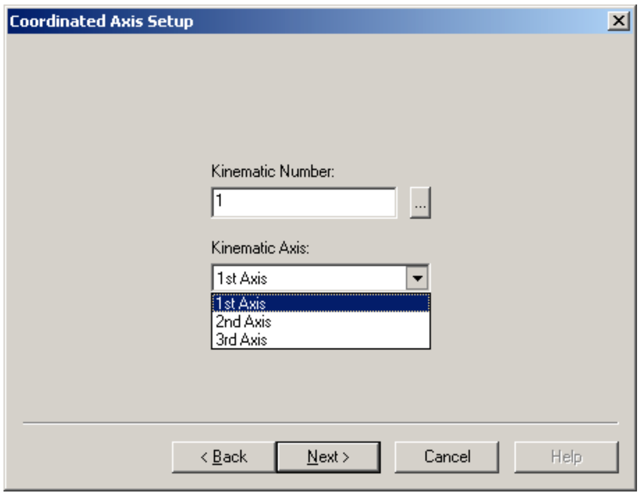
To setup an axis as coordinated, right click over the **Coordinated** name and select ***Add an axis***.

The following window appears first during the setup of a Coordinated Axis:

- Motion Selection on page 4-17

The following window is unique to the Coordinated Axis Setup and is used to select a **Kinematic Number** and **Kinematic Axis**.

Note: Each axis used in a coordinated application must be setup individually under a different Kinematic Axis.



CoordinatedAxisSetup.tif

Fig. 4-22: Coordinated Axis Setup

After the *Coordinated Axis Setup* window, the following two windows complete the setup:

- Axis Properties on page 4-18
- Configure Axis Probe on page 4-19

Velocity Axis Setup

To setup an axis as velocity, right click over the **Velocity** name and select ***Add an axis***. This axis type permits the selection of a Ramp instead of the default Step velocity changes used in single axis. Ramp velocity changes are based on current acceleration rates, whereas Step changes are made at the maximum rate allowed by the drive and motor.

The following windows are used to configure a velocity axis:

- Motion Selection on page 4-17
- Axis Properties on page 4-18
- Configure Axis Probe on page 4-19

Ratio Axis Setup

The Ratio Axis Setup window is used to assign a slave axis to a master (or controlling axis). More than one axis may be slaved to a single master. A ratioed axis may be assigned linear or rotary positioning, and may use the drive's probe capability to trigger events.

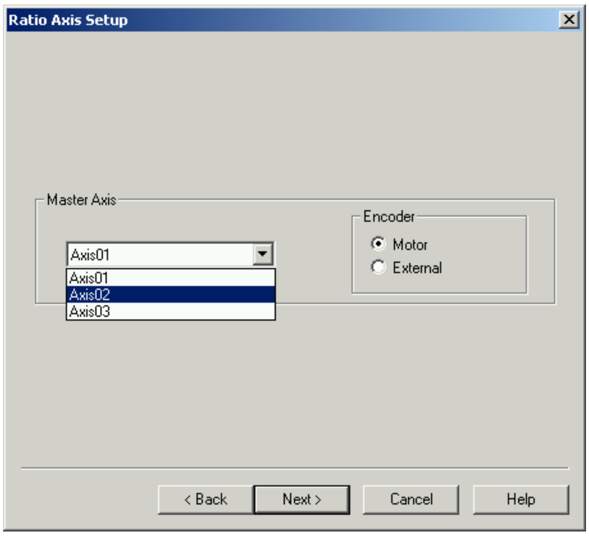
In control Ratio Mode, the slave axis follows the command position of the master axis if the master is a coordinated motion or control CAM axis. If the master is any other type of axis, the slave can follow the axis feedback or an external secondary encoder feedback position.

The following window appears first during the setup of a Ratio Axis:

- Motion Selection on page 4-17

The following window is unique to the *Ratio Axis Setup* and is used to select a **Master Axis** and **Encoder** type. To select a master axis, use the drop-down list.

Note: If the desired master axis is not displayed, it must first be setup before it can be selected as a Master Axis.



RatioAxisSetup.tif

Fig. 4-23: Ratio Axis Setup

After the *Ratio Axis Setup* window, the following two windows complete the setup:

- Axis Properties on page 4-18
- Configure Axis Probe on page 4-19

The axis mode can be switched to single-axis or velocity mode from within the program or for manual mode jogging. It is also possible to "slave off" from the master and position the axis independently. Setting the following ratio to 0 will stop the slave from following the master. A step rate (A-0-0037) allows gradual adjustment of the ratio.

Note: To keep the slave axis synchronized when the master axis is jogging, set Axis Control register bit 4 (Synchronized Jog bit). If the slave axis is never switched to single-axis mode, this bit should be mapped high.

Torque Axis Setup

To setup an axis as torque, right click over the **Torque** name and select ***Add an axis***. Torque mode is used to apply constant torque to the specified axis. The update rate is limit by the Sercos cycle.

The following windows are used to configure a torque axis:

- Motion Selection on page 4-17
- Axis Properties on page 4-18
- Configure Axis Probe on page 4-19

Torque Following Axis Setup

To setup an axis as torque following, right click over the **Torque Following** name and select ***Add an axis***. Torque following mode is used to setup a slave axis that follows the torque value of a master axis. The torque value is transmitted from the master to the slave via an analog interface cable.

The following windows are used to configure a velocity axis:

- Motion Selection on page 4-17
- Axis Properties on page 4-18
- Configure Axis Probe on page 4-19

ELS Group Axis Setup

To setup an axis as ELS Group axis, right click over the **ELS Group** name and select ***Add an axis***.

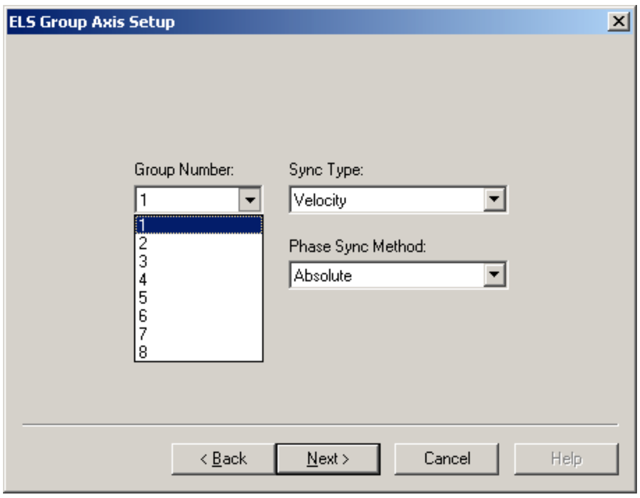
The following window appears first during the setup of an ELS Group:

- Motion Selection on page 4-17

The following window is unique to the *ELS Group Axis Setup* and is used to setup the **Sync Type** and **Phase Sync Method** for the selected **Group Number**. The following table details the available selections:

Selection	Details
Group Number:	Select the ELS Group number that will be assigned to the axis.
Sync Type:	The following selections are available: **Velocity** – the slave axis matches and follows an ELS Group's output velocity regardless of any phase differences. **Phase** – the slave axis matches and follows an ELS Group's output position. **Control CAM** – the slave axis follows a control CAM profile whose master position is the ELS Group's output. **Drive CAM** – the slave axis follows a drive CAM profile whose master position is the ELS Group's output. **Electronic Motion Profile** – the slave axis follows a defined motion profile based on the selection from 1 up to 8 segments per master revolution.
Phase Sync Method	When the Sync Type is Phase, Synch Method specifies exactly how the Group will Phase Sync to the master. **Absolute** - the Group will synchronize to the exact same position as the master (with zero offset) -- if the Master is at 100.3 degrees, the Group will sync to 100.3 degrees. The Group will always follow the exact position of the master not withstanding any phase adjustments the user later performs. **Relative** - the Group will synchronize to whatever position the master is at after the velocity has been matched. If the master is at 100.3 degrees and the Group is at 150 degrees after velocity has been matched, the Group will maintain this relationship/offset while following the master. Another way to look at Relative is that whatever difference in position exists after the velocity has been matched is incorporated as an offset. The Group is still following the position of the Master, but with a non-zero offset.

Table 4-11: ELS Group Axis Setup Settings



ElsGroupAxisSetup.tif

Fig. 4-24: ELS Group Axis Setup

After the *ELS Group Axis Setup* window, the following two windows complete the setup:

- Axis Properties on page 4-18
- Configure Axis Probe on page 4-19

Coordinated Articulation Axis Setup

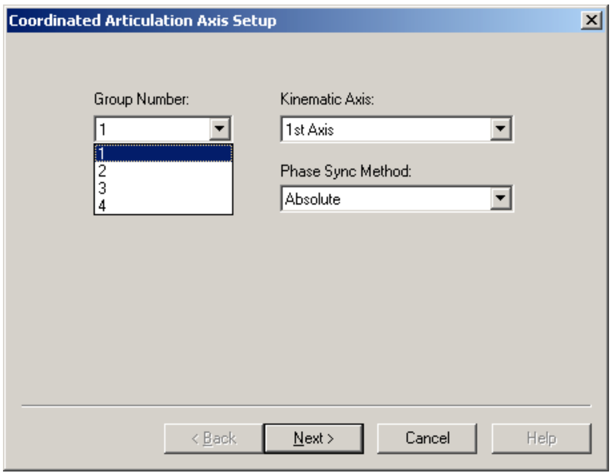
Coordinated Articulation is an advanced feature used to move up to six axes in coordinated fashion based on world coordinate inputs from an ELS Group output or manual positions. This feature provides the ability to link cyclic coordinated motion to an ELS master.

To setup a coordinated articulation axis, right click over the **Coordinated Articulation** name and select ***Add an axis***.

The following window appears first during the setup of a coordinated articulation axis:

- Motion Selection on page 4-17

The following window is unique to the Coordinated Articulation axis and is used to setup the **Group Number**, **Kinematic Axis**, and **Phase Sync Method**.



CoordArtSetup.tif

Fig. 4-25: Coordinated Articulation Axis Setup

The following table details the available selections:

Selection	Details
Group Number:	This number identifies the Coordinated Articulation configuration in a VisualMotion project. Up to 4 Coordinated Articulation configurations can exist in a project at one time.
Kinematic Axis:	Defines the Kinematic axis number in a Coordinated Articulation configuration. Each axis used in a coordinated articulation application must be setup individually under a new Coordinated Articulation axis setup.
Phase Sync Method	When the Sync Type is Phase, Synch Method specifies exactly how the Group will Phase Sync to the master. **Absolute** - the Group will synchronize to the exact same position as the master (with zero offset) -- if the Master is at 100.3 degrees, the Group will sync to 100.3 degrees. The Group will always follow the exact position of the master not withstanding any phase adjustments the user later performs. **Relative** - the Group will synchronize to whatever position the master is at after the velocity has been matched. If the master is at 100.3 degrees and the Group is at 150 degrees after velocity has been matched, the Group will maintain this relationship/offset while following the master. Another way to look at Relative is that whatever difference in position exists after the velocity has been matched is incorporated as an offset. The Group is still following the position of the Master, but with a non-zero offset.

Table 4-12: ELS Group Axis Setup Settings

After the *Coordinated Articulation Axis Setup* window, the following two windows complete the setup:

- Axis Properties on page 4-18
- Configure Axis Probe on page 4-19

User Position Axis Setup

VisualMotion 11 supports user implemented position control extensions from within IndraLogic (PLC). To setup a user position axis, right click over the **User Position** name and select ***Add an axis***.

The following windows are used to configure a user position axis:

- Motion Selection on page 4-17

Note:	The only Control Task that can be assigned to a User Position axis is the PLC.

- Axis Properties on page 4-18
- Configure Axis Probe on page 4-19

Note:	The IndraLogic MV_UserMotion library provides functions for implementing user position control functionality. The user is required to implement all motion control functionality from the IndraLogic project.

User Velocity Axis Setup

VisualMotion 11 supports user implemented velocity control extensions from within IndraLogic (PLC). To setup a user velocity axis, right click over the **User Velocity** name and select ***Add an axis***.

The following windows are used to configure a user velocity axis:

- Motion Selection on page 4-17

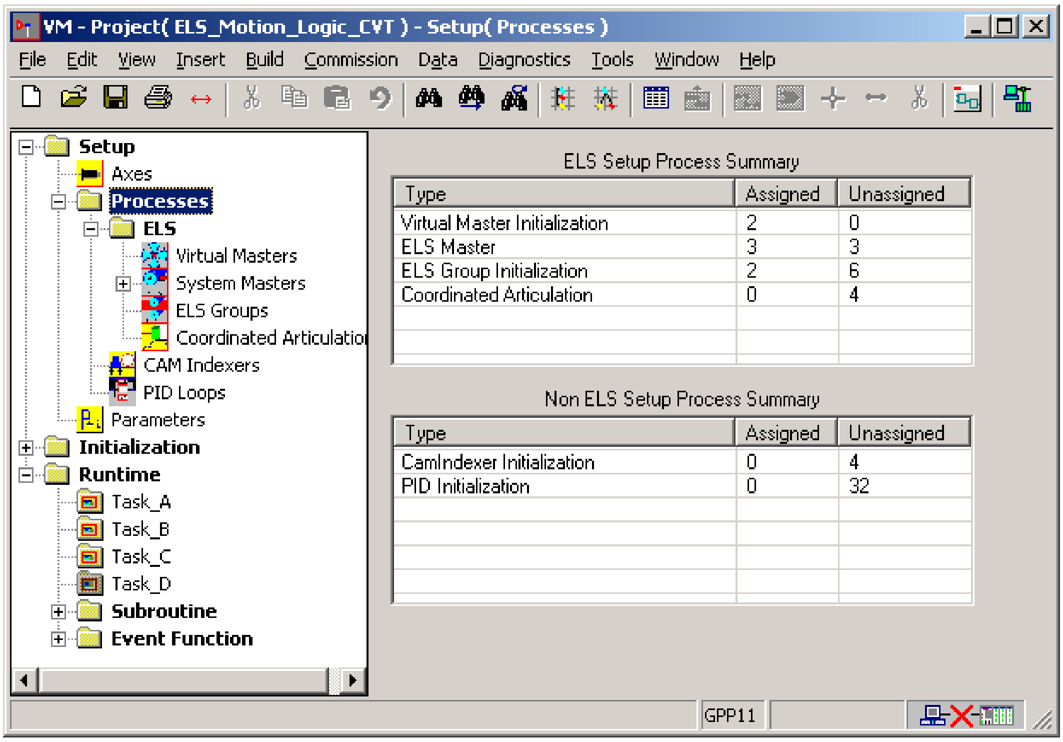
Note:	The only Control Task that can be assigned to a User Velocity axis is the PLC.

- Axis Properties on page 4-18
- Configure Axis Probe on page 4-19

Note:	The IndraLogic MV_UserMotion library provides functions for implementing user velocity control functionality. The user is required to implement all motion control functionality from the IndraLogic project.

4.4 Setup Processes

The Processes selection under Setup expands to list the configurations for **ELS**, **CAM Indexer** and **PID Loops**. A summary of the assigned and unassigned ELS and non-ELS setup processes are displayed to the right of the Project Navigator.



Processes.tif

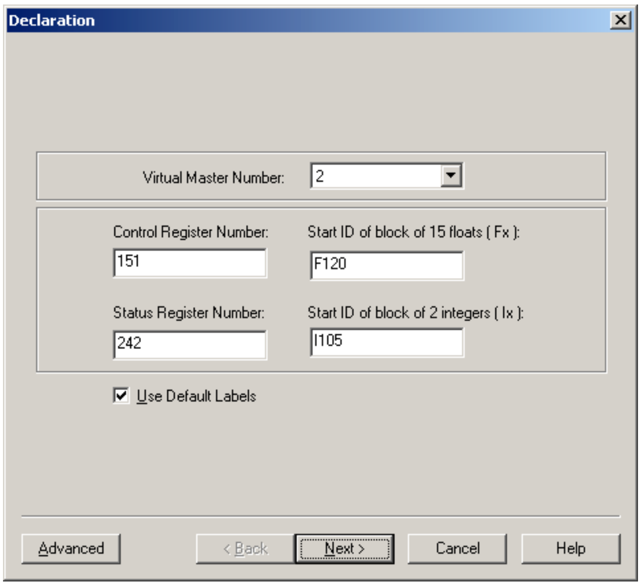
Fig. 4-26: Setup Processes

Default Labels and Initial Values

During the configuration process for any ELS, CAM Indexer, or PID Loop setup, the user has the option to select whether or not to use default labels and to define which program variables are to be initialized by the system or marked as retentive. Retentive program variables hold their last known value regardless of whether or not the system has been re-initialized.

Use Default Labels

Every declaration setup window contains a checkbox by which a user can choose to have default register and program variable labels automatically assigned by the wizard. By default the checkbox is selected. Labels will be added to each register and program variable starting from the value defined in the *Declaration* window.



DefaultLabels.tif

Fig. 4-27: Selecting Default Labels

Editing Default Labels

Default labels can be edited from the VM Data Table by selecting ***Edit ⇒ VM Data Table…*** and choosing the relevant tab (Registers, Bits, Ix, or Fx).

User-Defined Labels

It is recommended to initially use default labels. This way all default labels will be placed in the correct locations. Afterwards, the user can modify any labels to suit their specific application. However, the user must then edit the declaration again and remove the checkbox. Otherwise, all user-defined labels will be replaced with default labels if the declaration is edited and the **OK** button is clicked.

Special Considerations when using Default Labels

Default labels are assigned starting at the value defined for register numbers and start ids. The following behavior should be considered when modifying an existing register and start id numbers.

1. If default labels are not used, the user must label the register, register bits, and program variables starting at the assigned register numbers and start id bocks.
2. If modifications are made to the register numbers and/or start id blocks of an existing ELS process declaration, any existing default labels at the old location will be automatically deleted and re-labeled at the new locations. Any user-defined labels will be ignored. User-defined labels must be manually deleted.

Defining Process Variables to Retain Values

Every setup process (ELS, CAM Indexer, and PID Loop) contains program variables that can be set to retain their current value regardless of whether or not the control is phase transitioned or powered down.

All program variables have an additional selection that appears when right clicking over the value field for the specific variable. Refer to the following example for details:

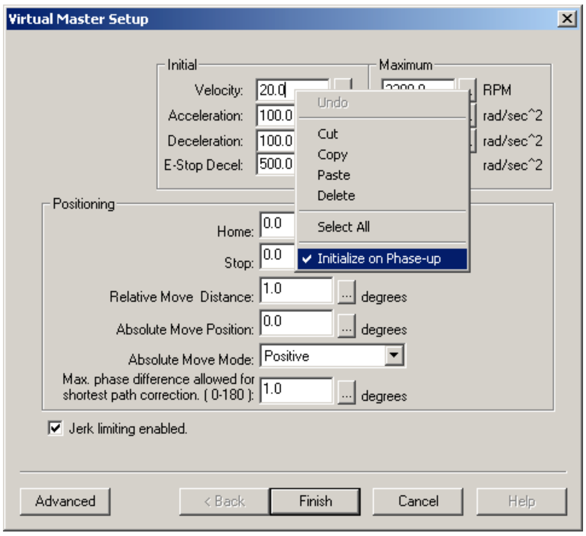


Fig. 4-28: Initialize on Phase-up Example

The default setting for all variables is to have them re-initialized every time the control is transitioned from Sercos phase 2 to Sercos phase 4. This is indicate by a check next to the *Initialize on Phase-up* option. Removing the check, allowing the variable to retain its current value when modified in the program. The system will not re-initialize the value for variables that have had the check removed next to Initialize on Phase-up option.

Advanced Button

This can also be accomplished by clicking on the **Advanced** button and removing the check from the desired program variable. The process that is currently being configured will appear expanded.

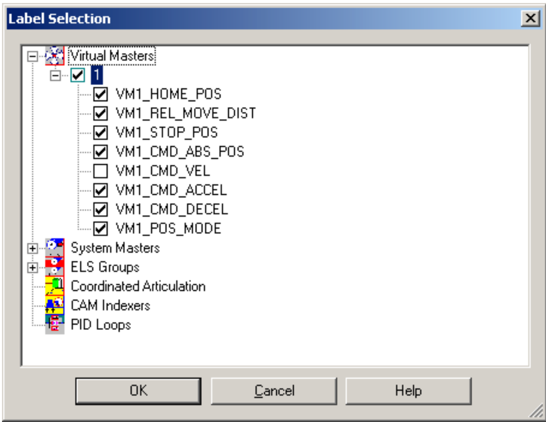


Fig. 4-29: Selecting Variables to Retain Value

The system can be configured to automatically check or uncheck process variables for processes configured under Setup. Refer to *Setup Options* under ***Tools*** ⇒ ***Options*** ⇒ ***Project*** in chapter 13, *Menu Description*, of volume 2.

Note: Only those processes that exist in Setup will appear in the Label Selection tree structure.

Once a variable is set to retain a value, the value is underlined and displayed in blue. The corresponding variable listed within the Advanced button (Label Selection window) will appear unchecked.

Rexroth
Bosch Group

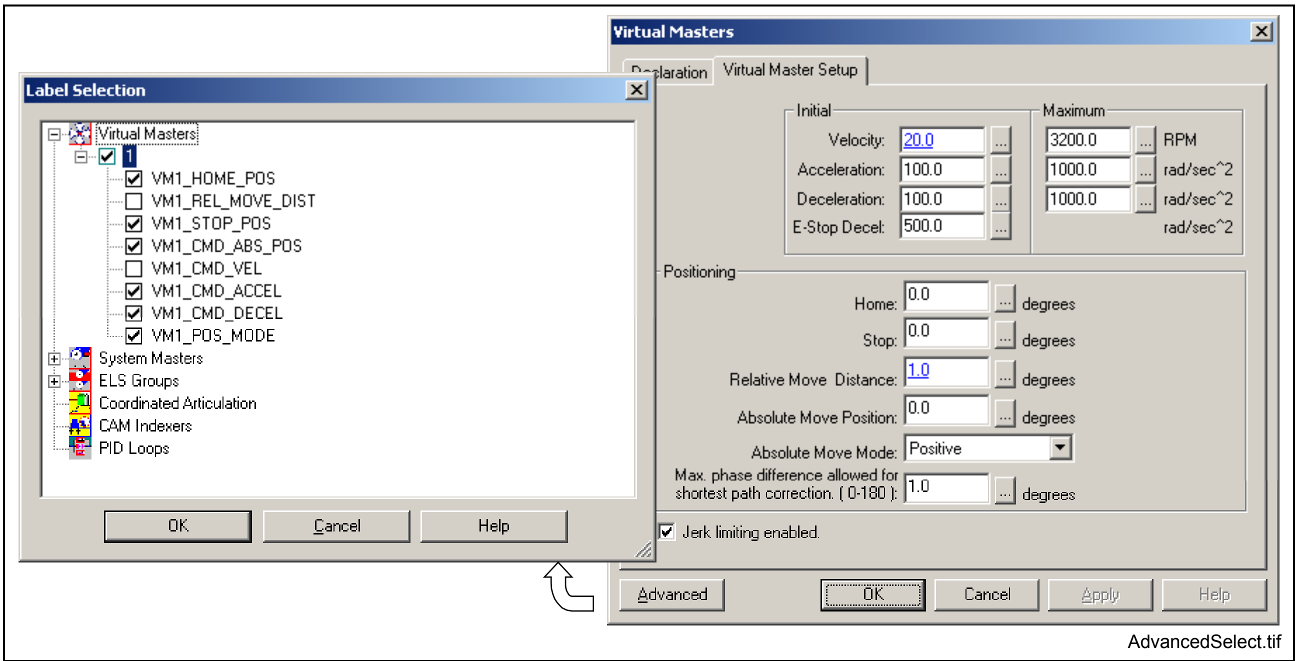


Fig. 4-30: Variables Displayed as Retaining Their Values

The value entered in any field, during the setup process, is compiled and downloaded to the control only once. Any modifications to the value of a variable selected to retain it's value, via user input or by the program, will be stored and not re-initialized by a Sercos phase transition. Any variable that is not marked to retain it's value will be re-initialized to the value initially set during the setup process, regardless of whether or not the value was modified during runtime.

CAUTION

Unexpected movement can result from a variable not starting from a safe initial value!

⇒ Once variables are set to retain their values, the user must be aware that a particular process will not have all of its relevant variable values starting from a safe initial value.

4.5 Setup ELS Process

The Setup ELS selection in the Project Navigator window expands to list the configurations for **Virtual Masters**, **System Masters** (including Slip Monitoring), **ELS Groups**, and **Coordinated Articulation**.

Note: All necessary ELS process setup can be performed within the Setup folder without the need for ELS icons in the Initialization task. ELS process icons are provided to allow the user to create conditional programming within the Initialization task.

The following two ELS options are only available when the ELS folder is selected:

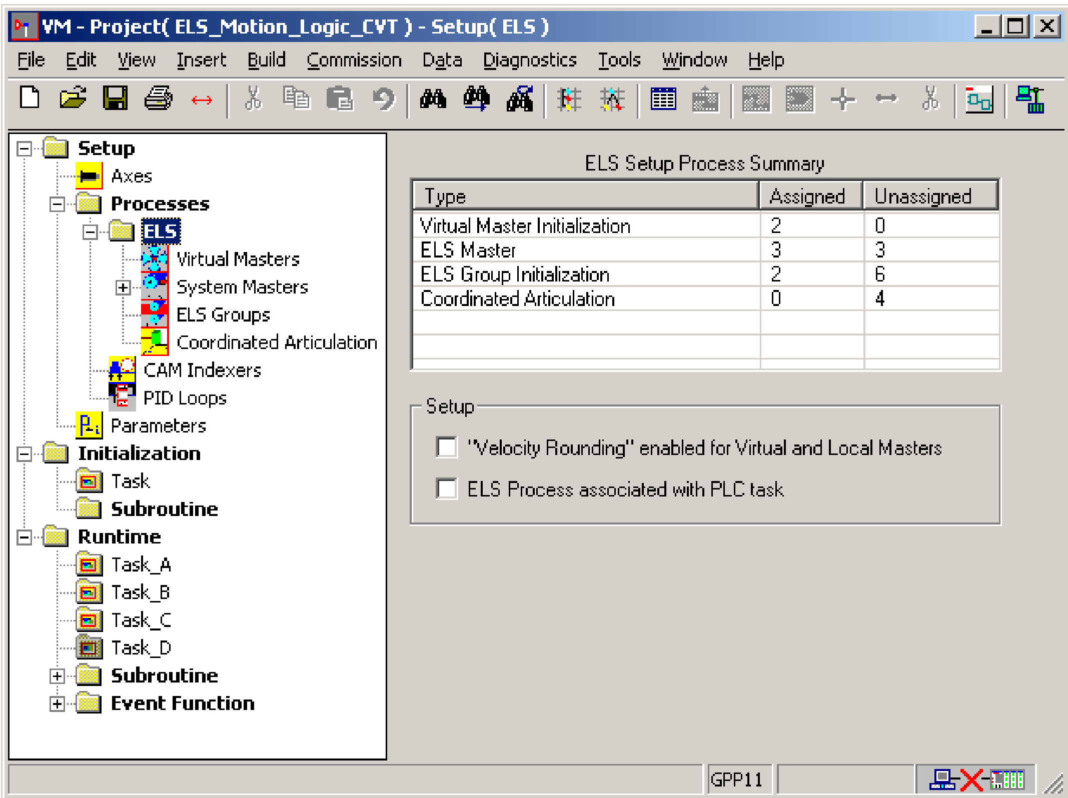
"Velocity Rounding" enabled for Virtual Master and Local Master

Velocity Rounding sets the Virtual Master and Group jogging velocities down to the nearest ELS increment to eliminate cycle-to-cycle variations in drive velocity. This results in the velocities being slightly less than their commanded values. This feature can be enabled and disabled in bit 30 (ELS_MSTR_CONFIG) of the ELS Group Configuration Word.

ELS Process associated with the PLC task

Setting this option allows the Integrated PLC to control the running of the ELS system. If the Integrated PLC program stop running, so will the ELS system.

The right window also displays a summary of the assigned and unassigned ELS Setup Process.



SetupELS.tif

Fig. 4-31: Setup ELS

Modifying ELS Program Variables

During the initial setup of ELS processes, all necessary registers and program variables are declared and assigned values. This includes all initialization and runtime variables. ELS process variables can be modified and fine-tuned to meet specific application requirements by using one of the following methods:

Procedure	Description
Icon Program	ELS icons can be used in the Initialization task to modify existing an ELS process.
ELS Runtime Tool	The ELS Runtime tool can be used in online mode to modify ELS processes.
Direct Access to Program Variables	Write directly to ELS program variables from the icon program or PLC project.
PLC Function Blocks	Using Virtual Master and ELS Group function blocks from the MV_Motion library in the PLC program.

Table 4-13: Modifying ELS Process Variables

Icon Initialization of ELS Program Variables

ELS icons can be used in the Initialization task to modify program variable values initially declared under Setup. This method of programming provides the user with the flexibility of creating multiple conditions or scenarios for initializing ELS program variables.

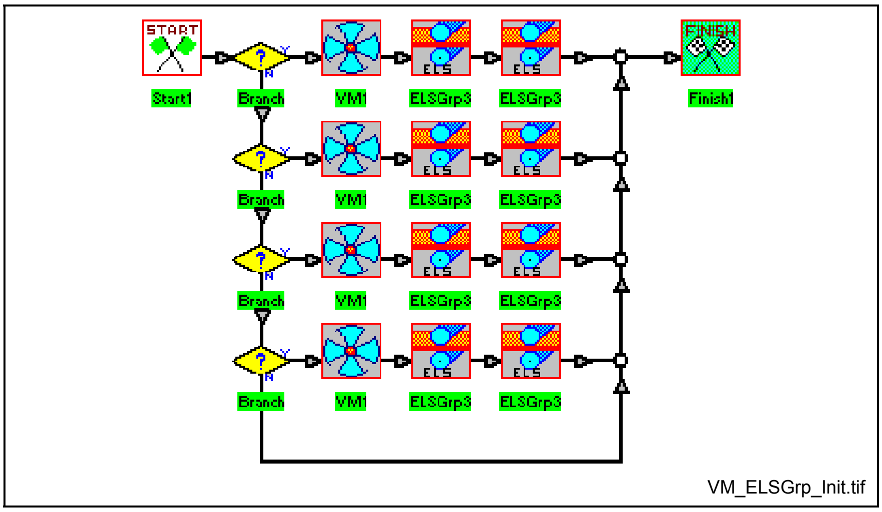


Fig. 4-32: Conditional Initialization of VM1 and ELSGrp3 Icons Example

Refer to the section 6.1, *Electronic Line Shafting*, for a complete listing of ELS system variables.

Direct Access to ELS Program Variables

ELS program variables can be written to directly from the icon program, using the Calc2 icon or from the PLC program using access methods such as, SysLibDirect, I/O Image, Functions, and/or Function Blocks.

Function Block Initialization of ELS Program Variables

Virtual Master and ELS Group program variables can be declared using the following MV_Motion function blocks in the PLC program:

Function Block	Description
MV_VmSetupMaxVelocity	sets up the maximum velocity, acceleration and deceleration values and the E-stop deceleration value for the selected Virtual Master or ELS Group Local Master
MV_VmSetupPositionData	sets up the shortest path positioning window and absolute positioning mode for the selected master
MV_ElsGroupSetupSynch	sets up synchronize configuration for ELS Group masters
MV_ElsGroupSetupEquation	sets up input values for determining ELS Group output values
MV_ElsGroupSetupLockCams	sets up CAM numbers and associated scaling factors for the Lock On/Lock Off functionality

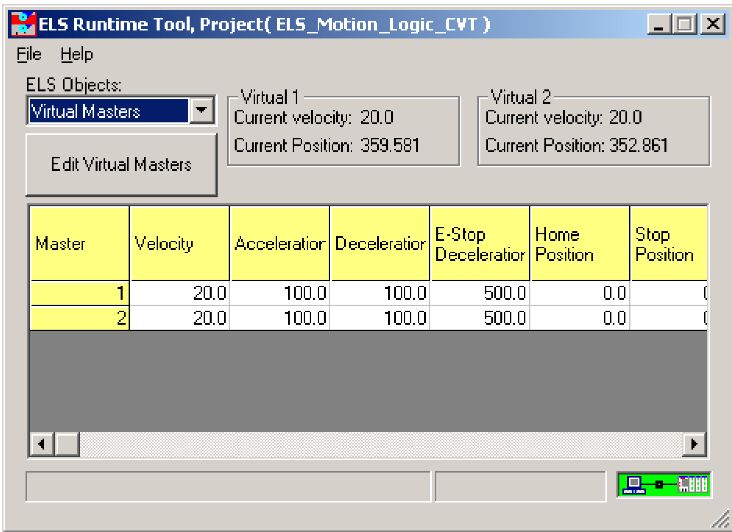
Table 4-14: ELS Virtual Master and ELS Group Function Blocks

Refer to the IndraLogic Help system under contents heading *Libraries ⇒ Rexroth Firmware Libraries ⇒ MV_Motion Library* for detail descriptions.

ELS Runtime Modification of ELS Program Variables

After a project is compiled and downloaded to the control, the ELS Runtime tool can be used in online mode to modify ELS program variables for the project.

All ELS program variables configured using the ELS Runtime tool are saved with the project. Using the ELS Runtime tool in service mode will only modify the data on the control's memory and not in the project files.



ELSRuntimeTool.tif

Fig. 4-33: ELS Runtime Tool

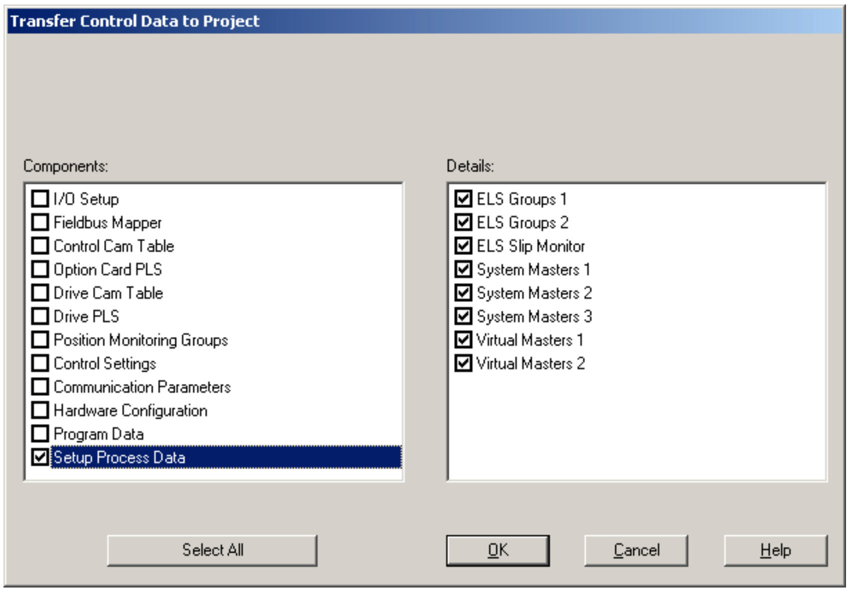
The ELS Runtime tool is located under menu selection ***Data ⇒ ELS…*** in VisualMotion Toolkit. Refer to section 8.3, *ELS Runtime Tool*, for details.

Importing Modified Project Data

ELS program variables that are modified on the control can be imported into the project in online mode. Importing modified ELS program variables from the control will update the current ELS process configurations under Setup.

Use the following steps to import ELS program variables from the control to the project:

1. Launch VisualMotion Toolkit with the relevant project opened.
2. Switch to Online mode by selection ***File*** ⇒ ***Online***, pressing F9 or clicking on the Online icon.
3. Select ***File*** ⇒ ***Import Project Components …***
4. From the *Transfer Control Data to Project* window place a check in the **Setup Process Data** checkbox.



ImportDataToProject.tif

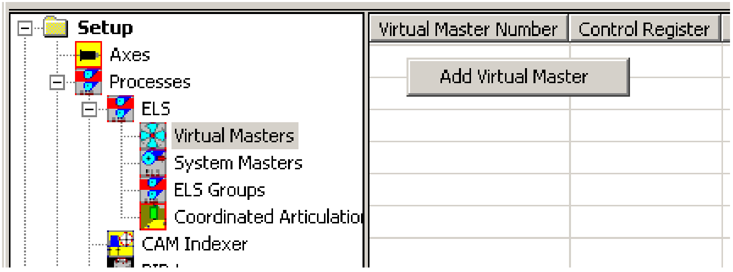
Fig. 4-34: Transferring ELS Data from the Control to the Project

5. By default, all the items in the right window (Details) are selected.
6. Select only the ELS Details that will imported into the project.
7. Edit the relevant ELS process under Setup to see the modified values.

Note: The modified values are now part of the VisualMotion project. These new values will appear under the Setup section and will be used to initialize the ELS system the next time the project is downloaded to the control.

Virtual Masters Setup

The Virtual Masters setup is used to configure up to 2 Virtual Master for an ELS system. During the initial configuration of a Virtual Master, a wizard is used to assists the assignment of control and status registers, default program variable (float and integer) labels, as well as initial, maximum and positioning values.



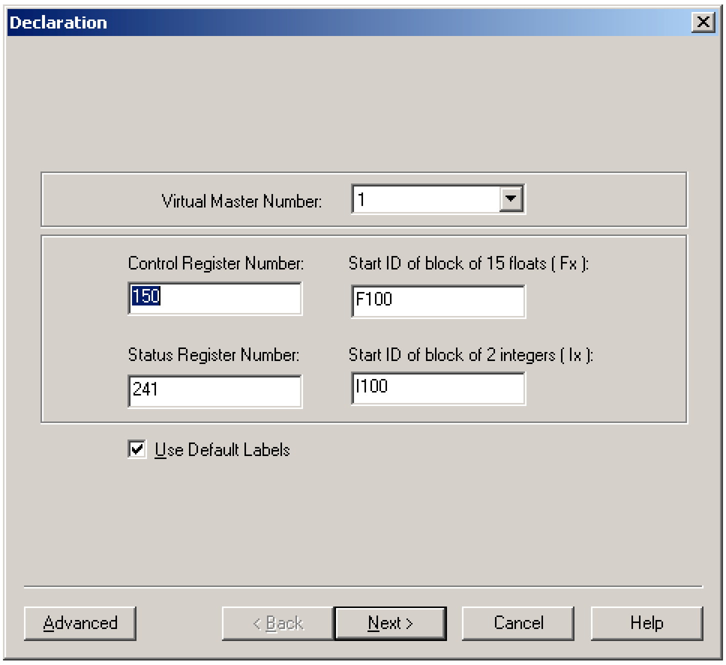
SelectVirtualMasterSetup.tif

Fig. 4-35: Virtual Master Setup

Virtual Master Declaration

To declare a Virtual Master, right click on an empty row and select ***Add Virtual Master***. The wizard will guide to user through the setup process.

The Virtual Master *Declaration* window displays the default register values and start ID blocks for the floats and integers for the selected Virtual Master number. 15 floats and 2 integers are reserved for each Virtual Master.



VmDataSetup.tif

Fig. 4-36: Virtual Master Declaration

Note: By default, a Virtual Master is assigned to Task A unless it is associated with the PLC task. All control and monitoring of the Virtual Master is performed by the control and status registers of Task A. All ELS Group axes that are Sync to Master and following the Virtual Master will stop if Task A is stopped.

Rexroth Bosch Group

Use Default Labels

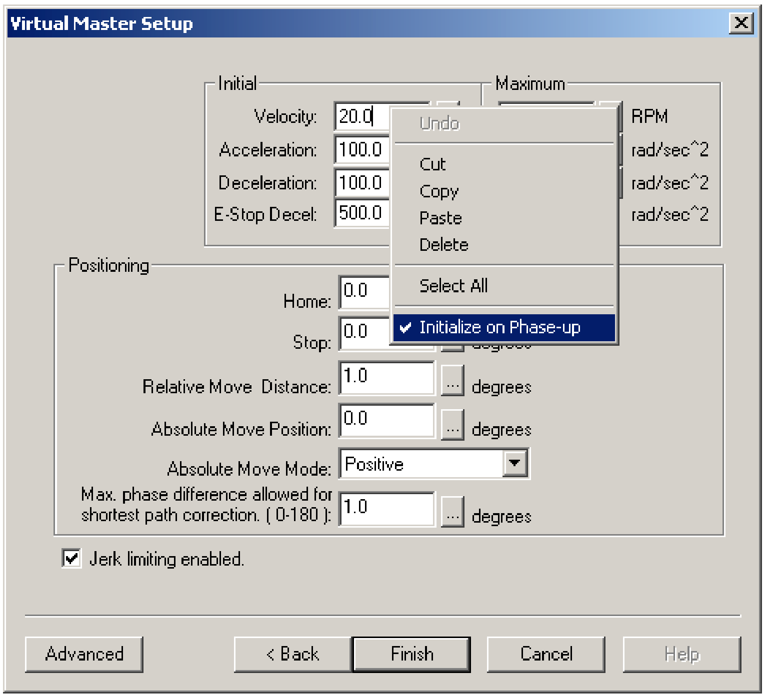
Default variable labels and comments can be added by simply leaving the *Use Default Labels* checkbox checked. This is the default setting.

All necessary labels for registers, register bits, and program variables will begin from the float and integer start id block entered in the *Declaration* window. Refer to Use Default Labels on page 4-25 for details.

For default register and variable label values, refer to section 6.1, *Electronic Line Shafting*, for details.

Virtual Master Setup

The Virtual Master Setup is used to configure the Initial, Maximum, and Positioning values for the selected Virtual Master. Refer to *Virtual Master* in section 6.1 for details of each field.



VirtualMasterSetup.tif

Fig. 4-37: Virtual Master Setup

Selecting Program Variables to Retain Data Value

Virtual Master Setup values can be selected to retain their last value during a system re-initialization or system power down. Refer to Defining Process Variables to Retain Values on page 4-26 for details.

Edit or Delete a Virtual Master

Once configured, a Virtual Master can be edited or deleted by right clicking over the Virtual Master Number and selecting the relevant option. They can also be edited by double clicking over a Virtual Master.

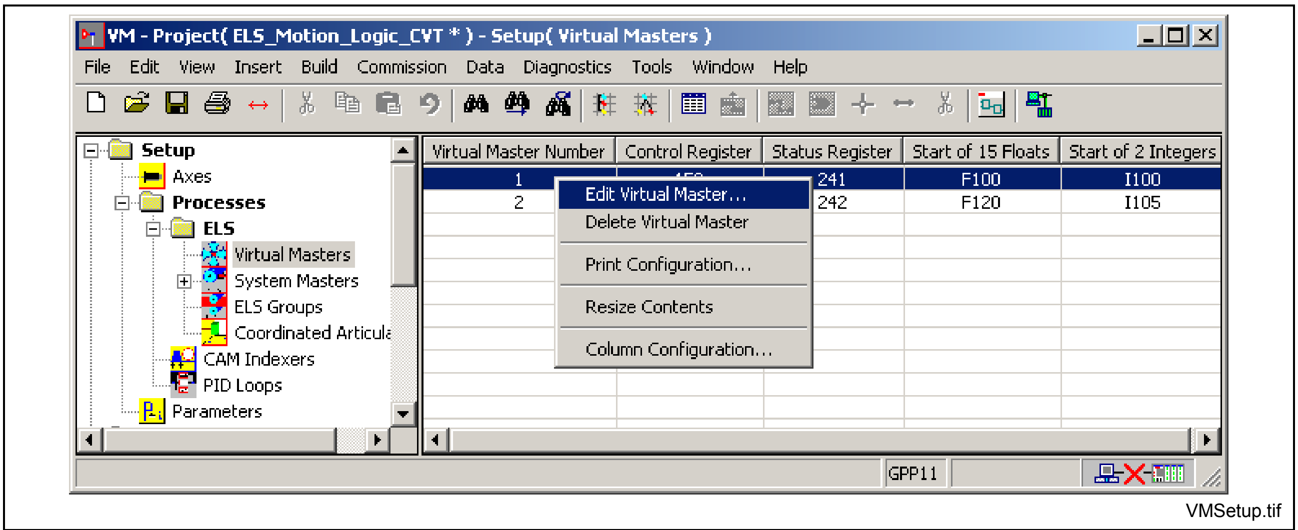


Fig. 4-38: Virtual Masters Setup

When editing a configured Virtual Master, the data is displayed in windows containing tabs for either *Declaration* or *Virtual Master Setup*. Select the relevant tab and modify the necessary data. Modifications made to a Virtual Master setup require that the project be recompiled and downloaded to the control.

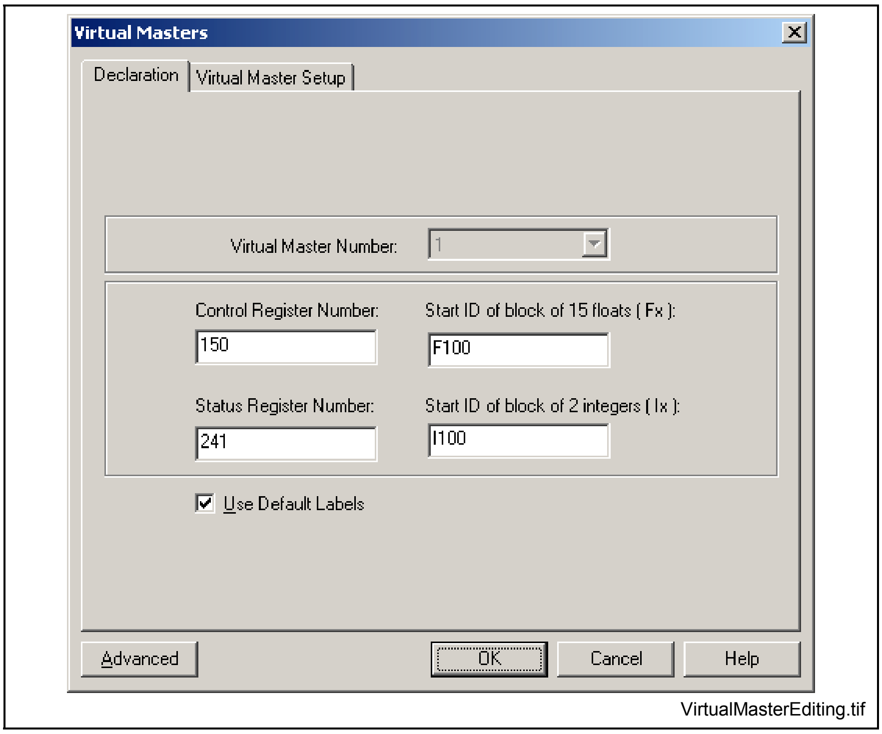
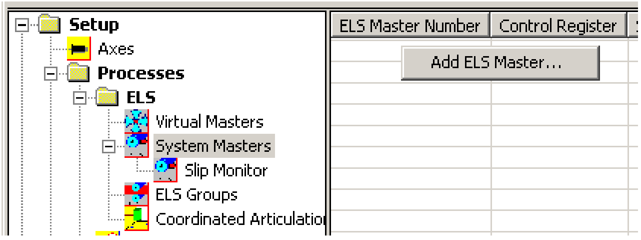


Fig. 4-39: Virtual Master Editing

System Masters Setup

The ELS Master Setup is used to configure initialization data for up to 6 ELS System Masters.



SelectELSMaster.tif

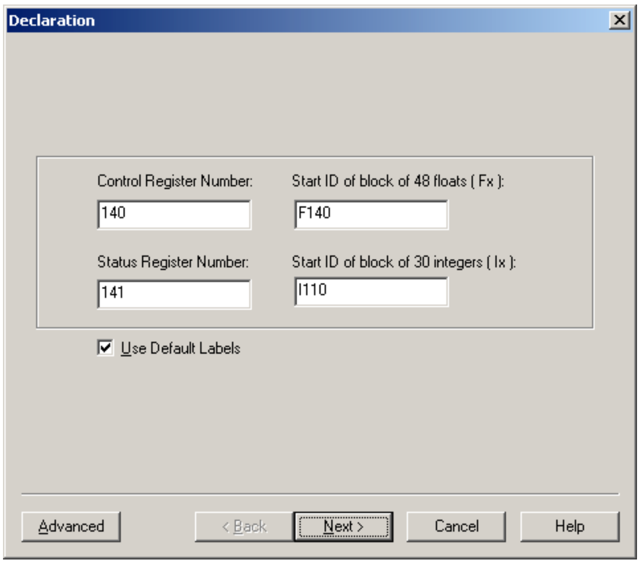
Fig. 4-40: System Master Setup

System Master Declaration

To declare a System Master, right click on an empty row and select ***Add ELS Master***. The wizard will guide to user through the setup process.

Note: Unlike the other ELS processes that require register and variable declarations for each individual configuration, the first ELS System Master that is added declares all the required registers and variables for all 6 ELS Masters.

The ELS Master *Declaration* window displays the default register values and start ID blocks for the floats and integers for the selected ELS Master number. 48 floats and 30 integers are reserved for all 6 ELS Masters.



ELSMasterDataSetup.tif

Fig. 4-41: System Master Declaration

Note: By default, an ELS System Master is assigned to Task A unless it is associated with the PLC task. All control and monitoring of ELS System Masters is performed by the control and status registers of Task A. All ELS Group axes that are Sync to Master and following an ELS System Master will stop if Task A is stopped.

Use Default ELS Master Labels

Default variable labels and comments can be added by simply leaving the *Use Default Labels* checkbox checked. This is the default setting.

All necessary labels for registers, register bits, and program variables will begin from the float and integer start id block entered in the *Declaration* window. Refer to Use Default Labels on page 4-25 for details.

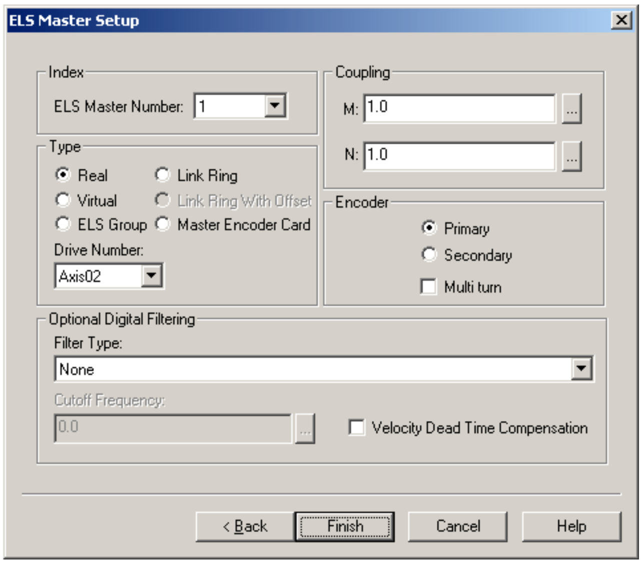
Note: Default labels are added for all 6 ELS System Master whether or not all 6 are configured in a project.

For default register and variable label values, refer to section 6.1, *Electronic Line Shafting*, for details.

ELS Master Setup

The ELS Master Setup is used to configure the master Type along with any relevant settings for that given type. Refer to the relevant *ELS Master* type in section 6.1 for details.

Note: The ELS Master Setup window does not contain an **Advanced** button. Unlike the other ELS processes, all ELS Master data is require to be re-initialized to a know value.



ELSMasterSetup.tif

Fig. 4-42: ELS Master Setup

Index

The *ELS Master Number* is a unique number assigned to the ELS Master currently being configured. Up to 6 ELS System Master can be setup in a VisualMotion project.

Type

Real A Real Master is either a primary (motor) or secondary encoder (position feedback) from a drive. Each drive in the system can potentially provide two Real Masters. The raw position value of the Real Master can be filtered and geared by a M/N ratio. A maximum of three Real Masters can be assigned.

Note: A Real Master can be primary encoder that are not a slave of an ELS Group or secondary encoder.

Virtual A Virtual Master is an internal motion engine with an independent set of control parameters. A maximum if two Virtual Masters can be assigned. Each Virtual Master can be used independently from the other. A Virtual Master is controlled by VisualMotion and/or a PLC using I/O registers and program variables.

Note: Virtual Masters are initialized using the Virtual Master icon before they are assigned a number.

ELS Group An ELS Group Master is the output of an ELS Group that can be used as an input master signal to a different ELS Group.

Link Ring This option sets the selected ELS System Master to receive the master position of the *External Number* Link Ring node.

Master Encoder Card The Master Encoder Card (DAG) card can also provide up to two feedback signals from a GSD and/or a 1 V peak-to-peak encoder.

Coupling (Integer) (for Real Master only)

The M/N ratio is only available for Real Master outputs. The positional value for a Real Master output is multiplied by the M/N ratio and/or a Digital Filter and used as a new position value for an ELS Group input.

Example:

If a Real Master position is at 180° and a M/N value of 2/1 is used, then the ELS Group input will follow a position value of 360°.

If a Real Master position is at 180° and a M/N value of 1/2 is used, then the ELS Group input will follow a position value of 90°.

Encoder

Select the encoder type as either a **Primary** or a **Secondary** encoder. If the encoder type selected is an absolute or multi-turn encoder, place a check in the **Multi turn** box.

Optional Digital Filtering

Digital filtering is available for Real Masters and PID loops.

Filter Type:

None

First order low-pass, $G(s)=1/(s+1)$

Second order low-pass, $G(s)=1/(s^2 +2s +1)$

Third order low-pass, $G(s)=1/(s^3+3s^2+3s+1)$

Second order Butterworth, $G(s)=1/(s^2 + 2^{1/2}s + 1)$

Third order Butterworth, $G(s)=1/(s^3+2s^2+2s+1)$

Modified 2nd order low-pass with velocity ramp tracking,

$$G(s)=(2s+1)/(s^2 + 2s + 1)$$

Modified 3rd order low-pass with accel ramp tracking,

$$G(s)=(3s^2+3s+1)/(s^3+3s^2+3s+1)$$

Cutoff Frequency (float): When a filter type is chosen, a cutoff frequency for the filter must be entered. The cutoff frequency is the frequency where the signal is reduced by 3db. When set to 0 the filter is disabled.

To ensure a stable system, use the following calculation when entering a value for the Digital Filter Cutoff Frequency:

$$\text{Cutoff Frequency} \leq \frac{1}{2 * \text{Sampling Rate(sec.)}}$$

The sampling rate for a drive axis is the set phase 4 Sercos cycle time (S-0-0002), entered in seconds.

Example: For a 2 ms Sercos cycle time, the cutoff frequency is calculated as follows:

$$\text{Cutoff Frequency} \leq \frac{1}{2 * 0.002} = 250 \text{ Hz}$$

The following figure illustrates the frequency vs. degrees for each filter. The cutoff frequency is 10 hertz and the sampling rate is 4 ms.

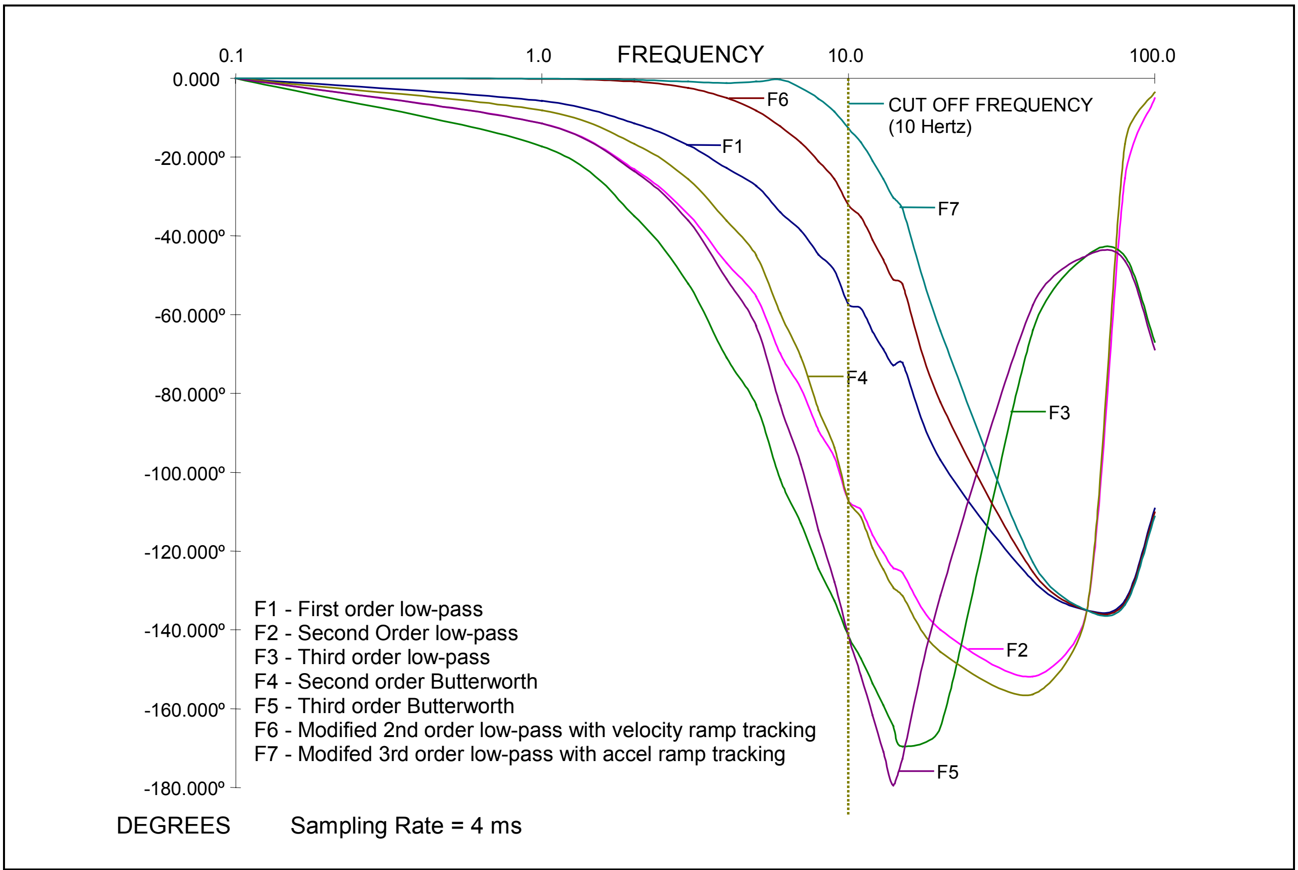


Fig. 4-43: Frequency vs. Degree Filter Chart

The following figure illustrates the gain vs. frequency for each filter. The cutoff frequency is 10 hertz and the sampling rate is 4 ms.

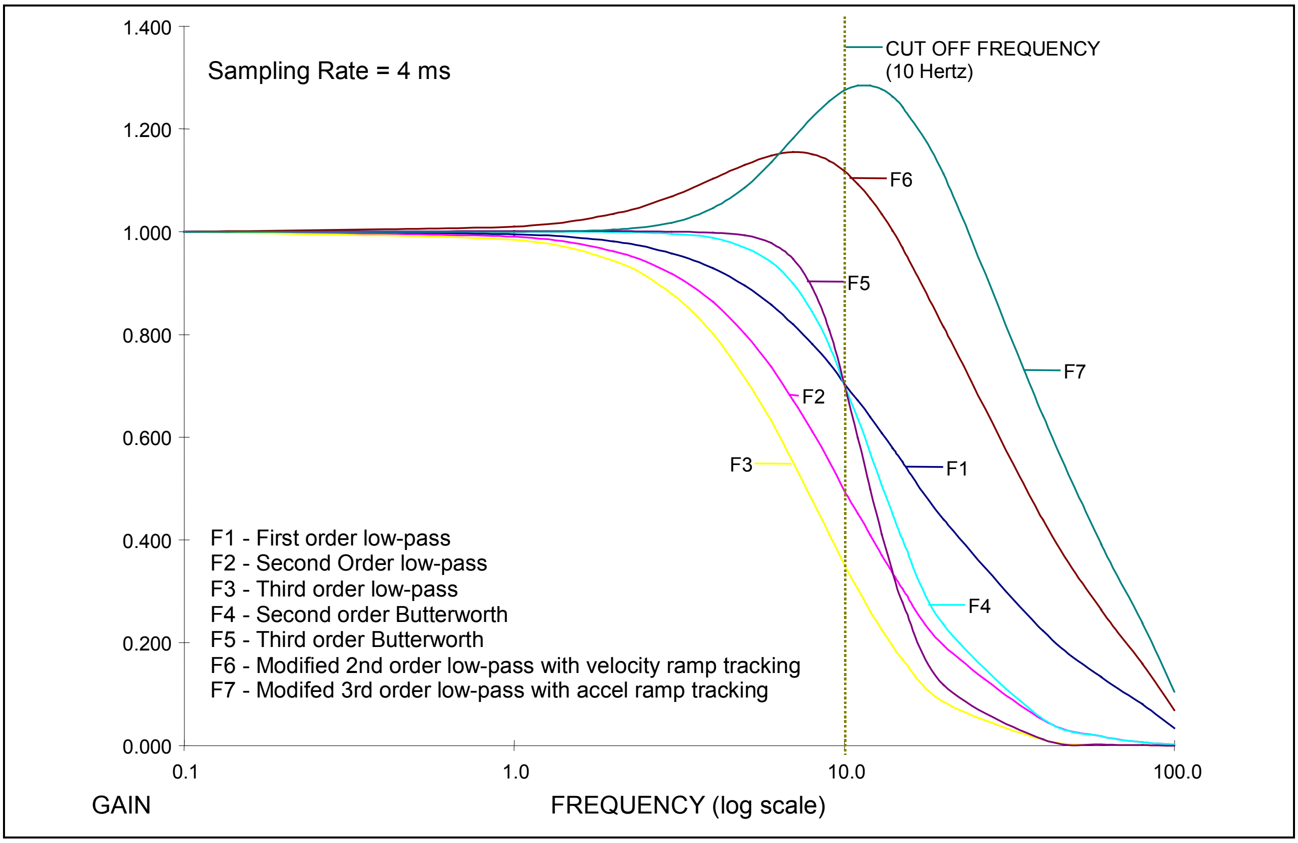


Fig. 4-44: Gain vs. Frequency Filter Chart

When a filter is chosen, the cutoff frequency for the filter must be entered. For example, the cutoff frequency for the First order low-pass filter is the frequency where the signal is reduced 3db [.707 gain, db=20*log(gain)].

Velocity Dead Time Compensation

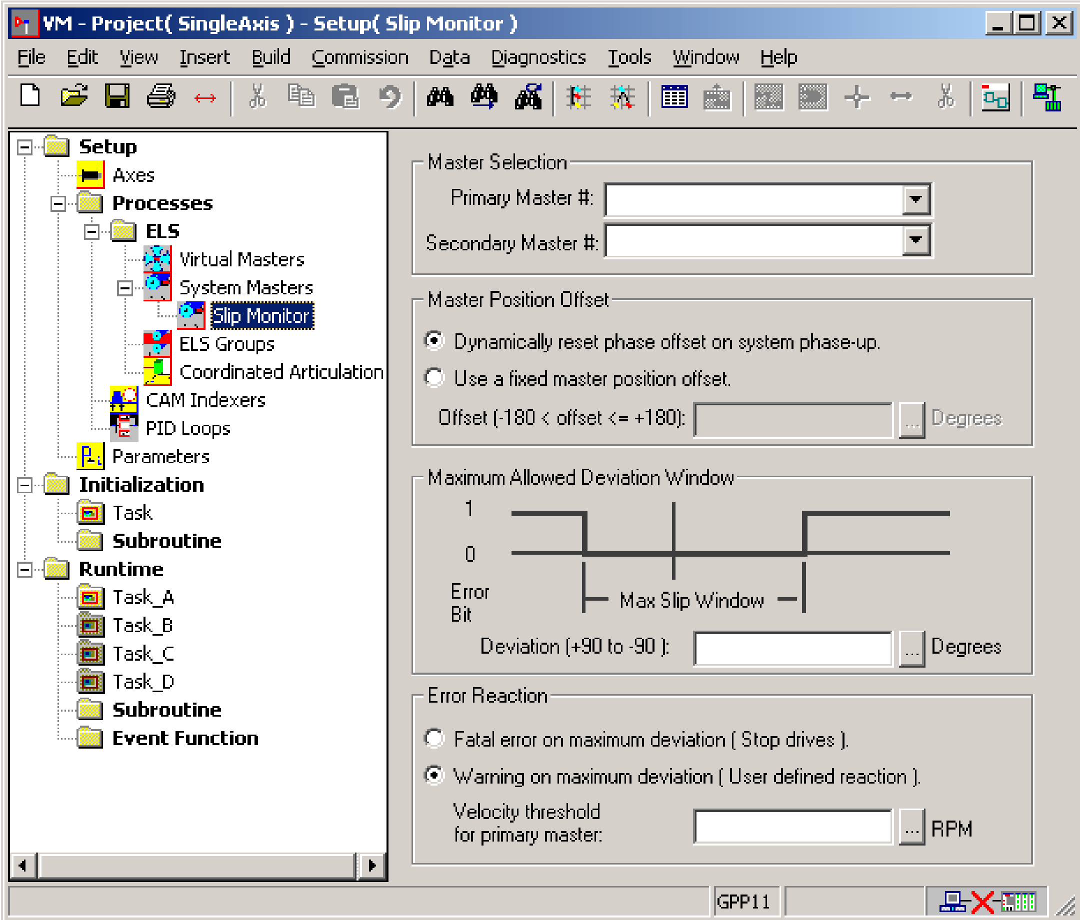
Velocity Dead Time Compensation becomes active when a digital filter is applied to a Real Master. This feature allows the user to add 4 Sercos cycles of velocity feed forward phase advance to an ELS Real Master to compensate for delays in control processing. The phase advance is performed where the Real Master position data is brought into the ELS System Masters such that the ELS System Master output user variables will reflect the phase advance. Since ELS propagates information from the Masters to the Groups via position and velocity (primarily by velocity), both the Real Master’s position and velocity are advanced.

The Dead Time Compensation can be disabled for individual Real Masters for applications in which undershoot/overshoot during velocity changes could cause problems. The functionality of the feature is such that compensation is enabled by default. Users will need to disable it if they do not want to use it with their application.

Note: Dead Time Compensation ONLY compensates for the phase lag created by the 4 cycles of ELS processing / Sercos delays. It DOES NOT compensate for additional dead time incurred by the various Real Master filters.

Slip Monitor

The Slip Monitor feature is used to compare the position signal of two ELS System Masters. Initially, the slip monitor is configured under Setup but can be modified in the Initialization task by using the SlipMon icon. Refer to Slip Monitoring for ELS Masters in section 6.1, *Electronic Line Shafting*.



SlipMonitor.tif

Fig. 4-45: ELS Master Slip Monitor

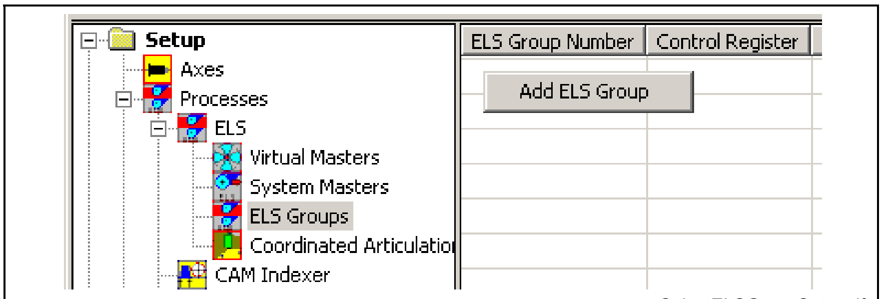
Edit or Delete an System Master

Once configured, a System Master can be edited or deleted by right clicking over the System Master Number and selecting to appropriated option. They can also be edited by double clicking over a System Master. Refer to Edit or Delete a Virtual Master on page 4-35 for an example.

ELS Group Setup

The ELS Group setup is used to assign control and status registers, variable (float and integer) start ID blocks for up to 8 ELS Groups, CAM and Initialization options.

Note: Axes used in an ELS Group are setup under ***Setup*** ⇒ ***Axes***. Refer to ELS Group Axis Setup on page 4-22.



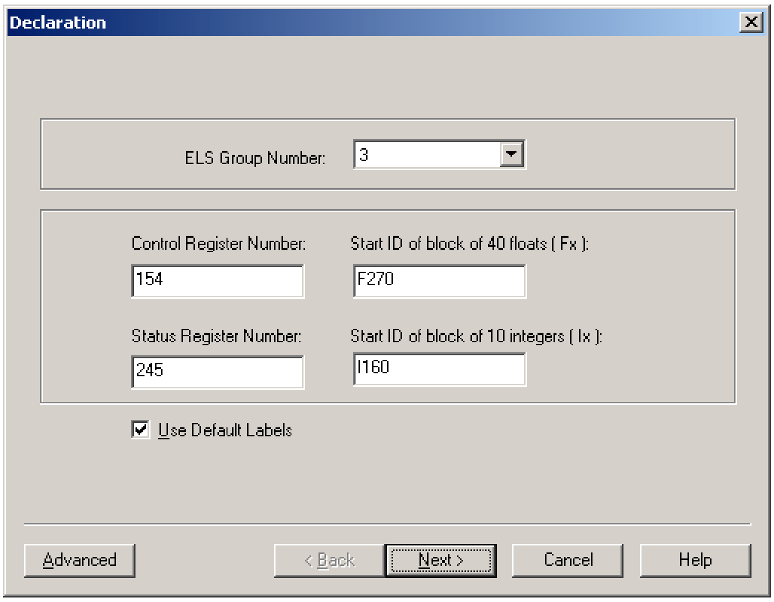
SelectELSGroupSetup.tif

Fig. 4-46: ELS Group Setup

ELS Group Declaration

To declare an ELS Group, right click on an empty row and select ***Add ELS Group***. The wizard will guide to user through the setup process.

The ELS Group *Declaration* window displaying the default register values and start ID blocks for the floats and integers for the selected ELS Group number. 40 floats and 10 integers are reserved for each ELS Group.



ElsGroupDataSetup.tif

Fig. 4-47: ELS Group Declaration

Note: By default, an ELS Group is assigned to Task A unless it is associated with the PLC task. All control and monitoring for an ELS Group is performed by the control and status registers of Task A. All ELS Group axes that are Sync to Master and following the Virtual Master will stop if Task A is stopped.

Use Default ELS Group Labels

Default variable labels and comments can be added by simply leaving the *Use Default Labels* checkbox checked. This is the default setting.

All necessary labels for registers, register bits, and program variables will begin from the float and integer start id block entered in the *Declaration* window. Refer to Use Default Labels on page 4-25 for details.

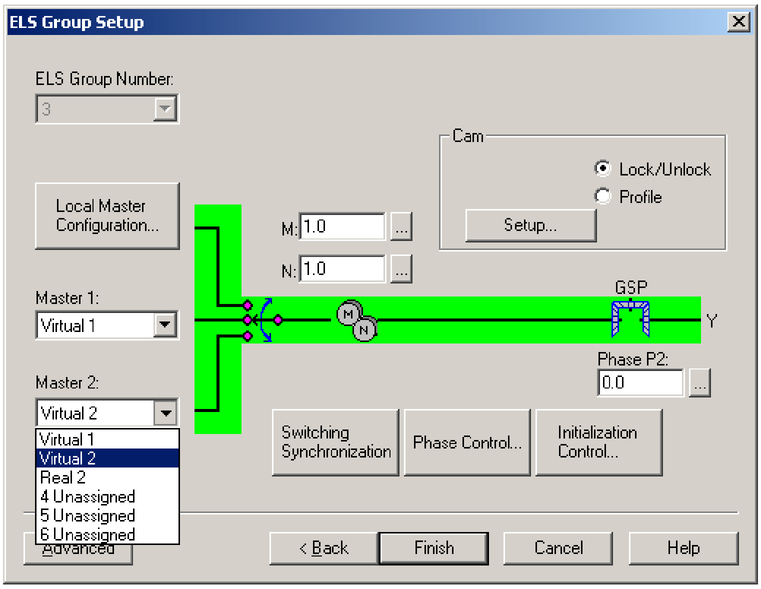
For default register and variable label values, refer to section 6.1, *Electronic Line Shafting*, for details.

ELS Group Setup

The ELS Group Setup is used to configure the following initialization data:

- ELS Group Number Selection
- Local Master Configuration button
- Master Selection
- CAM Selection
- Switching Synchronization button
- Phase Control button
- Initialization Control

Clicking the browse ... button to the right of M, N, H, Phase P1 or Phase P2 allows selection of a variable from the VM Data Table.



ElsGroupSetup.tif

Fig. 4-48: ELS Group Setup

ELS Group Number:

Select an ELS Group number from the drop-down list. Only ELS Groups configured in the Setup will be available for selection.

Local Master Configuration

The *Local Master Configuration* is used to set the Initial, Maximum and Positioning values for the ELS Group. For specific information on how to setup the Local Master Configuration, refer to section 6.1 in volume 1 of the *Rexroth VisualMotion Functional Description.*

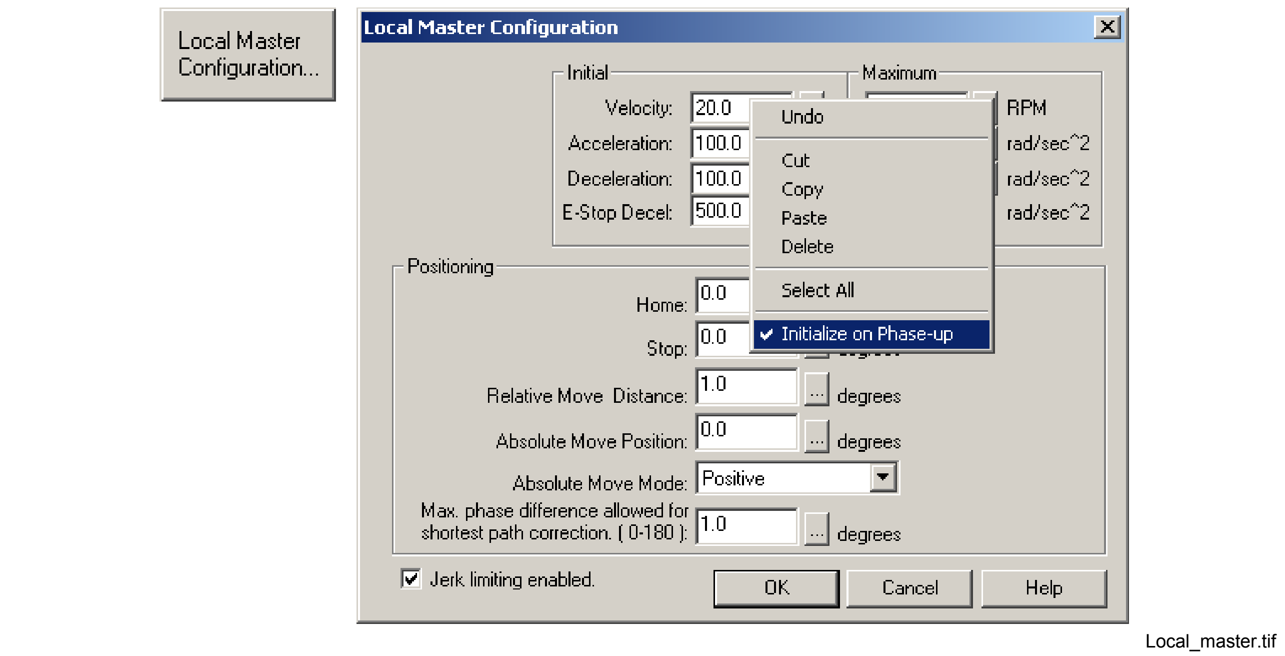


Fig. 4-49: Local Master Configuration

Initialize on Phase-up

Certain Local Master Configuration values can be selected to retain their last value during a system re-initialization or system power down. Refer to Defining Process Variables to Retain Values on page 4-26 for details.

Master Selection

The *Master 1* and *Master 2* drop-down list is used to set the ELS System Master type that will be used as the input master for the current ELS Group. The available master types are dependent on the *ELS Masters* configured under **Setup**. Refer to System Masters Setup on page 4-36 for details.

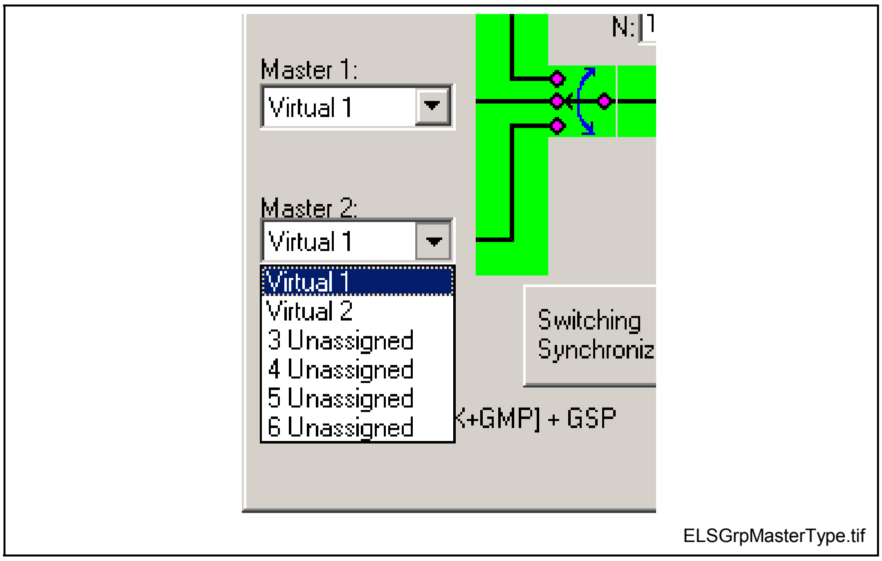
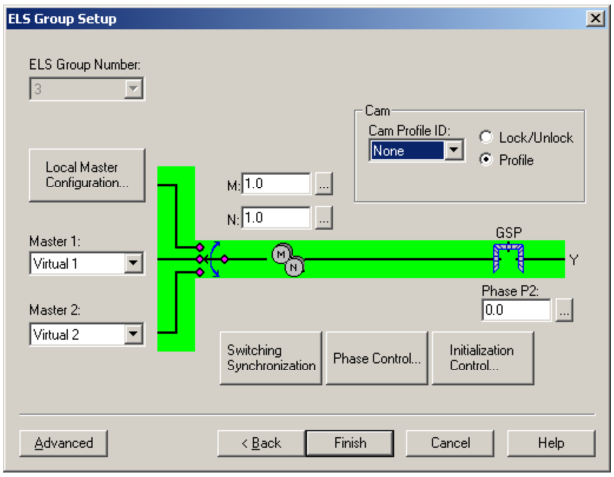
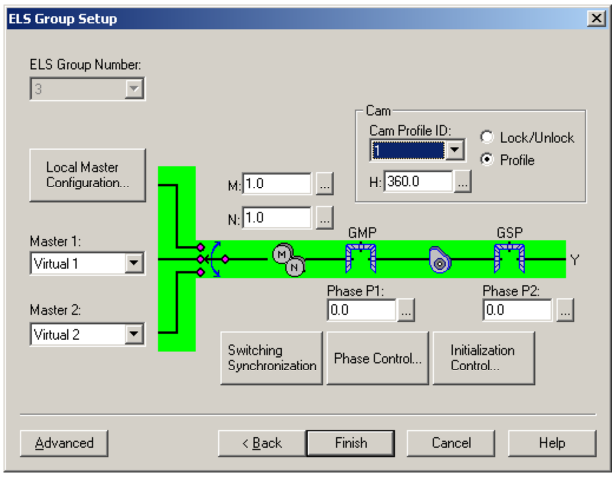


Fig. 4-50: ELS Group Master Type

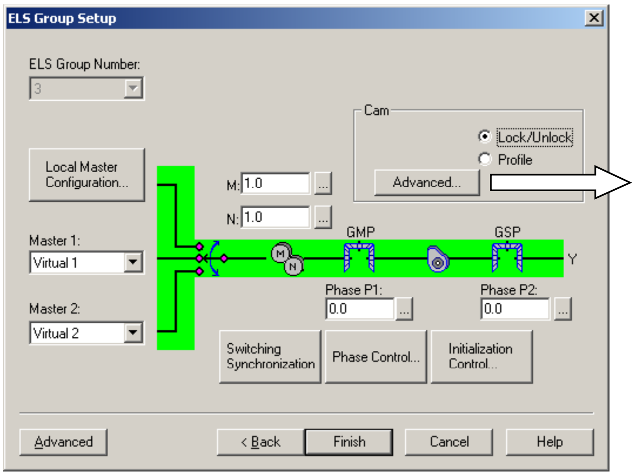
CAM Section The options available in the CAM section depend on the radio button that is selected. The available radio buttons are Lock/Unlock and Profile.



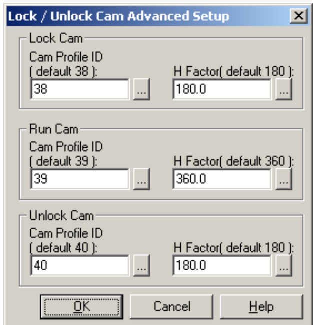
CAM Profile ID = None
ELS Group output will not have a GMP (relative Master phase) or CAM profile applied.



CAM profile ID = CAM number
ELS Group output can have a GMP (relative Master phase) or CAM profile applied.



CAM = Lock/Unlock
ELS Group output can have a GMP (relative Master phase) and default Lock / Unlock CAMs applied.



Advanced…
Displays default Lock / Run / Unlock CAMs (38, 39 and 40). User defined CAMs can be selected.

Fig. 4-51: ELS Group CAM Section

Rexroth Bosch Group

Synchronization Setup

For specific information on how to setup synchronization, refer to section 6.1 in volume 1 of the *Rexroth VisualMotion Functional Description.*

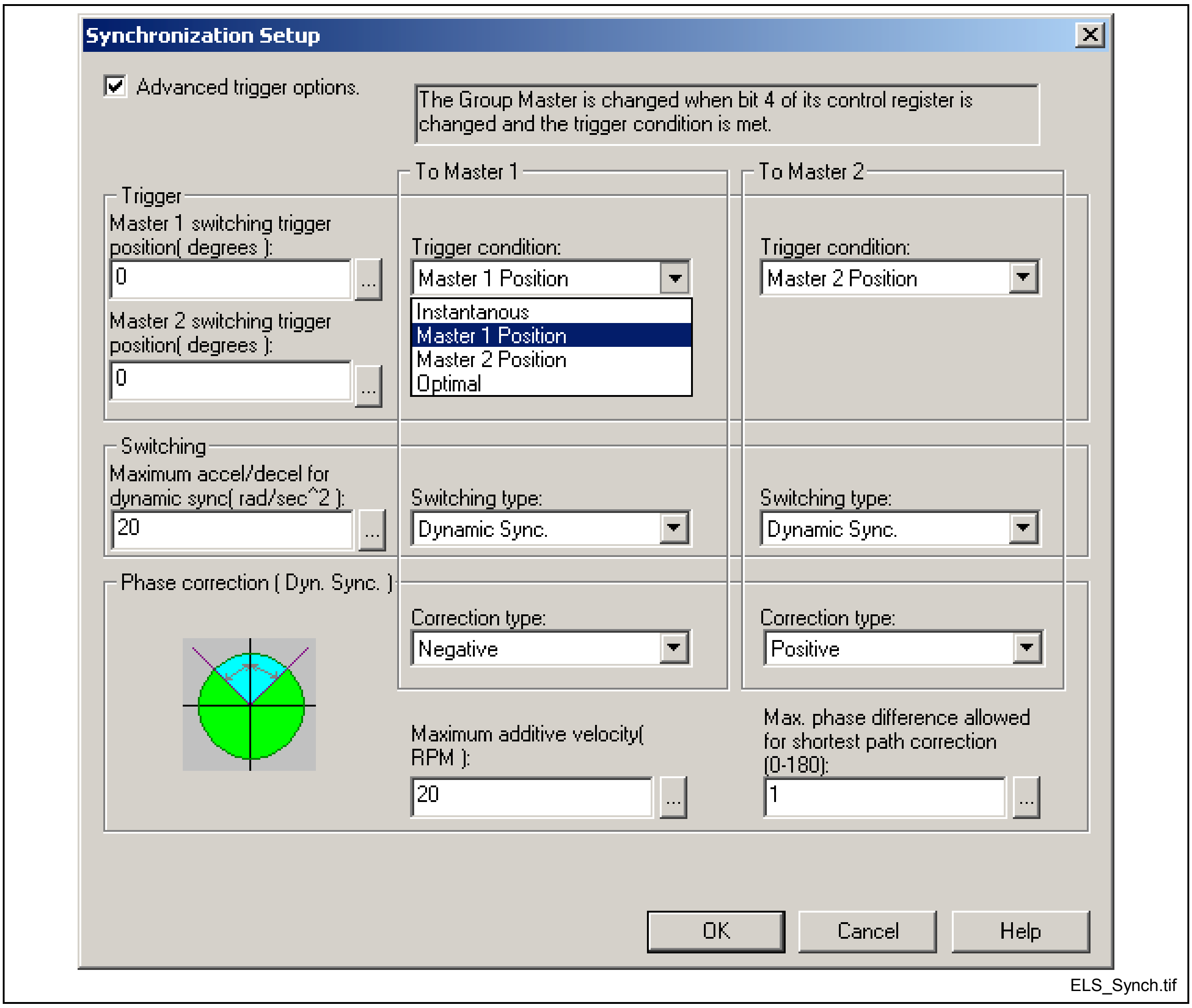


Fig. 4-52: ELS Synchronization Setup

Phase Control

For specific information on how to setup phase control, refer to section 6.1 in volume 1 of the *Rexroth VisualMotion Functional Description.*

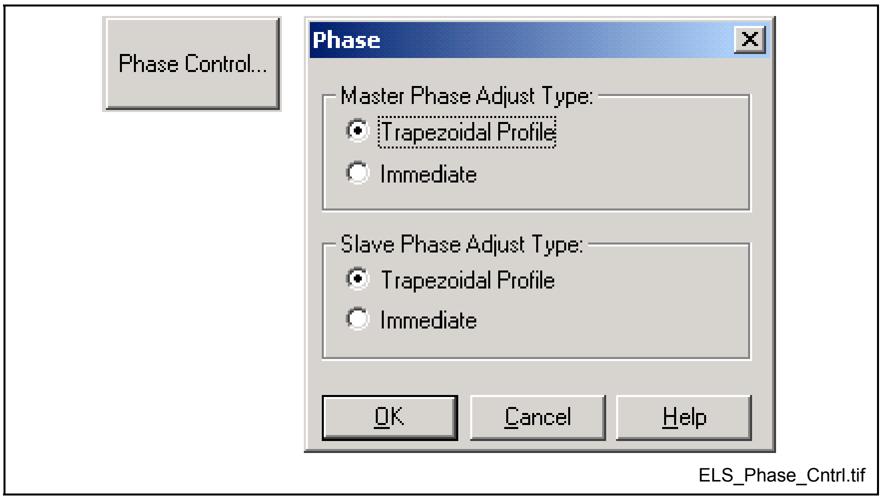
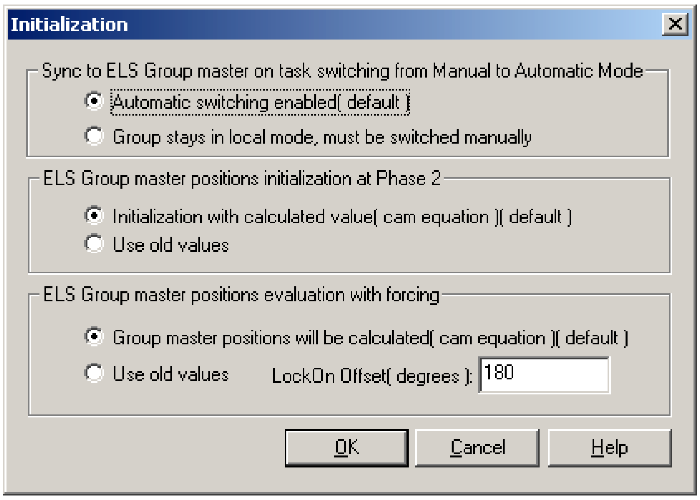


Fig. 4-53: ELS Phase Control

ELS Group Initialization

The *ELS Group Initialization* window is used to select the ELS group master operation initialization options.



ELSGroupInit.tif

Fig. 4-54: ELS Group Initialization

ELS Group Master Operation Initialization

Sync to ELS Group master on task switching from Manual to Automatic Mode

This can be set at automatic or you can switch the group switch the ELS group to the master manually. The group will remain in local mode when the project comes out of parameter mode until it is manually switched.

ELS Group master positions initialization at Phase 2

During phase 2 of power up, you can select to use either old values or one of the following:

- the variables initialized by the internal group master being set to the value of the active group input master
- the cam table input position set to the internal group input master position plus the group master offset
- the state of the state machine set to 1
- the old values kept for absolute master and group slave offset
- the ELS group master position calculated with the cam equation

ELS Group master positions evaluation with forcing

In phase 4 of power up, the ELS group can be reinitialized when the local mode is active and the ELS group master is at standstill (G#_ST_MOTION is 0). Under these conditions, the following variables are not updated by the control and can be overwritten:

- Internal group input master position
- Group cam table input position (used to calculate the ELS group master position)
- ELS group master position (only when bit 9 in the ELS configuration word is set to 1)
- State of the state machine for lock on/lock off (used to calculate the ELS group master position)

Edit or Delete an ELS Group

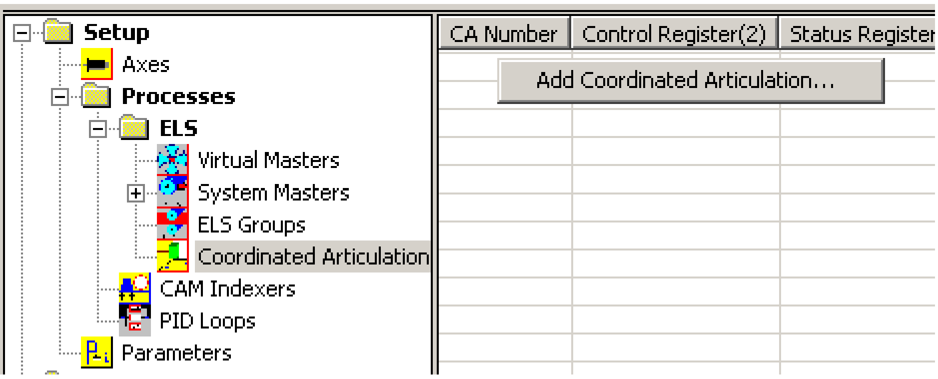
Once configured, an ELS Group can be edited or deleted by right clicking over the ELS Group Number and selecting to relevant option. They can also be edited by double clicking over an ELS Group. Refer to Edit or Delete a Virtual Master on page 4-35 for an example.

Coordinated Articulation Setup

Coordinated Articulation is an advanced feature in VisualMotion 11 used to move up to six axes in coordinated fashion based on world coordinate inputs from an ELS Group output or manual positions. This feature provides the ability to link cyclic coordinated motion to an ELS master.

The Coordinated Articulation Setup is used to assign the control and status registers, enter predefined Kinematic equation values, and the variables (floats and integers) start ID blocks for up to 4 Coordinated Articulation configurations.

Note: Axes used in a Coordinated Articulation configuration are setup under ***Setup ⇒ Axes***. Refer to Coordinated Articulation Axis Setup on page 4-23.



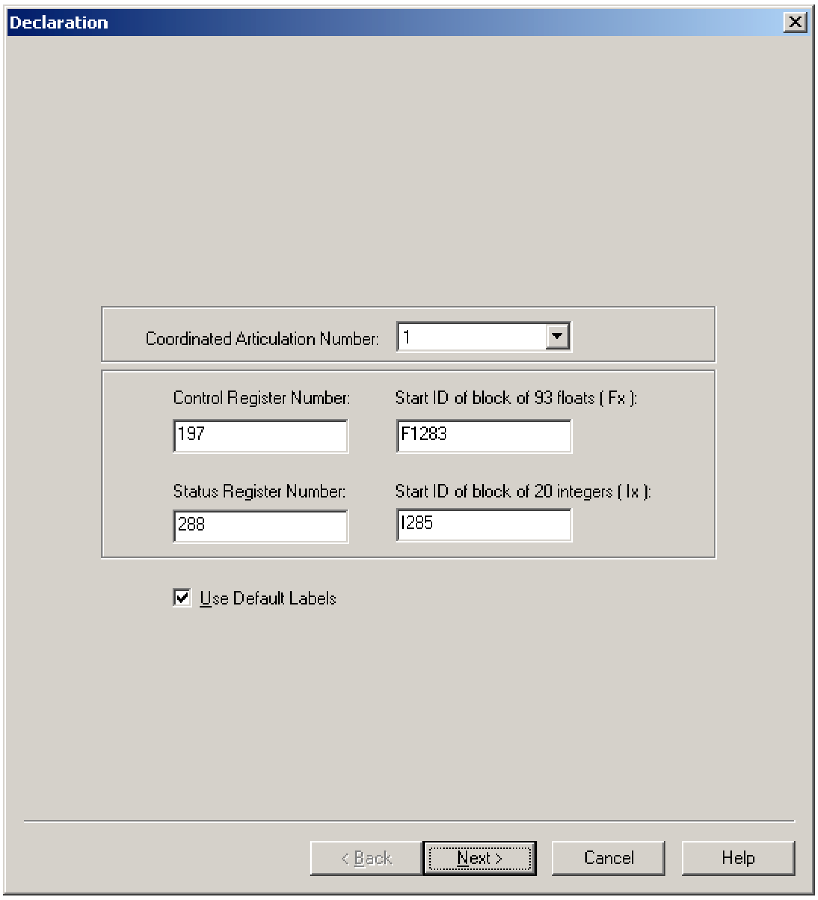
SelectCoordArtSetup.tif

Fig. 4-55: Coordinated Articulation Setup

Coordinated Articulation Declaration

To declare a Coordinated Articulation configuration, right click on an empty row and select ***Add Coordinated Articulation…***

This opens the *Coordinated Articulation Data* window displaying the default register values and start ID blocks for the floats and integers of the selected Coordinated Articulation number. 93 floats and 20 integers are reserved for each configuration.



CoordArtSetupData.tif

Fig. 4-56: Coordinated Articulation Data

For default register and variable label values, refer to section 6.1 for details.

Use Default Coordinated Articulation Labels

Default variable labels and comments can be added by simply leaving the *Use Default Labels* checkbox checked. This is the default setting.

All necessary labels for registers, register bits, and program variables will begin from the float and integer start id block entered in the *Declaration* window. Refer to Use Default Labels on page 4-25 for details.

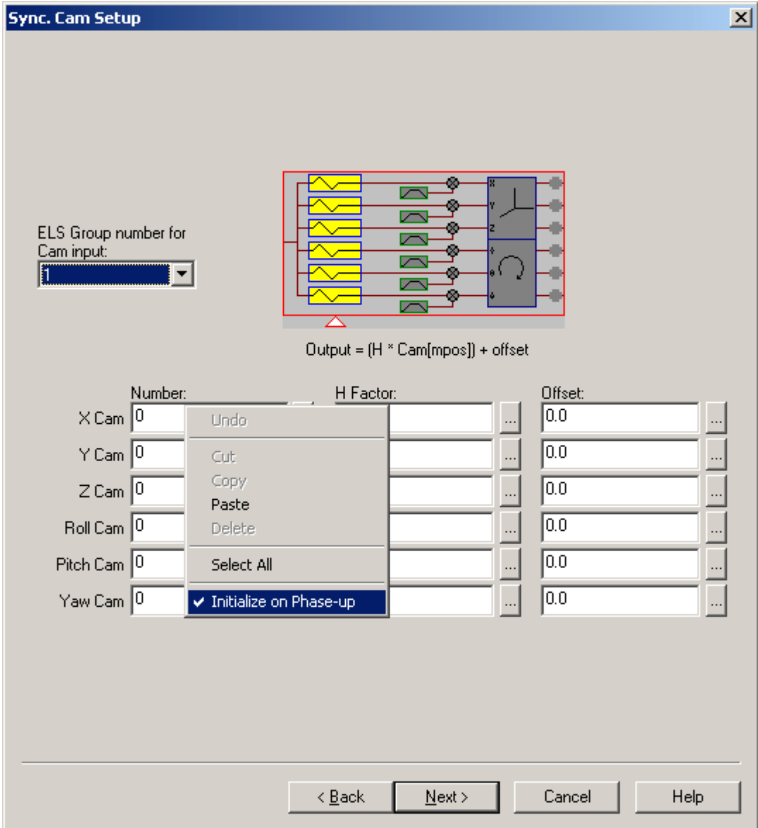
For default register and variable label values, refer to section 6.3, *Coordinated Motion*, for details.

Rexroth Bosch Group

Sync. CAM Setup

The *Sync. CAM Setup* window is used to configure the H factor and Offset for each CAM assigned to the selected ELS group.

Note: Only configured ELS Groups will appear as a selectable ELS Group number.



CA_CamSetup.tif

Fig. 4-57: Sync. CAM Setup

Initialize on Phase-up

Sync. CAM Setup values can be selected to retain it's last value during a system re-initialization or system power down. Refer to Defining Process Variables to Retain Values on page 4-26 for details.

Manual Mode Setup

The *Manual Mode Setup* window is used to set the acceleration, deceleration and velocity that will be used when moving Coordinated Articulation axes in local mode.

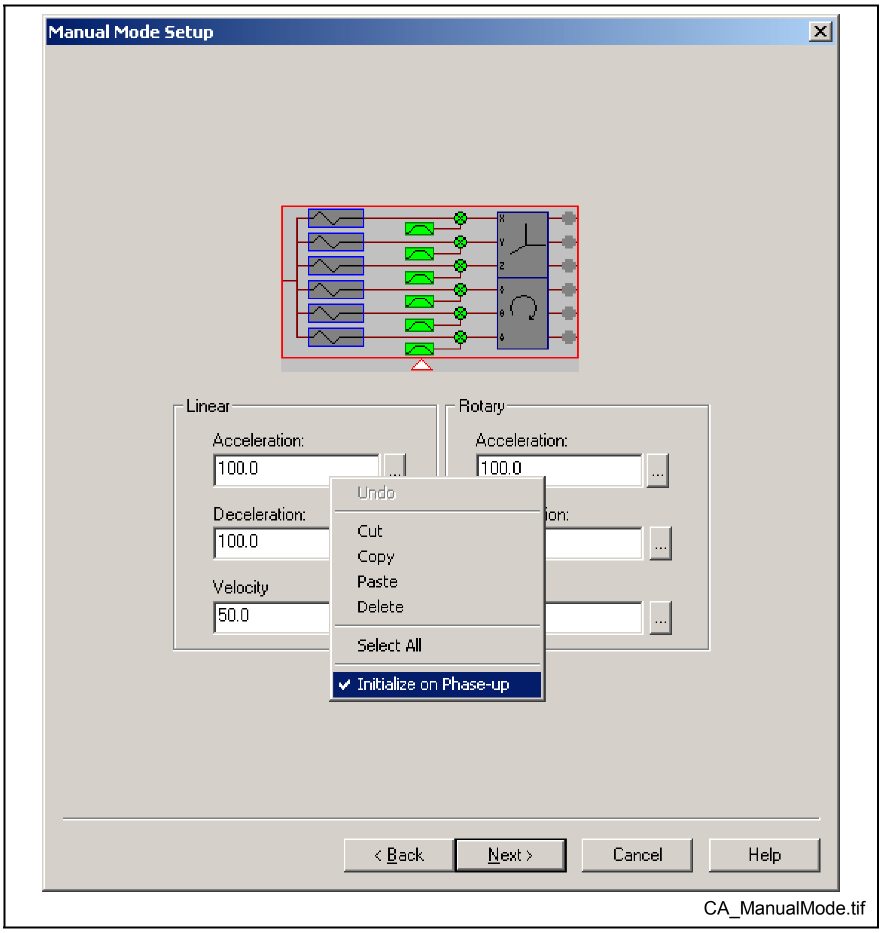


Fig. 4-58: Manual Mode Setup

Initialize on Phase-up

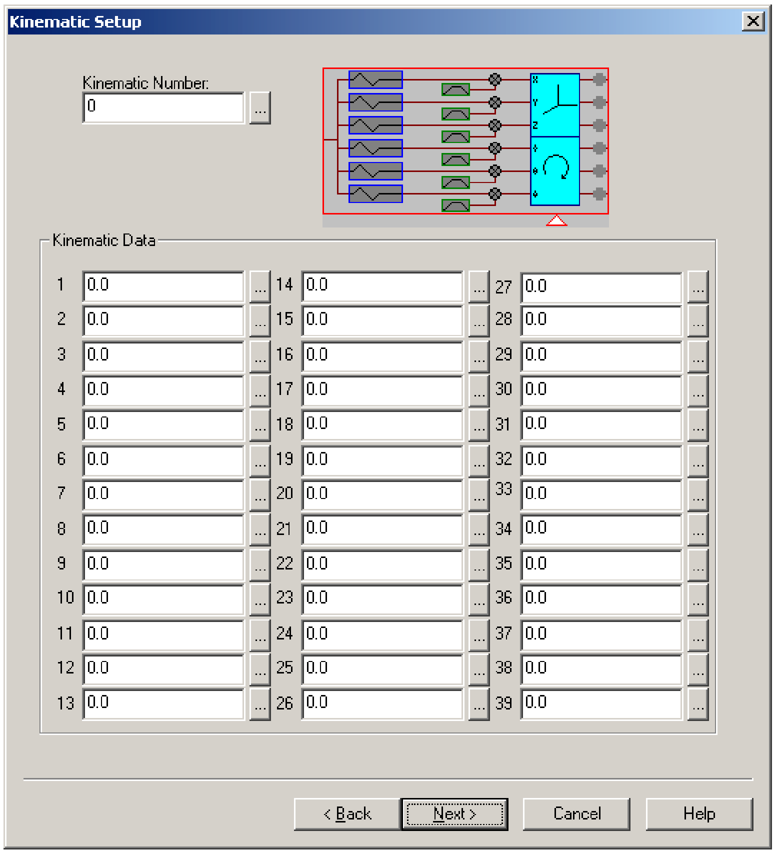
Manual Mode Setup values can be selected to retain it's last value during a system re-initialization or system power down. Refer to Defining Process Variables to Retain Values on page 4-26 for details.

Kinematic Setup

The *Kinematic Setup* window is used to enter the predefined values of the kinematic equation that will be used to determined the positioning of up to six coordinated axes.

Note: Kinematic numbers and data are unique to each application. For a complete list of available Kinematics, refer to section 6.3, *Coordinated Motion*, for details.

Kinematic 13 is a pass-through kinematic. Positional values are pass directly to each axis without any manipulation by a kinematic. No **Kinematic Data** values are required for this kinematic.



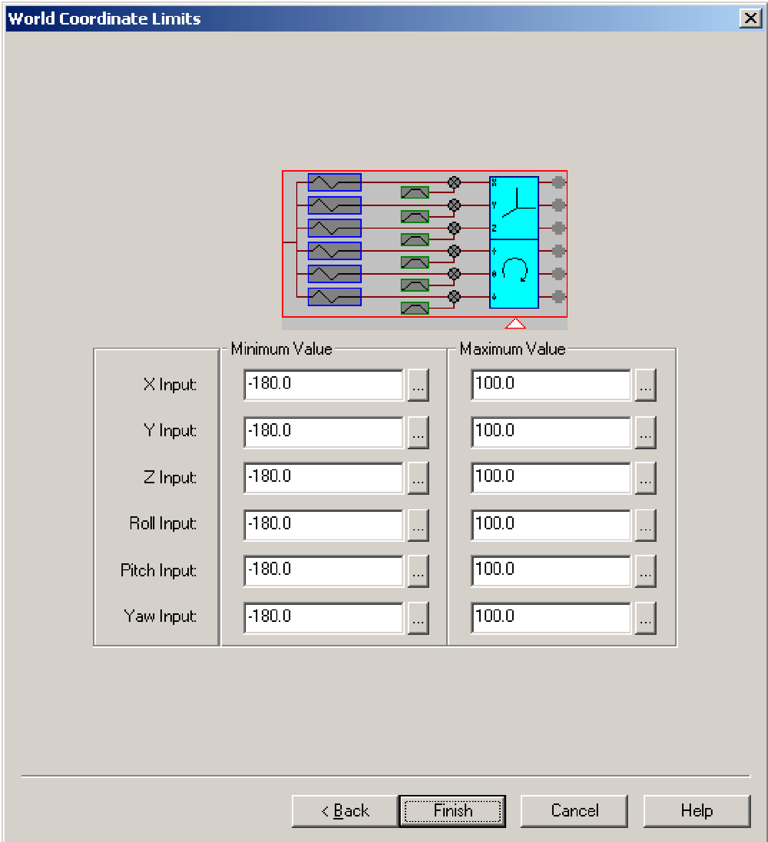
CA_KinematicSetup.tif

Fig. 4-59: Kinematic Setup

World Coordinate Limits

The *World Coordinate Limits* window is used to set the minimum and maximum boundaries that the system will use for the X, Y, Z, Roll Pitch, and Yaw input values.

Note: It is strongly recommended that the world coordinate limits be setup to ensure protection of all hardware.



CA_WorldCoord.tif

Fig. 4-60: World Coordinate Limits

Edit or Delete a Coordinated Articulation Configuration

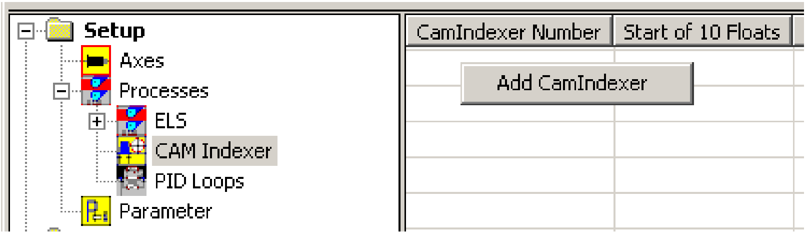
Once configured, a Coordinated Articulation configuration can be edited or deleted by right clicking over the Number and selecting to relevant option. They can also be edited by double clicking over a configuration. Refer to Fig. 4-38 for an example.

4.6 CAM Indexer Setup

Index CAMs are control CAMs that use equations to compute a position, as opposed to a normal CAM, which uses a point table. They operate in real-time, allowing their start and end positions to be freely changed within the next index cycle. This makes them ideally suited for high-speed film feed applications where the seal time must be held constant over line speed.

The CAM Indexer setup is used to declare the registers and assign the variables used to initialize a CAM Indexer and axis Registration.

To declare a CAM Indexer, right click on an empty row and select ***Add CamIndexer***.



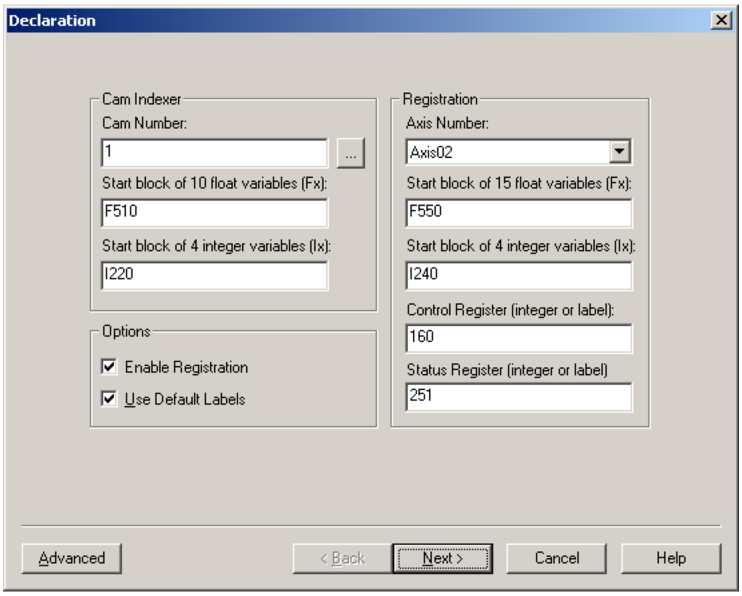
SelectCAMindexerSetup.tif

Fig. 4-61: CAM Indexer Setup

CAM Indexer Declaration

CAM Indexers are identified by a unique number. The number of active CAMs (control table CAMs and CAM Indexers combined) in a VisualMotion control system is limited to the following:

- The maximum of 4 active running CAM Indexers are supported
- A maximum of 40 built CAMs are supported in the system



CAMindexerData.tif

Fig. 4-62: CAM Indexer Data

The *CAM Indexer Declaration* window displays the default variable values for the CAM Indexer and default variable values and registers used for Registration. The registration values for a CAM Indexer are used only if the **Enable Registration** checkbox is selected.

Note: The **Registration** section of the CAM Indexer setup requires that the axis used for the registration feature exist under ***Setup ⇒ Axes***. If a corresponding axis does not exist, the *Registration* section will be disabled.

Use Default CAM Indexer Labels

Default variable labels and comments can be added by simply leaving the *Use Default Labels* checkbox checked. This is the default setting.

All necessary labels for registers, register bits, and program variables will begin from the float and integer start id block entered in the *Declaration* window. Refer to Use Default Labels on page 4-25 for details.

CAM Indexer Default Labels

This section displays the CAM number and default variable start ID blocks (floats and integers) for the specified CAM number.

The following tables list the floats and integer variable labels:

CAM Indexer Float	CAM Indexer			
Default Label	**1**	**2**	**3**	**4**
CAMI_#__PRO_LENGTH	F510	F520	F530	F540
CAMI_#_PRO_START	F511	F521	F531	F541
CAMI_#_PRO_STOP	F512	F522	F532	F542
CAMI_#_DIAMETER	F513	F523	F533	F543
CAMI_#_EVENT_TIME	F514	F524	F534	F544
CAMI_#_ACCEL	F515	F525	F535	F545
CAMI_#_DECEL	F516	F526	F536	F546
CAMI_#_DWELL	F517	F527	F537	F547
CAMI_#_PREDWELL	F518	F528	F538	F548
CAMI_#_DWELL_CENTR	F519	F529	F539	F549
# represents an entry for each CAM Indexer numbered 1 – 4.				

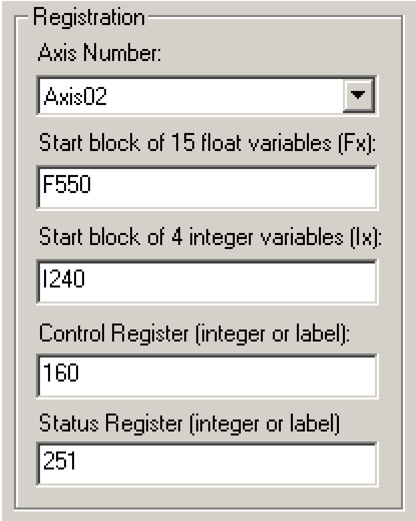
Table 4-15: CAM Indexer Default Float Program Variables

CAM Indexer Integer	CAM Indexer			
Default Label	**1**	**2**	**3**	**4**
CAMI_#_FLAG1	I220	I224	I228	I232
CAMI_#_PRO_TYPE	I221	I225	I229	I233
CAMI_#_RESERVE_I2	I222	I226	I230	I234
CAMI_#_RESERVE_I2	I223	I227	I231	I235
# represents an entry for each CAM Indexer numbered 1 – 4.				

Table 4-16: CAM Indexer Default Integers Program Variables

Registration Variables and Registers

This section is active when the *Enable Registration* checkbox is set and displays the Axis number, the default variable start ID blocks (floats and integers) for the specified Axis number, and the control and status registers.



RegistrationSection.tif

Fig. 4-63: Registration Section

The following tables list the floats and integer variable labels:

Registration Float Default Label	Registration 1	2	3	4
RGIS_##_POS	F550	F570	F590	F610
RGIS_##_PRE_PROBE	F551	F571	F591	F611
RGIS_##_POST_PROBE	F552	F572	F592	F612
RGIS_##_COR_START	F553	F573	F593	F613
RGIS_##_COR_STOP	F554	F574	F594	F614
RGIS_##_MAX_CORRECT	F555	F575	F595	F615
RGIS_##_SETPOINT	F556	F576	F596	F616
RGIS_##_COR_ERROR	F557	F577	F597	F617
RGIS_##_PROBE_VALUE	F558	F578	F598	F618
RGIS_##_LAST_CORR	F559	F579	F599	F619
RGIS_##_RESERVE_F5	F560	F580	F600	F620
RGIS_##_RESERVE_F4	F561	F581	F601	F621
RGIS_##_RESERVE_F3	F562	F582	F602	F622
RGIS_##_RESERVE_F2	F563	F583	F603	F623
RGIS_##_RESERVE_F1	F564	F584	F604	F664
## represents the Sercos address for the assigned axis.				

Table 4-17: Registration Default Float Program Variables

Registration Integer Default Label	Registration			
	1	2	3	4
RGIS_##_FLAG1	I240	I244	I248	I252
RGIS_##_MISS_MARKS	I241	I245	I249	I253
RGIS_##_COR_TYPE	I242	I246	I250	I254
RGIS_##_RESERVE_I1	I243	I247	I251	I255
## represents the Sercos address for the assigned axis.				

Table 4-18: Registration Default Integers Program Variables

Control Register Bit Description	Registration Register-Bit				Functional Description
	1	2	3	4	
Enable Preset	160-09	161-09	162-09	163-09	When set to 1, it resets the FIFO buffer and or the average if buffering or averaging correction is used. Otherwise, this bit is ignored.
Enable Registration	160-10	161-10	162-10	163-10	When set to 1, Registration is enabled. When set to 0, Registration is disabled.

Table 4-19: Registration Default Control Registers

Status Register Bit Description	Registration Register-Bit				Functional Description
	1	2	3	4	
Correction at Max	251-09	252-09	253-09	254-09	1 – Probe Distance exceeds the *Max. Correction*. 0 – resets after the next mark is found and Probe Distance is less than or equal to Max. Correction.
Reserved	251-10	252-10	253-10	254-10	Reserved
Mark in Window	251-11	252-11	253-11	254-11	1 – Registration mark is within Probe Window 0 – initially zero or when registration mark is outside Probe Window.
Preset Enabled	251-12	252-12	253-12	254-12	Acknowledgement for Enable Preset control bit
Registration Enabled	251-13	252-13	253-13	254-13	1– Registration is enabled 0 – Registration is disabled
Max Missed Marks	251-14	252-14	253-14	254-14	1 – Mark counter exceeds RGIS_##_MISS_MARKS 0 – Reset with a 1→ 0 transition of Enable Registration control bit.

Table 4-20: Registration Default Status Registers

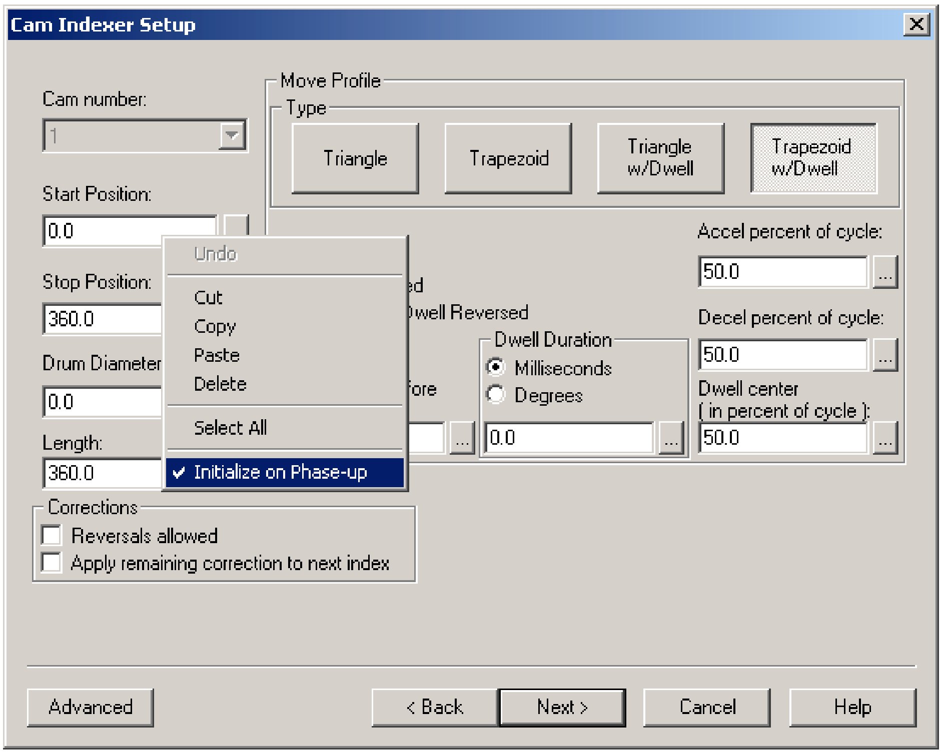
CAM Indexer Setup

CAM Indexer Setup is used to initially configure the following data:

- Start Position
- Stop Position
- Drum Diameter
- Length
- Corrections
- Move Profile

Note: When using CAM Indexers, increase the Sercos cycle time (C-0-0099) to 4ms or more to allow time for the equation calculations.

Refer to section 8.4, *CAM Indexer*, for details on the functionality of the CAM Indexer settings and options.



CAMindexerSetup.tif

Fig. 4-64: CAM Indexer Setup

Initialize on Phase-up

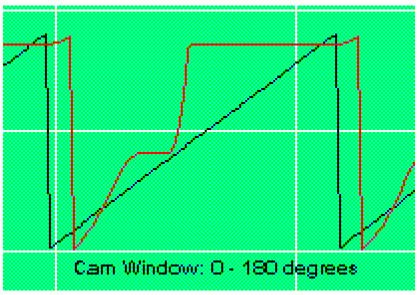
Some CAM Indexer Setup values can be selected to retain their last value during a system re-initialization or system power down. Refer to Defining Process Variables to Retain Values on page 4-26 for details.

Start Position (Ps)

Defines the virtual or real master position (in degrees) at which the CAM index profile starts.

Stop (Pe) Position

Defines the virtual or real master position at which the CAM index profile stops.



These two parameters define the CAM's cycle time, **Tc**,

Tc = ABS(**Pe**-**Ps**)/Vm

where ABS is the absolute value of the difference and **Vm** is the velocity of the master in degrees/sec.

Length (D)

Defines the distance (in EE units) the slave axis moves during an index cycle. The velocity profile developed depends on the start/stop positions, profile type, index length and master speed.

Drum Diameter (Dia) Allows for automatic scaling of the index distance into degrees. The following equation is applied:

Angle = (**D**/**Dia**)(360/π)

Internal CAM units (degrees) are computed automatically through this parameter.

Note: When the Drum Diameter is set to zero, its default value is 360°/rev. The benefit of this is that internal 64 bit math can be used to minimize rounding errors.

Move Profile

The following types of Move Profiles are available:

Profile	**Description**
Triangle	Step Acc/Dec of 50%
Triangle (S-Curve)	Ramp Acc/Dec of 50%
Trapezoid (Fixed)	Step Acc/Dec of 33%
Trapezoid	Step Acc(x%)/Dec(y%), x+y≤100%
Trapezoid (S-Curve)	Ramp Acc(x%)/Dec(y%), x+y≤100%
Triangle w/Dwell	Step Acc/Dec, center dwell
Triangle w/Dwell (S-Curve)	Ramp Acc/Dec, center dwell
Trapezoid w/Dwell (S-Curve)	Ramp Acc/Dec, center dwell
Trapezoid w/Dwell	Step Acc/Dec, center dwell
Trapezoid w/Dwell	Step Acc(x%)/Dec(y%) x+y≤100%, Sw% dwell

Table 4-21: Move Profiles

CAMs with ramped acceleration (S-shaped velocity profile) are defined as third order polynomials, those with stepped acceleration (ramp velocity profile) as second order polynomials.

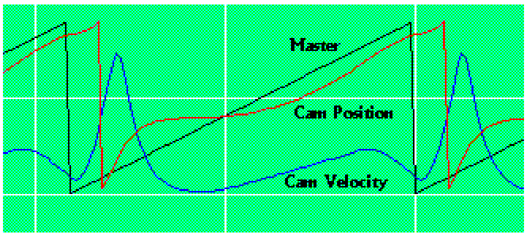
The Triangle profile without the S-Curve shaper has the simplest equation, thus it takes the least amount of time to execute, but has maximum jerk. Selecting the S-Curve shaper takes longer but minimizes jerk.

The Trapezoid profile provides CAM shaping through an accel./decel. percent setting but takes the longest to execute due to equation complexity.

CAMs with dwells are slightly more complex than those without dwells and so require slightly more time to execute than those without dwells.

CAM Acceleration/Deceleration(Acc/Dec) - Some CAM types (see table) use these parameters to define the percent of their cycle time devoted to acceleration and deceleration.

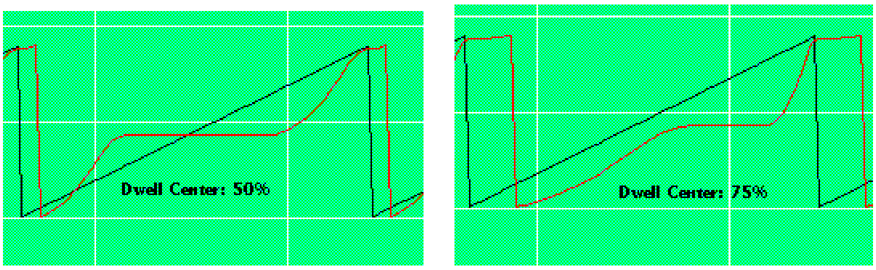
Example: A Trapezoid CAM profile with an acceleration setting to 20% and deceleration setting to 50% would step accelerate for 20% of its cycle time, constant velocity for 30% and decelerate for 50%.



CAM Dwell (Cw) - Some CAMs (see table) use this parameter to determine the center of the dwell. This is defined as a percent, which is limited to 25% to 75% of the CAM cycle.

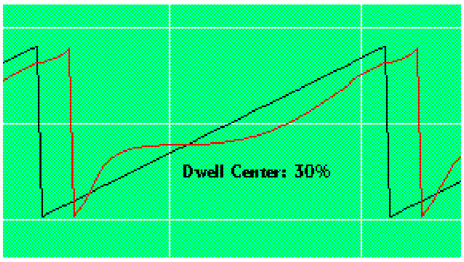
Example1: Cw = 50% Ps = 0 Pe=360 then dwell center

Pw = (360-0)*0.5 + 0 = 180 deg.



Example2: Cw = 30% Ps=90 Pe=270 then dwell center

Pw = (270-90)*0.3+90 = 144 deg.



CAM Dwell (Tw) - Some CAMs use this parameter to determine the amount of dwell within their cycle. A dwell is a period of time during which the CAM is stopped.

Tw is limited by the following relationship:

Max(Tw) = 2.0*Min[0.40*(Pe-Pw)/Vm, 0.40*(Pe-Pw)/Vm]

where Pw (deg) is the dwell center angle and Vm (deg/sec) is master speed.

Assigning a CAM Indexer to an Axis or ELS Group

A CAM Indexer is assigned to any ELS Group under ***Setup*** ⇒ ***Process*** ⇒ ***ELS*** ⇒ ***ELS Group***. The ELS Group icon can be used to assign a different CAM Indexer. All slave axis assigned to the ELS Group will use the output of the ELS Group, including the CAM Indexer equation, as the master input. In order for a CAM Indexer to function in an ELS Group, the Lock On / Lock Off State Machine must be disabled.

A CAM Indexer can function in an associated slave axis while the ELS Group has an active Lock On / Lock Off State Machine.

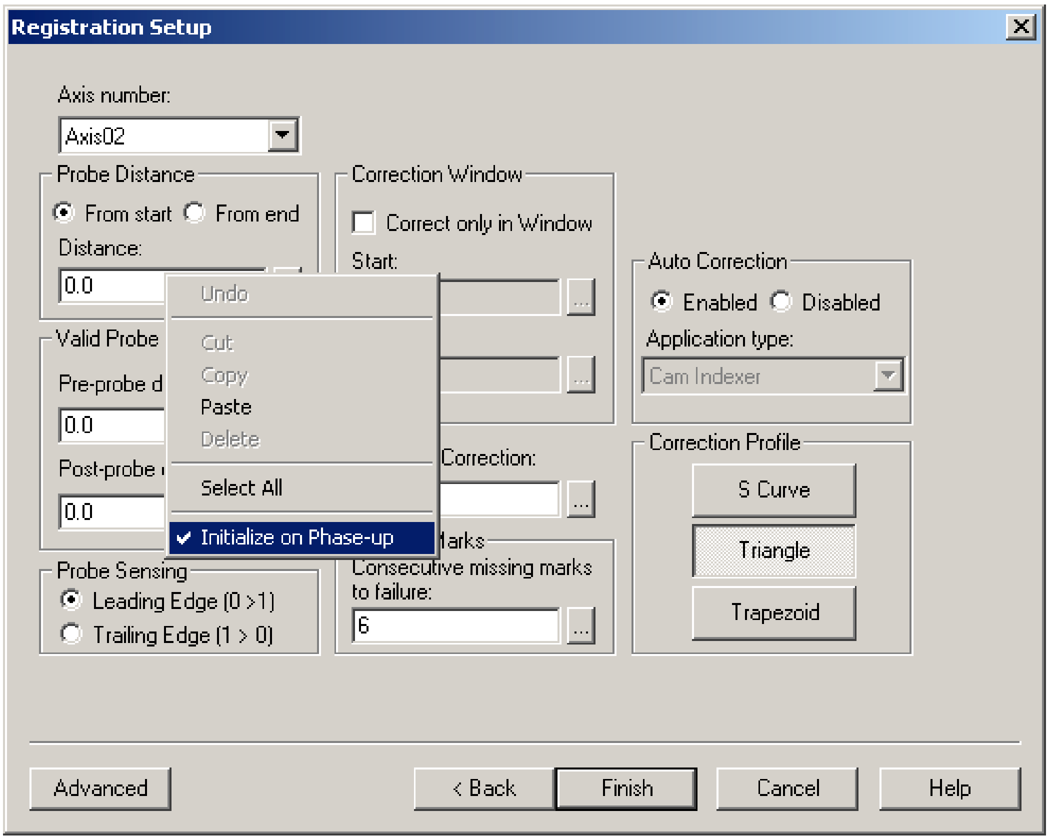
Assigning a slave axis to a CAM Indexer requires that the axis be declared as a control CAM and then associated to the CAM Indexer using the CAM icon.

Registration Setup

Registration Setup is used to initially configure the following data:

- Probe Distance
- Valid Probe Window
- Probe Sensing
- Correction Window
- Maximum Correction
- Auto Correction
- Correction Profile

For specific information on Registration functionality, refer to section 8.4, *CAM Indexer*.



RegistrationSetup.tif

Fig. 4-65: Registration Setup

Edit or Delete a CAM Indexer

Once configured, a CAM Indexer can be edited or deleted by right clicking over the Number and selecting to relevant option. They can also be edited by double clicking over a configured row. Refer to Fig. 4-38 for an example.

4.7 PID Loops Setup

The PID Loop setup is used to assign initialization variables for up to 32 PID Loops. These PID Loops act as background processes with user controlled variables.

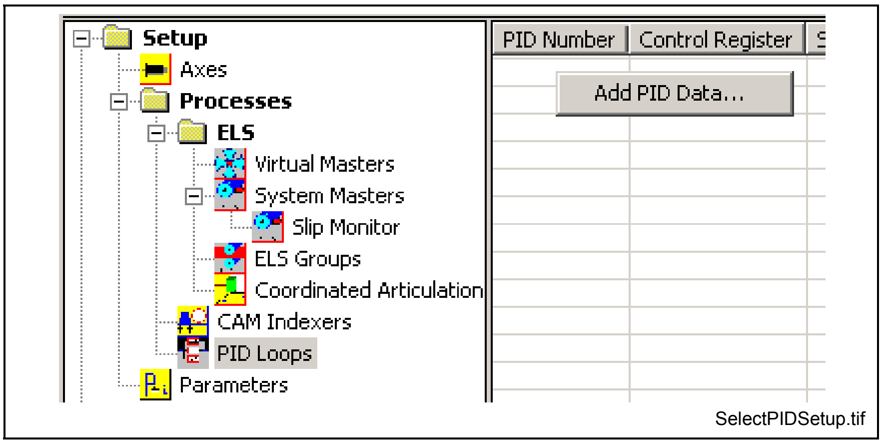


Fig. 4-66: PID Loop Setup

PID Loop Declaration

To declare a PID Loop, right click on an empty row and select ***Add PID Data***. The wizard will guide the user through the setup process.

The *PID Declaration* window displays the default variables and registers for the selected PID Loop number.

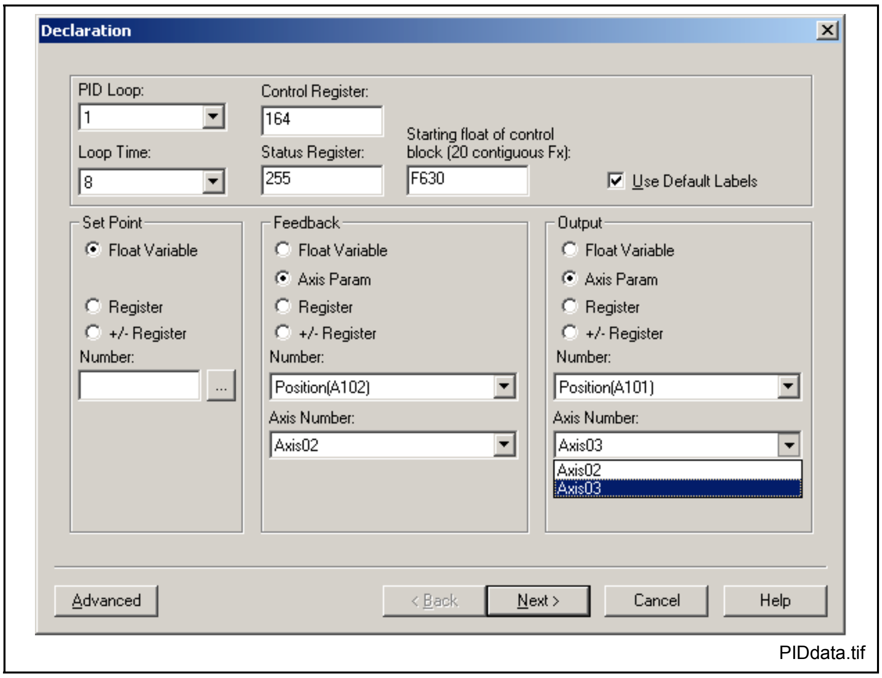


Fig. 4-67: PID Data Window

Use Default PID Loop Labels

Default variable labels and comments can be added by simply leaving the *Use Default Labels* checkbox checked. This is the default setting.

All necessary labels for registers, register bits, and program variables will begin from the float and integer start id block entered in the *Declaration* window. Refer to Use Default Labels on page 4-25 for details.

The following table describes the different setting in the PID Data window:

Settings	**Description**
PID Loop	PID Loop configuration number. Up to 32 loops are supported in a G*P 11 project.
Loop Time	PID Loop update rate, selectable in multiples of 8 ms from 8 to 152 ms
Control Register	Register number assigned for control functions.
Status Register	Register number assigned for status functions.
Starting Float Block	First float number in a block of 20 contiguous floats.
Set Point	Float or register where user defined PID target value is stored.
Feedback	Float, axis parameter, or register where current calculated feedback value from PID is stored.
Output	Float, axis parameter, or register where current resultant PID output value is stored.

Table 4-22: PID Data Settings

PID Loop Variables

The following table lists the default float variable labels used for PID Loops:

PID Loop Float Default Label	PID Number (Up to 32 Loops) 1	2	3	…	32	Description	Update Phase
PID#_CMD_SCALER	F630	F650	F670	…	F1250	Command scalar value, default 1.0	4
PID#_CMD_BIAS	F631	F651	F671	…	F1251	Command bias value, default 0.0	4
PID#_FDBK_SCALER	F632	F652	F672	…	F1252	Feedback scalar value, default 1.0	4
PID#_FDBK_BIAS	F633	F653	F673	…	F1253	Feedback bias value, default 0.0	4
PID#_KP_VALUE	F634	F654	F674	…	F1254	Kp value, default 1.0	4
PID#_KI_VALUE	F635	F655	F675	…	F1255	Ki value, default 0.0	4
PID#_Kd_VALUE	F636	F656	F676	…	F1256	Kd value, default 0.0	4
PID#_KI_MAX_VALUE	F637	F657	F677	…	F1257	Ki limit value, default 0.0	4
PID#_MIN_OUTPUT	F638	F658	F678	…	F1258	Minimum output value, default -10.0	4
PID#_MAX_OUTPUT	F639	F659	F679	…	F1259	Maximum output value, default 10.0	4
PID#_KI_PRESET	F640	F660	F680	…	F1260	Preset value, default 0.0	4
PID#_OUT_SCALER	F641	F661	F681	…	F1261	Output scalar value, default 1.0	4
PID#_OUT_BIAS	F642	F662	F682	…	F1262	Output bias value, default 0.0	4
PID#_FDBK_CUTOFF	F643	F663	F683	…	F1263	Feedback cutoff frequency (Hz), default 0	4
PID#_FDBK_FILTER	F644	F664	F684	…	F1264	Feedback filter type, default 0 0 - None 1 - 1st order Low-pass 2 - 2nd order Low-pass 3 - 3rd order Butterworth 4 - 2nd order Butterworth 5 - 3rd order Butterworth 6 - modified 2nd order Low-pass with velocity feed forward 7 - modified 3rd order Low-pass with velocity and acceleration feed forward	4
PID#_FDBK_VALUE	F645	F665	F685	…	F1265	Internal feedback after conditioning with bias, scalar and filter	4
PID#_CMD_VALUE	F646	F666	F686	…	F1266	Internal command value after conditioning with bias and scalar	4
PID#_OUT_VALUE	F647	F667	F687	…	F1267	Output value	4
PID#_RESERVE_F2	F648	F668	F688	…	F1268	Reserved	4
PID#_RESERVE_F1	F649	F669	F689	…	F1269	Reserved	4
# represents an entry for each PID Loop number from 1-32							

Table 4-23: PID Loop Default Float Program Variables

PID Loop Control and Status Registers

The following table lists the default control and status registers used for PID Loops:

Control Register Bit Description	PID Register-Bit (Up to 32 Loops) 1	2	3	…	32	Functional Description
Enable Preset	164-04	165-04	166-04	…	195-04	If set, loads integral preset on program activation or when loop enabled by bit 5
Enable PID	164-05	165-05	166-05	…	195-05	If set high in phase 4, the PID loop is enabled. When disabled, the output is set to zero on the falling edge. This bit is checked on every PID loop time interval.

Table 4-24: PID Loop Default Control Registers

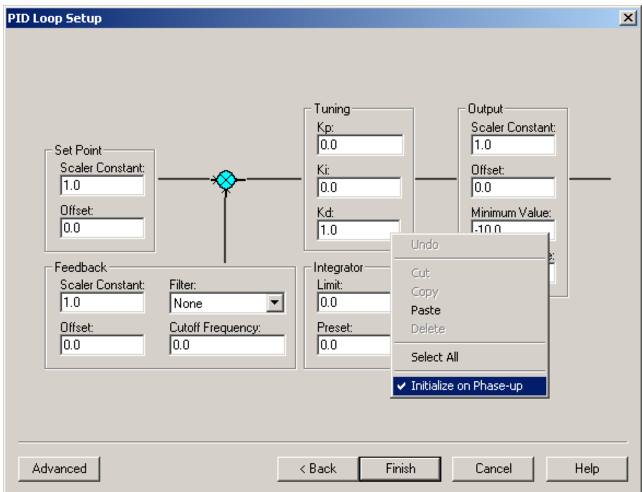
Status Register Bit Description	PID Register-Bit (Up to 32 Loops) 1	2	3	…	32	Functional Description
Output above Max	255-01	256-01	257-01	…	286-01	If the output goes above the user defined max value, this bit will be set.
Output below Min	255-02	256-02	257-02	…	286-02	If the output goes below the user defined min value, this bit will be set
Output in Window	255-03	256-03	257-03	…	286-03	If the output is at or above the min value and at or below the max value, this bit will be set. It is redundant to bits 1 and 2.
Preset Enabled	255-04	256-04	257-04	…	286-04	Acknowledgment of control register preset bit 4
PID Enabled	255-05	256-05	257-05	…	286-05	Acknowledgment of control register enable bit 5

Table 4-25: PID Loop Default Status Registers

PID Loop Setup

The PID Loop Setup window is used to configure the following data:

- Set Point
- Feedback
- Tuning
- Integrator
- Output



PIDLoopSetup.tif

Fig. 4-68: PID Loop Setup

Rexroth
Bosch Group

Initialize on Phase-up

Some PID Loop Setup values can be selected to retain their last value during a system re-initialization or system power down. Refer to Defining Process Variables to Retain Values on page 4-26 for details.

Feedback:

Digital filtering is available for PID loops and Real Masters.

Filter Type:

None

First order low-pass, $G(s)=1/(s+1)$

Second order low-pass, $G(s)=1/(s^2 +2s +1)$

Third order low-pass, $G(s)=1/(s^3+3s^2+3s+1)$

Second order Butterworth, $G(s)=1/(s^2 +2^{1/2}s +1)$

Third order Butterworth, $G(s)=1/(s^3+2s^2+2s+1)$

Modified 2nd order low-pass with velocity ramp tracking,

$G(s)=(2s+1)/(s^2 +2s +1)$

Modified 3rd order low-pass with accel ramp tracking,

$G(s)=(3s^2+3s+1)/(s^3+3s^2+3s+1)$

Cutoff Frequency(float):

When a filter type is chosen, a cutoff frequency for the filter must be entered. The cutoff frequency is the frequency where the signal is reduced by 3db. When set to 0 the filter is disabled.

To ensure a stable system, use the following calculation when entering a value for the Digital Filter Cutoff Frequency:

$$\text{Cutoff Frequency} \le \frac{1}{2 * \text{Sampling Rate(sec.)}}$$

The sampling rate for a PID loop is the set PID Loop Time, entered in seconds.

Example: For a 8 ms PID Loop Time, the cutoff frequency is calculated as follows:

$$\text{Cutoff Frequency} \le \frac{1}{2 * 0.008} = 62.5 \text{ Hz}$$

The following figure illustrates the frequency vs. degrees for each filter. The cutoff frequency is 10 hertz and the sampling rate is 4 ms.

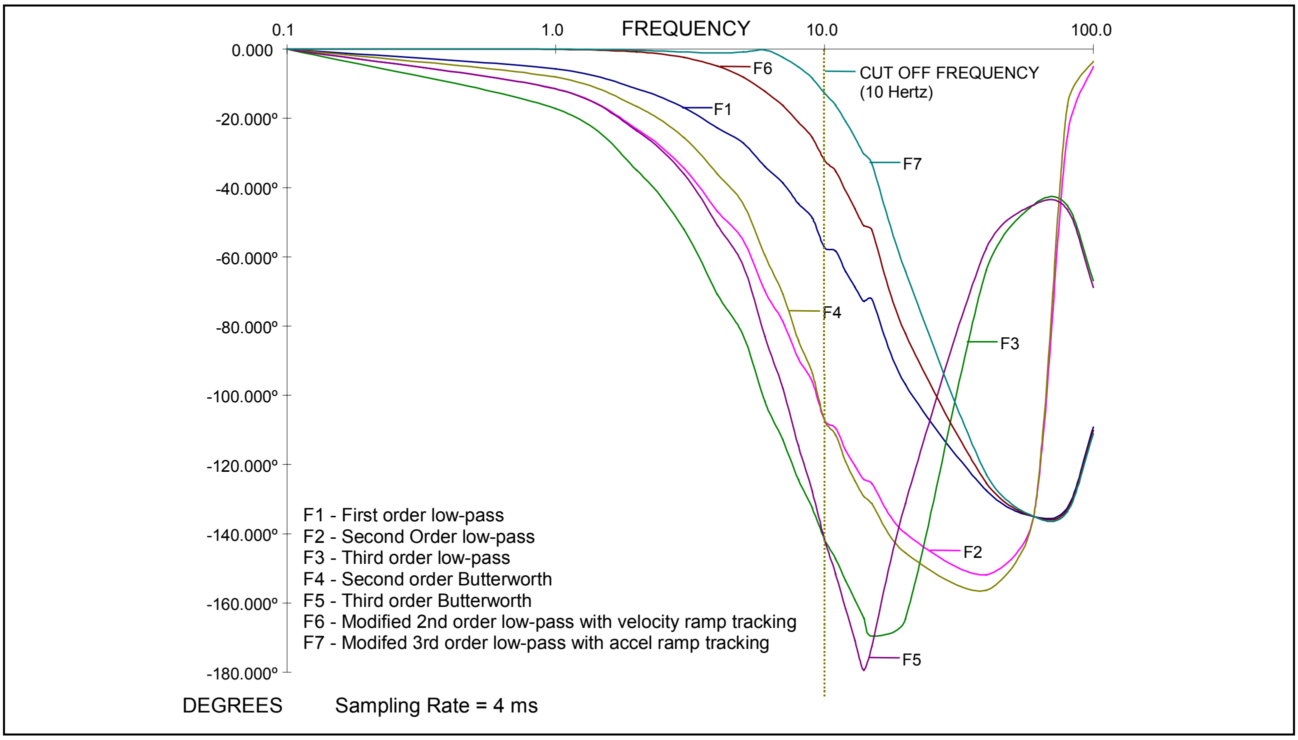


Fig. 4-69: Frequency vs. Degree Filter Chart

The following figure illustrates the gain vs. frequency for each filter. The cutoff frequency is 10 hertz and the sampling rate is 4 ms.

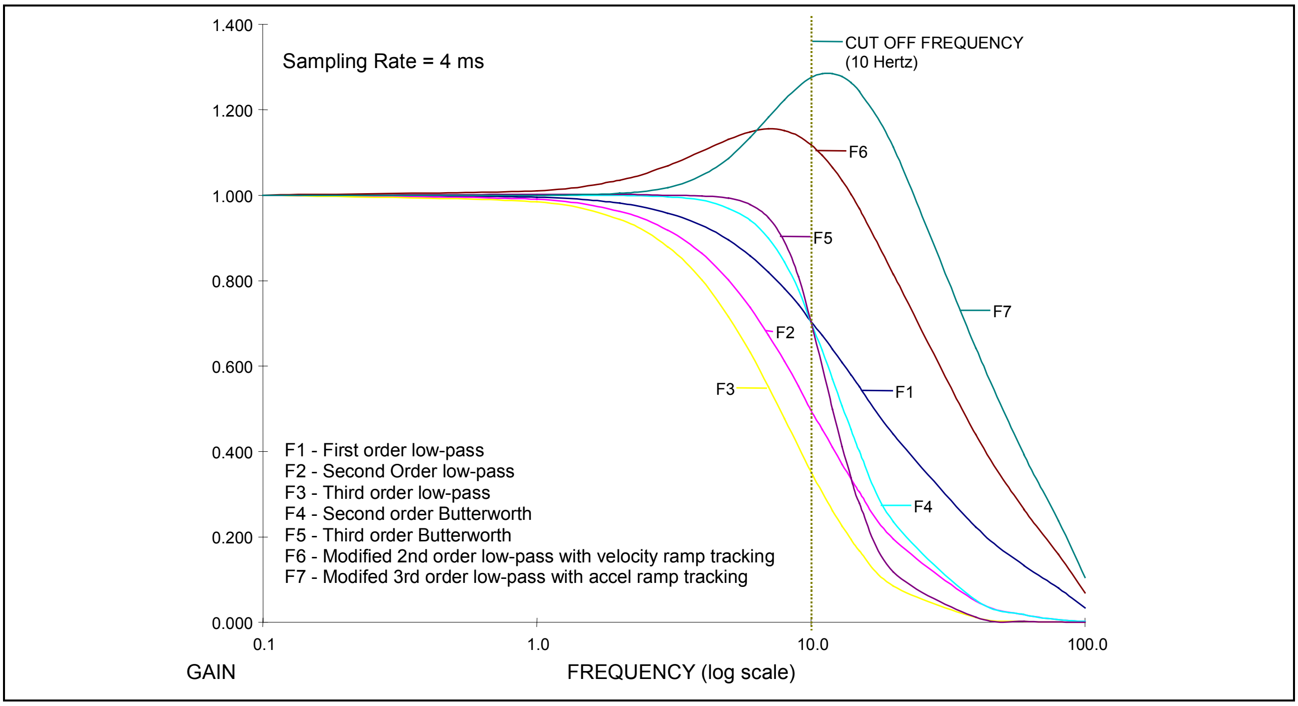


Fig. 4-70: Gain vs. Frequency Filter Chart

When a filter is chosen, the cutoff frequency for the filter must be entered. For example, the cutoff frequency for the First order low-pass filter is the frequency where the signal is reduced 3db [.707 gain, db=20*log(gain)].

Edit or Delete a PID Loop

Once configured, a PID Loop can be edited or deleted by right clicking over the Number and selecting to relevant option. They can also be edited by double clicking over a configured row. Refer to Fig. 4-38 for an example.

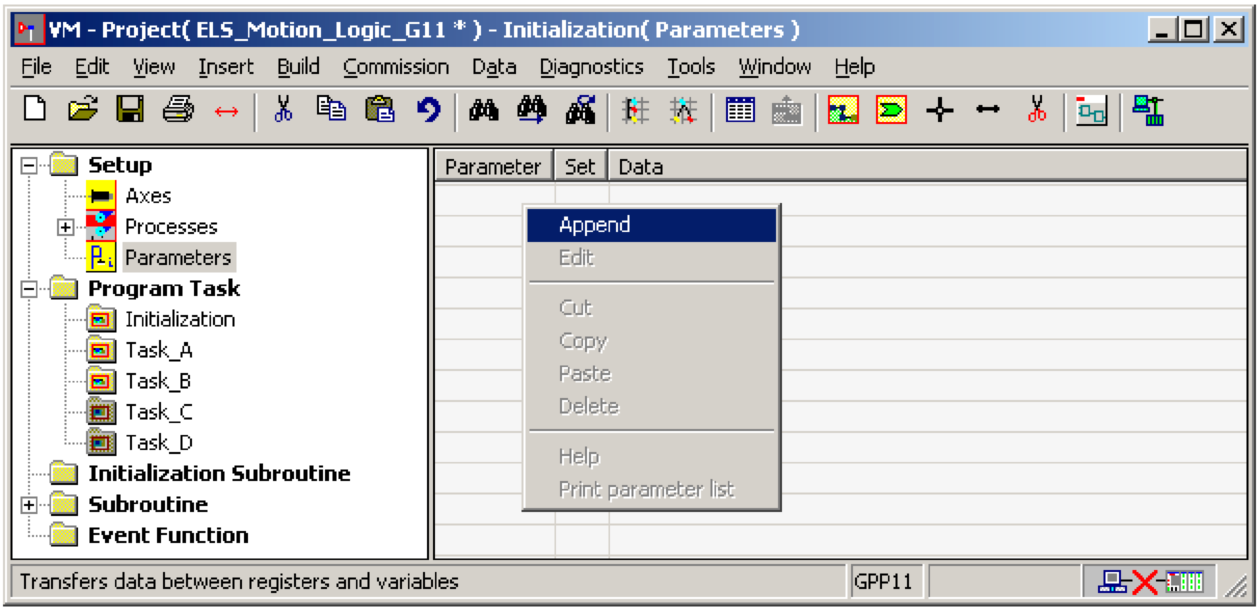
4.8 Setup Parameters

VisualMotion 11 has combined the functionality of the *PrmInt* and *PrmBit* icons as a setup list configuration. The ***Setup*** ⇒ ***Parameters*** selection is used for the initialization of system parameters in a G*P 11 project. The user can setup single or multiple parameters as well as individual parameter bit state changes of binary parameters.

Note: The **PrmInt** and **PrmBit** icons have been removed from the initialization palette for G*P 11 projects. Pre-G*P 11 projects, opened using VisualMotion Toolkit 11 (without converting to a G*P 11 project) will display the PrmInt and PrmBit icons in the initialization palette. These programs can be compiled and downloaded to the control using VisualMotion Toolkit 11 with complete support for the PrmInt and PrmBit icons.

Older firmware targets (i.e., G*P 10) that are converted to a G*P 11 project should use ***Setup*** ⇒ ***Parameters*** for any new parameter initialization. Existing PrmInt and PrmBit icons in the program will be supported and can be compiled and downloaded to the control.

After selecting **Parameters** from the Project Navigator window, right click and select ***Append*** to begin.



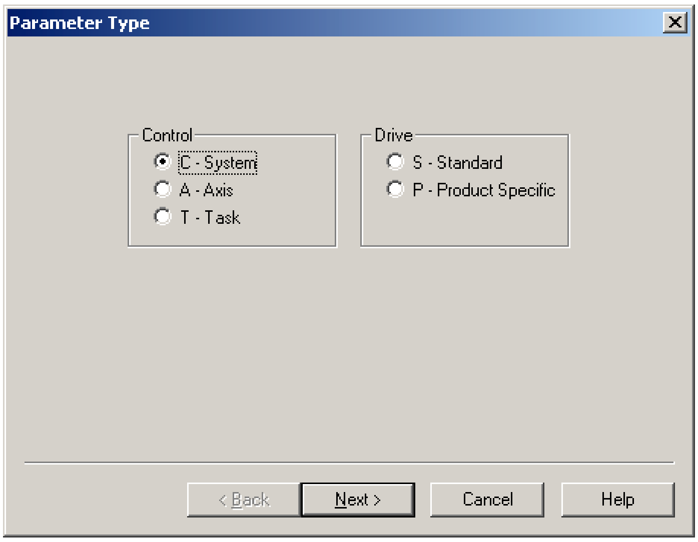
ParameterSetup.tif

Fig. 4-71: Project Navigator showing Parameter Setup

Note: Communication errors can occur when a program is activated that contains parameter initializations that modify the X10 or X16 communication parameters. Refer to *Communication Errors After Program Activation* in section 10.3 under heading *Online Full Restore* for details.

Parameter Type

The *Parameter Type* window is used to select control, axis, task, or drive parameters for setup initialization.



ParameterTypeSelection.tif

Fig. 4-72: Parameter Type Selection

When the ***Next >*** button is selected, the window that follows depends on the parameter type selected. Refer to the following table for details:

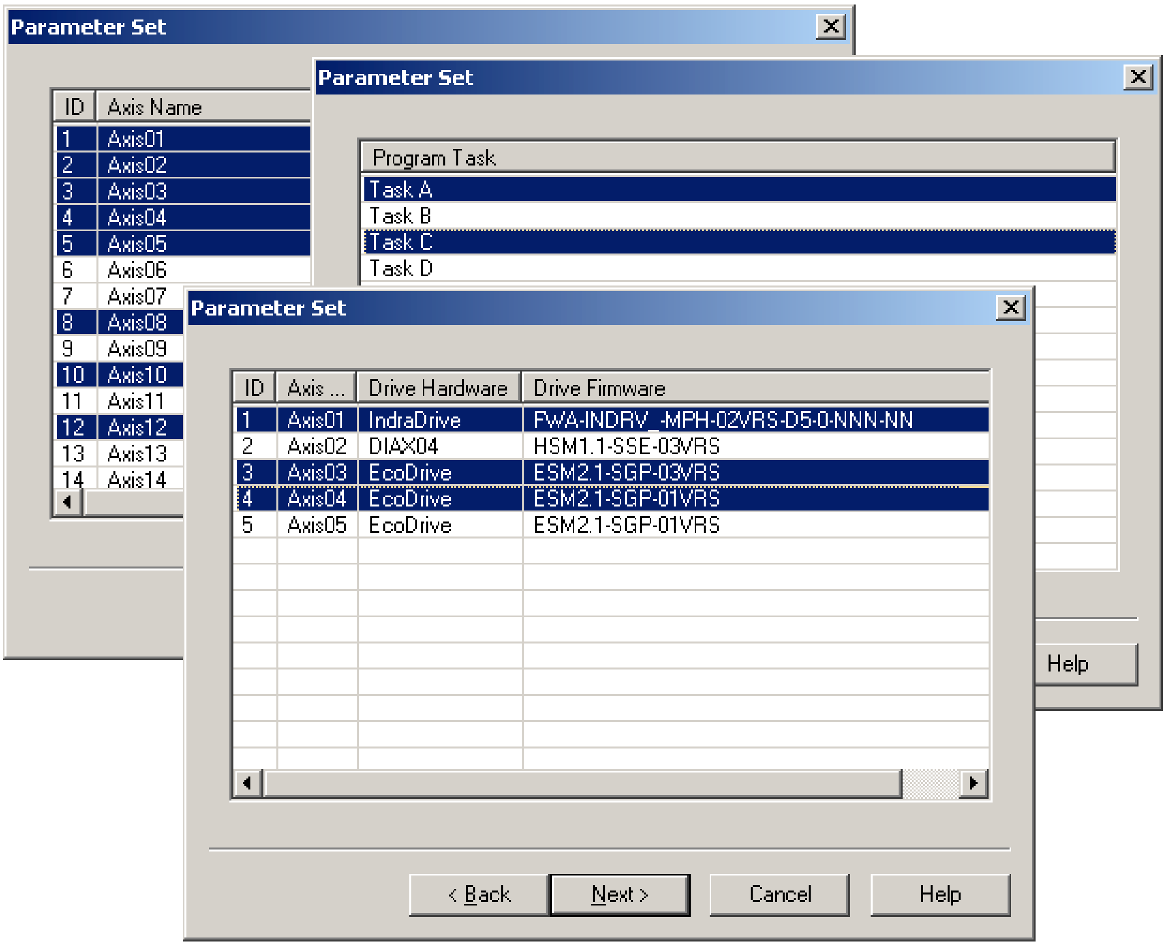
Parameter Type	Next Window
C – System	Parameter Number
A-Axis, Task, Drive	Parameter Set

Table 4-26: Windows following Parameter Type

Parameter Set

From the *Parameter Set* window, one or more axis, task, or drive (S and P) parameter types can have parameter values initialized at one time. To select multiple items, for example axis 1, 3, 5, hold the ***Ctrl*** key down while making selections. To select a range of items, for example axis 1-5, hold the ***Shift*** key down while making selections.

Rexroth Bosch Group



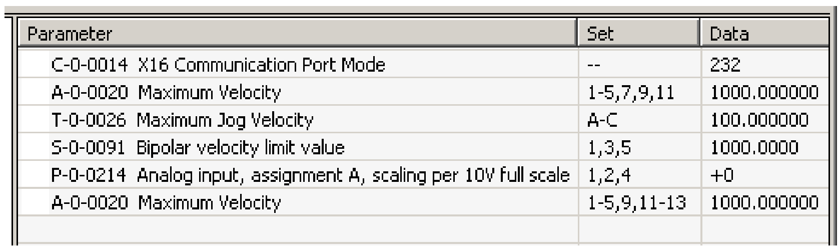
ParamSetSelection.tif

Fig. 4-73: Selecting Axis Parameters

The number of items displayed in the *Parameter Set* window depends on the type selected. Refer to the following table for details:

Parameter Type	**Available Number of Items**
Axis	All 64 axis are available. The Axis Name is assigned during the initial Axes Setup.
Task	Task A-D are available
Drive	Only drives assigned to axes are available.

Table 4-27: Parameter Set Selection



Parameter	Set	Data
C-0-0014 X16 Communication Port Mode	--	232
A-0-0020 Maximum Velocity	1-5,7,9,11	1000.000000
T-0-0026 Maximum Jog Velocity	A-C	100.000000
S-0-0091 Bipolar velocity limit value	1,3,5	1000.0000
P-0-0214 Analog input, assignment A, scaling per 10V full scale	1,2,4	+0
A-0-0020 Maximum Velocity	1-5,9,11-13	1000.000000

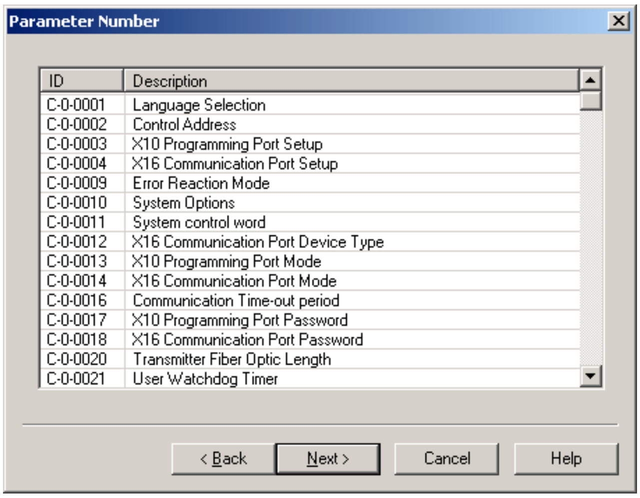
ParamInitExample.tif

Fig. 4-74: Parameter Initialization Setup List Example

Clicking the ***Next >*** button, opens the *Parameter Number* window containing a list of valid parameters that can be configured for the items selected in the *Parameter Set* window.

Parameter Number

The *Parameter Number* window is used by all parameter types to select an individual parameter to be initialized at program activation. When opened, the *Parameter Number* window loads a list of valid control, axis, task, or drive (S or P) parameters. Double clicking on a specific parameter opens the *Parameter Value* window. Refer to Setting Parameter and Parameter Bit Values for details.

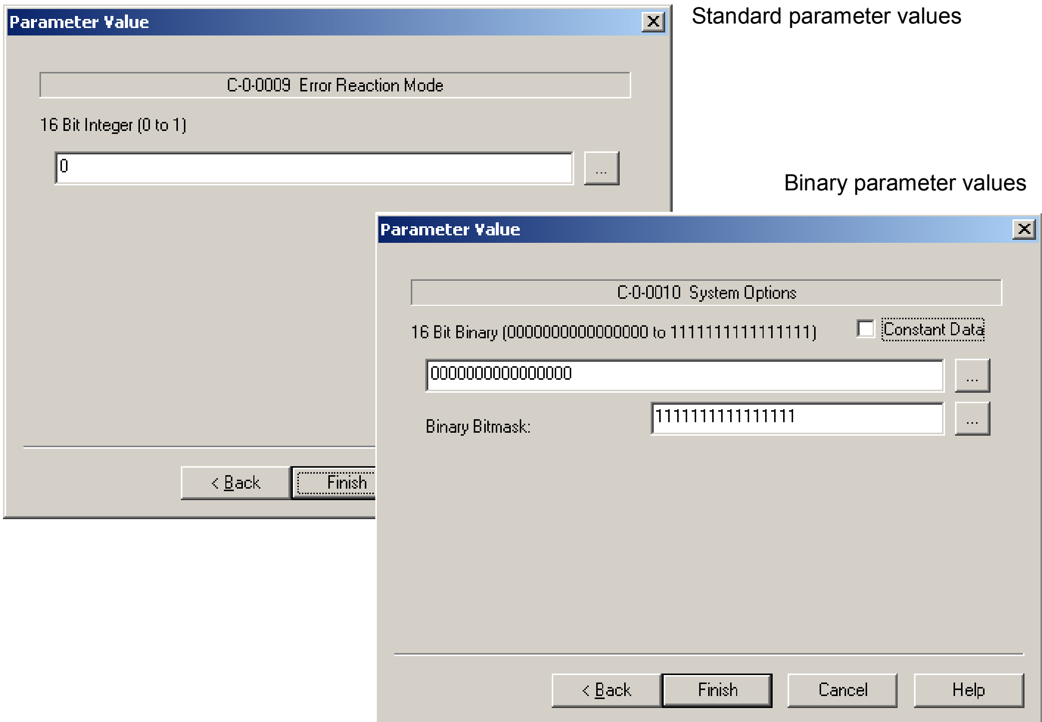


ValidCparameters.tif

Fig. 4-75: Valid Control Parameters

Setting Parameter and Parameter Bit Values

The Parameter Value window is used to enter the desired value for the selected parameter to be initialized. Individual bit states can be selected when a binary parameter is selected. The following figure shows the different options available when setting parameter values:



InitParamValue.tif

Fig. 4-76: Initializing Parameter Values

Rexroth
Bosch Group

Constant Data

By default the **Constant Data** checkbox for binary type parameters is selected. It allows the user to enter a constant value for the parameter. The allowable settings for each bit are as follows:

- 0 (off state)
- 1 (on state)
- X (state does not matter)

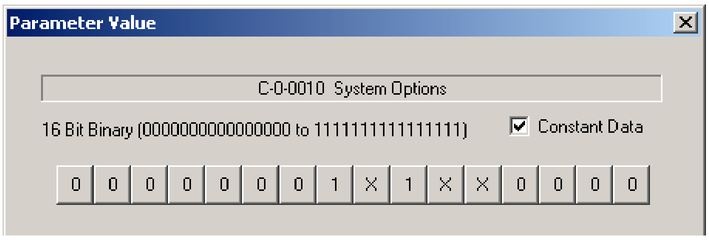


Fig. 4-77: Selecting Constant Data

4.9 Defining Project Integers, Floats and Variables

Defining Project Values

Values assigned to an icon for position, or movement can be a numeric value or a text label. When a numeric value is assigned in an icon, the value is fixed in the project once it has been downloaded to the control. Changing the numeric value requires recompiling and downloading the project.

For greater flexibility, a label, known as a variable, can be assigned in the icon. The variable is a user-defined, text-based name that is assigned a positive, whole number. Both the variable name and number identifies a location in the control's memory where a numeric value is stored. The assigned value of a variable can be changed while a project is running without having to recompile the project.

Two types of values can be used in a VisualMotion Project:

- Integer – a positive or negative whole number
- Floats – a positive or negative decimal

For each float and integer in a project, a variable and number can be assigned.

Defining Project Variables

There are three designations for variables that indicate the accessibility of the variable:

- Global Variables – available to any project
- Program Variables – available throughout the project they are assigned to
- Local Variables – available to the task, subroutine, or event function in which it is assigned.

Global Variables

Global variables, designated GF[#] (global float) and GI[#] (global integer), are stored in the control's memory and their values are not retained after power is disconnected. Global floats and integers can, however, be saved to flash memory using VisualMotion Toolkit's Data Editor. Refer to the *VisualMotion 11 Functional Description* manual for information on the Data Editor.

Program Variables

Program variables are designated F[#] (float) and I[#] (integer). Program variables are stored in Autostore and retain their values during power off. The variables can be addressed in a project by assigning a label to the variable number.

Local Variables

Local or *stack based* variables exist only while in the task, subroutine, or event function where they are declared. Function arguments are used within a subroutine for local data only. This type of variable is useful for temporary results within a function or to pass values to a function.

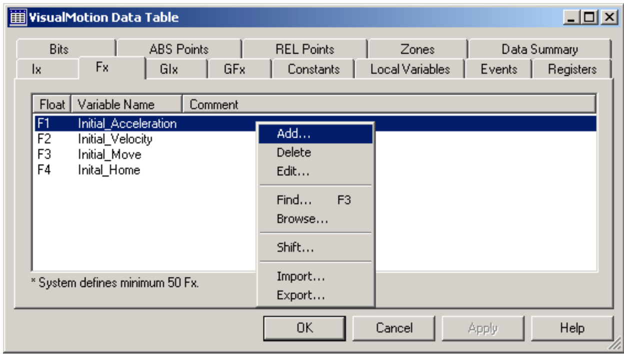
A fourth designator for a float and integer is a constant. Constant labels are accessed and modified in the VM Data table only, see Fig. 4-78

Constant

A constant is a float or integer that is fixed in the control. Unlike standard float and integers, the value assigned to a constant can not be changed after the project has been downloaded to the control.

Adding Variables

A convenient method for programming with VisualMotion is to declare all variables in the VM Data Table before creating the icon program. Select the VM Data Table button ( ) from the VisualMotion tool bar or select ***Edit*** ⇒ ***VM Data…*** to open the *VM Data Table* window, see Fig. 4-78. After selecting the appropriate tab in the *VM Data Table* window, right click in the field to add an item or right click on an item in the field to edit it. With the VM Data Table complete, variables can be accessed in the icon setup windows by selecting the browse button.



VM_Data_Table.tif

Fig. 4-78: Sample Program Floats in VM Data Table

For the sample program in section 4.10, the following four floats will be defined:

- F1 = Initial_Acceleration
- F2 = Initial_Velocity
- F3 = Initial_Move
- F4 = Initial_Home

Note: When creating a label for a variable, do not use VisualMotion keywords or icon labels.

Editing Variables

Variable numbers, labels, and values are listed in the *Data Editor* window in VisualMotion Toolkit, which is opened by selecting ***Data ⇒ Variables***. The window lists all program and global floats and integers, see Fig. 4-79. Each of these can be edited by double-clicking them. Modifications to variables are read immediately by the control and take effect when the program flow reaches the icon the variable is assigned to.

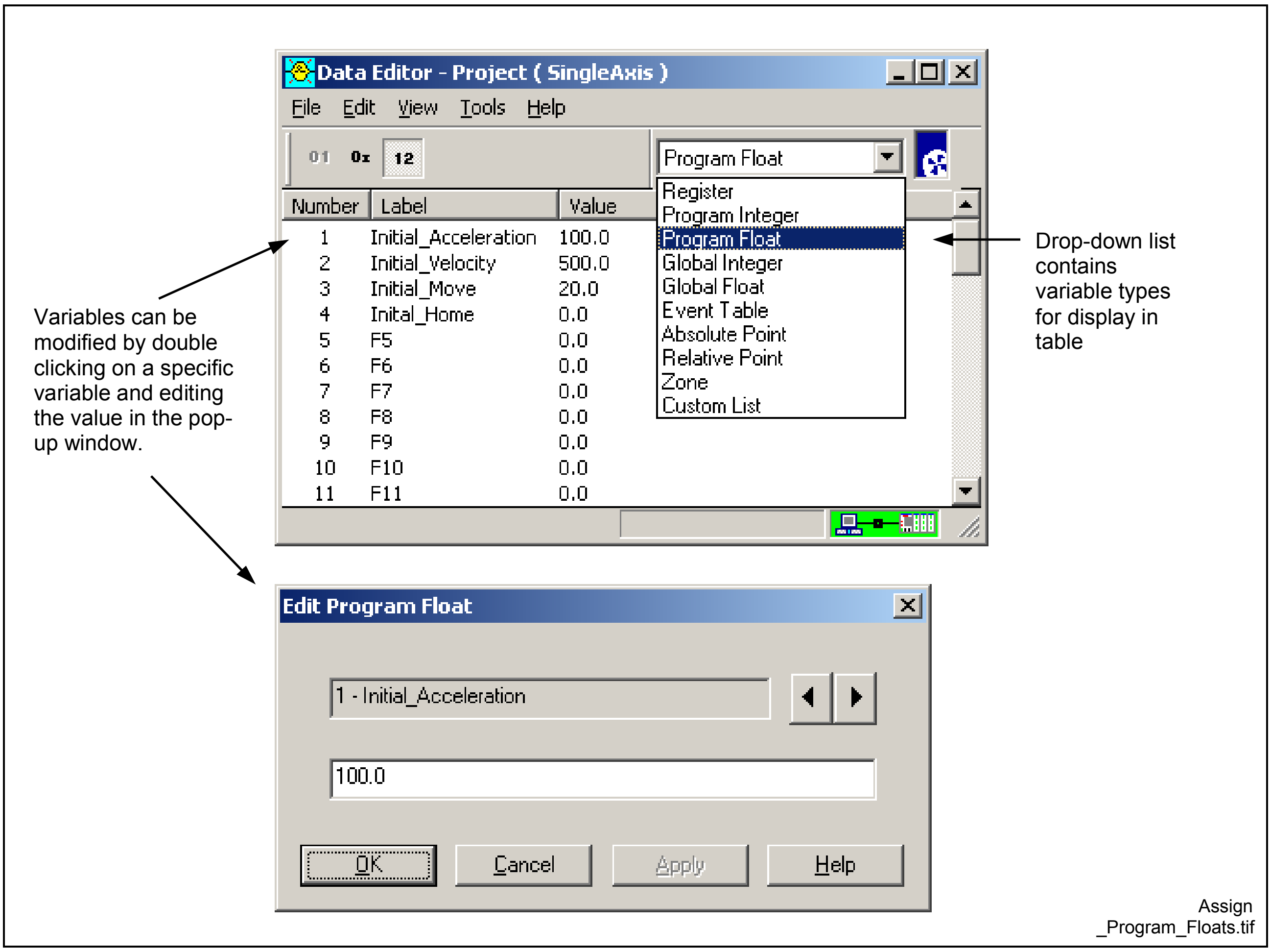


Fig. 4-79: Viewing and Editing Variables

4.10 Single Axis Sample Project

The sample project illustrated in this section is a single axis program that defines movement of an axis. The following table lists the programming icons that will be used for this project example and the task in which they will be used.

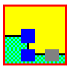
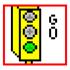
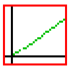
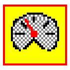
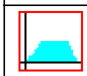
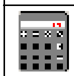
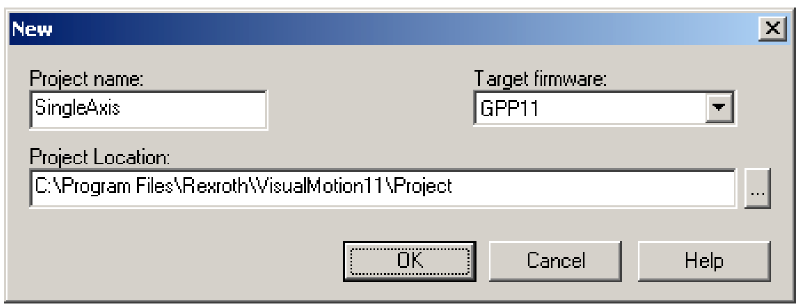
Icon	Description	Task
	Start icon used in every icon workspace to show the beginning of the program flow.	Initialization, Task A and Subroutine
	Subroutine icon used to call a subprogram in the project	Initialization
	Home icon used to command a homing procedure from drives connected to motors with a single turn encoder.	Task A
	Go icon used to command an axis that is initially halted.	Task A
	Acceleration icon used to command a acceleration rate to an axis	Task A
	Velocity icon used to command a velocity rate to an axis	Task A
	Move icon used to command relative or absolute positions to an axis	Task A
	Wait icon used to monetarily stop program flow until the specified condition is met	Task A
	Branch icon used to redirect program flow based on specified condition	Task A
	Calc icon used to perform calculation for the project.	Subroutine
	Finish icon used to indicate completion of program flow	Initialization, Task A and Subroutine

Table 4-28: Single Axis Sample Project Icons

Step 1: Creating a New Project

The following steps outline the initial start and preparation of a VisualMotion 11 project:

1. Start VisualMotion Toolkit by selecting ***Start*** ⇒ ***Programs*** ⇒ ***Rexroth*** ⇒ ***VisualMotion*** ⇒ ***11VRS*** ⇒ ***Icon Editor*** or double click on the VisualMotion Windows desktop Icon and select *VMT*PC-INB-11VRS*.
2. Name the project **SingleAxis**, select **GPP11** as the target firmware and select a project location on the hard drive.



NewProject.tif

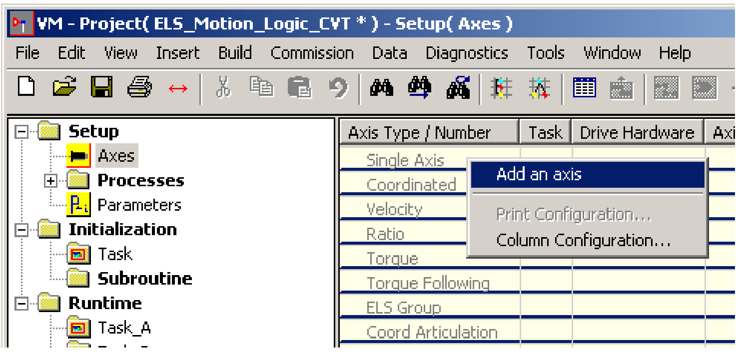
Fig. 4-80: New Project

Step 2: Setup Axes

Before placing icons in the initialization task or in any of the main task A-D or subroutines, setup all axes and any processes such as ELS that will be used in the program. Refer to Setup Axes and Setup Processes on pages 4-14 and 4-25 for details.

Note: The **Axis** icon has been removed from the initialization palette for G*P 11 projects. All axes used in a G*P 11 project are now configured from the *Setup* folder in the Project Navigator window.

1. To setup an axes, expand the *Setup* folder and select the **Axes** icon folder.
2. Right click over *Single Axis* and select **Add an axis**.



AddaSingleAxis.tif

Fig. 4-81: Adding a Single Axis

3. From the *Motion Selection* window, identify the following settings:
4. Axis (Number and Label)
5. Motion (Type and Control Task)
6. Drive (Hardware and Firmware)

Note: Drive hardware and firmware must be defined by the user. Refer to Specifying Axis Hardware and Firmware on page 4-16

7. After making the necessary settings, click the **Next >** button to continue to the *Axis Properties* window.
8. From the *Axis Properties* window, set the options according to the following figure:

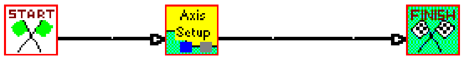
Option	**Setting**
Lagless	checked
Positioning Mode	Linear
Positioning Type	Absolute
Encoder	Motor
Units	inches

Table 4-29: Axis Properties Window Settings

9. After making the necessary settings, click the **Next >** button to continue to the *Configure Axis Probes* window.
10. This sample project will not use probe functionality. Click the **Finish** button to complete the initialization of the axis.

Step 3: Create the Initialization Task

The completed Initialization task will appear as shown in Fig. 4-82.



Init_Task_Icon_Prgrm.tif

Fig. 4-82: Initialization Task Icon Sequence

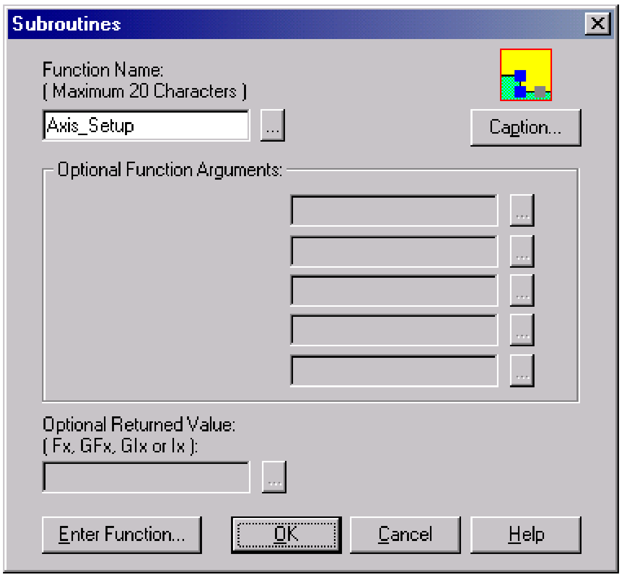
Select the Task folder under Initialization in VisualMotion's project navigator. The Initialization task will add the Start and Finish icons in the icon workspace. All tasks and subroutines must begin and end with Start and Finish icons.

Create a Subroutine

A subroutine can be used in the Initialization task to improve program readability.

To add a Subroutine Icon:

1. Click the Subroutine icon in the toolbar and place it between the Start and Finish icons.
2. Type the name of the subroutine in the *Subroutines* window.



Subroutine_Name.tif

Fig. 4-83: Subroutine Window

3. Click on the **Enter Function**… button to enter the initialization subroutine icon workspace.
4. Place a Calc icon for each variable in the icon program after the Start icon and set the following values:
   - **Resultant**: Use the browse button to select the variable for the variable name of one of the floats created in the VM Data Table. Refer to Adding Variables on page4-73 for details.
   - **Equation**: Enter the following values for the variables:

     Initial_Acceleration = 100

     Initial_Velocity = 500

     Initial_Move = 20

     Initial_Home = 0

Note: Refer to Adding Variables on page 4-73 for details.

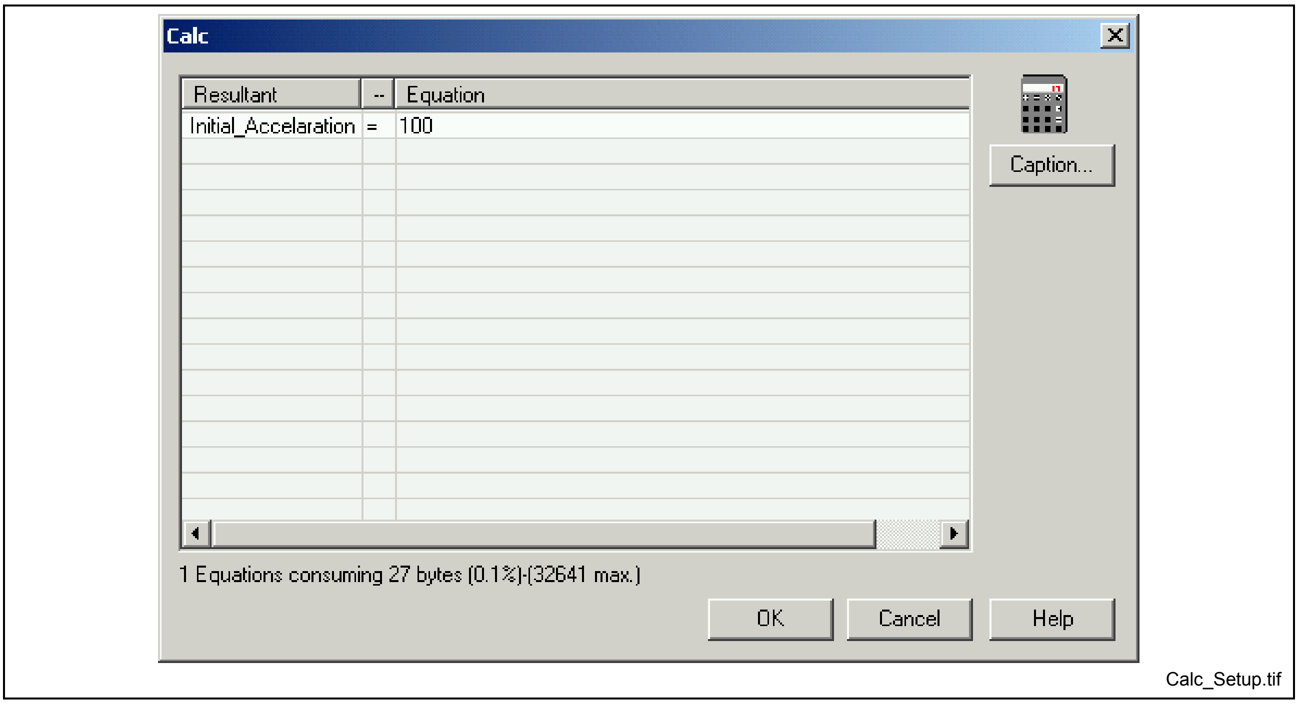


Fig. 4-84: Calc Icon Setup Window

Variable names can be created or selected from the VM Data Table by selecting the browse button to the right of the **Resultant** field.

Note: A caption for the Calc icon can be added in the *Calculation* window by selecting the ***Caption…*** button. Captions appear below the icon in the icon workspace if ***View ⇒ Icon captions*** is selected from the main menu in VisualMotion.

The following figure is an example of the completed *Axis_Setup* subroutine:

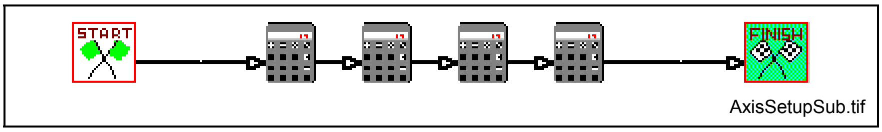


Fig. 4-85: Axis Setup Initialization Subroutine

Step 4: Create Task A

The completed Task A will appear as shown in Fig. 4-86.

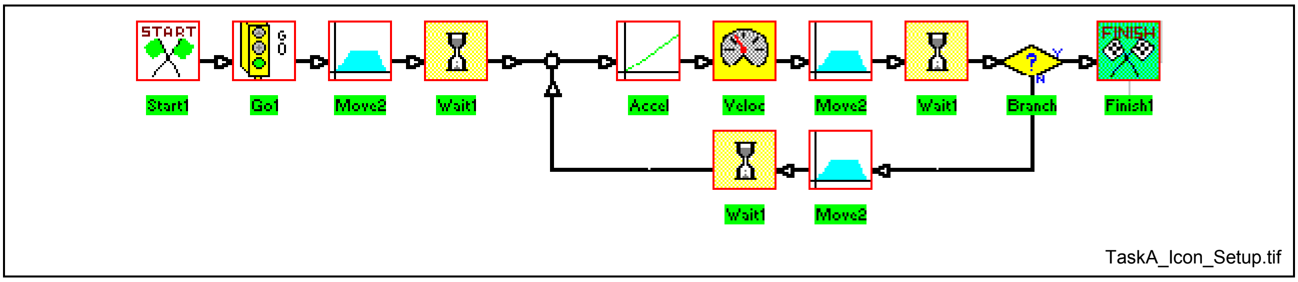


Fig. 4-86: Task A from Sample Program

To add a task to the project, switch to the Task A icon workspace:

1. In the Project Navigator, select **Task_A** under the Runtime folder.

Note: When opening a Task for the first time, VisualMotion Toolkit will prompt you to automatically place a Start and Finish icon in the icon workspace.

---

2. Select the Single Axis tab in the icon palette.

The next step is to set the drive homing command in the icon program. For single turn encoders, a Home icon is used to command the drive to run a homing routine. For multi turn encoders, a Move icon is used to move the axis to an absolute position, which will serve as the reference point for a move. Refer to Homing a Drive on page 4-83 for details.

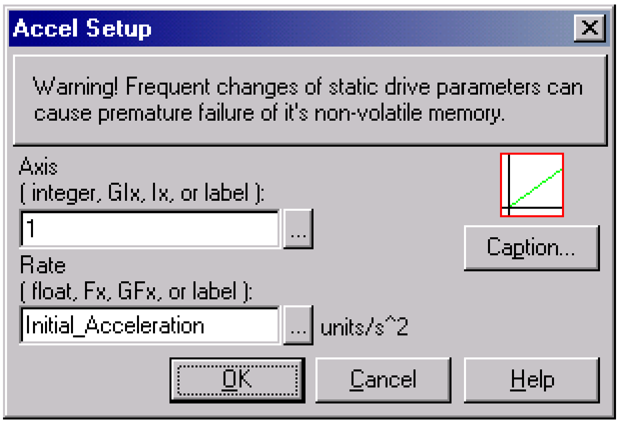
Homing an Absolute Encoder

For purposes of an example, this icon program will assume that a motor with an absolute encoder is being used.

3. Place a **Go** icon to the right of the Start icon, set the motion type to **Non-Coord**, and identify the axis as **1**.
4. Place a **Move** icon to the right of the Go icon, select **Absolute** for the move type and enter the axis number **1** and distance of **0**.
5. Place a **Wait** icon to the right of the Move icon, set *Wait for* to **Axis in Position** and identify the axis as **1**.

Programming Motion

6. Place the **Accel** icon to the right of the Wait icon and identify the axis as **1**.
7. Use the browse button to select the float for the **Rate** field. In this example, the **Initial_Acceleration** float was established with the value for the rate of acceleration.

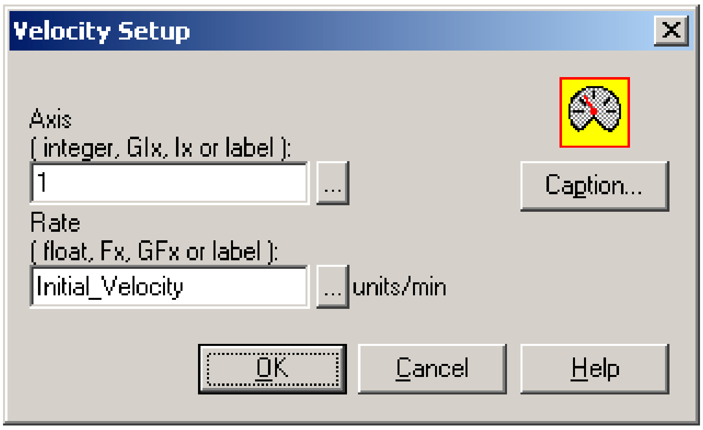


Accel_Setup.tif

Fig. 4-87: Accel Control Box

8. Place the **Velocity** icon to the right of the Accel icon. This icon sends the velocity rate that will be used in the move calculation to the drive .
9. Enter a **1** to specify the axis.
10. Use the browse button to select the float for the **Rate** field. In this example, the **Intial_Velocity** float was established with the value for the velocity rate.

The units for the velocity of the axis appear by default in the window according to the type of axis motion set in the axis icon.



Velocity_Setup.tif

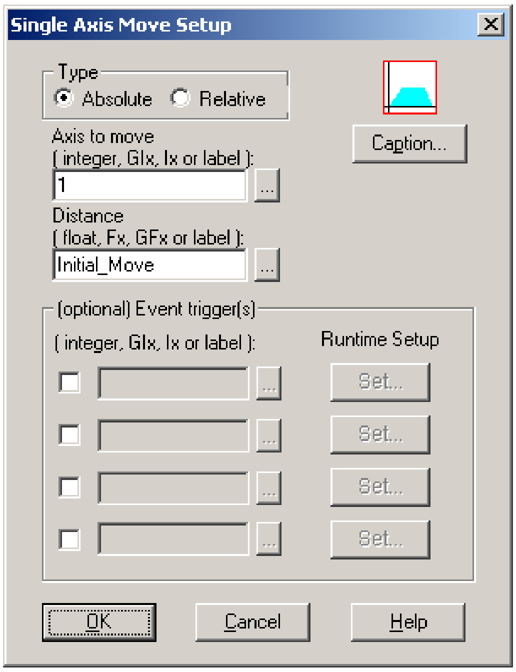
Fig. 4-88: Velocity Control Box

11. Place the **Move** icon to the right of the Velocity icon. This icon sets the distance that will be traveled by the specified axis.
12. Select **Absolute** for the move **Type**.

Note: A relative move is an incremental distance that is traveled every time the move icon is encountered in the program flow. An absolute move is an exact position that is reached when the move icon is encountered and is not repeated unless the absolute position changes.

---

13. Enter a **1** to specify the axis number.
14. Use the browse button to select the float for the **Distance** field. In this example, the **Initial_Move** float represents the distance the axis will move.



Move_Setup.tif

Fig. 4-89: Single Axis Move Setup

15. Place a **Wait** icon to the right of the Move icon. The task execution (program flow) is suspended at this point until the condition set in the Wait icon is true. In this sample program, the task waits until **Axis in Position**.

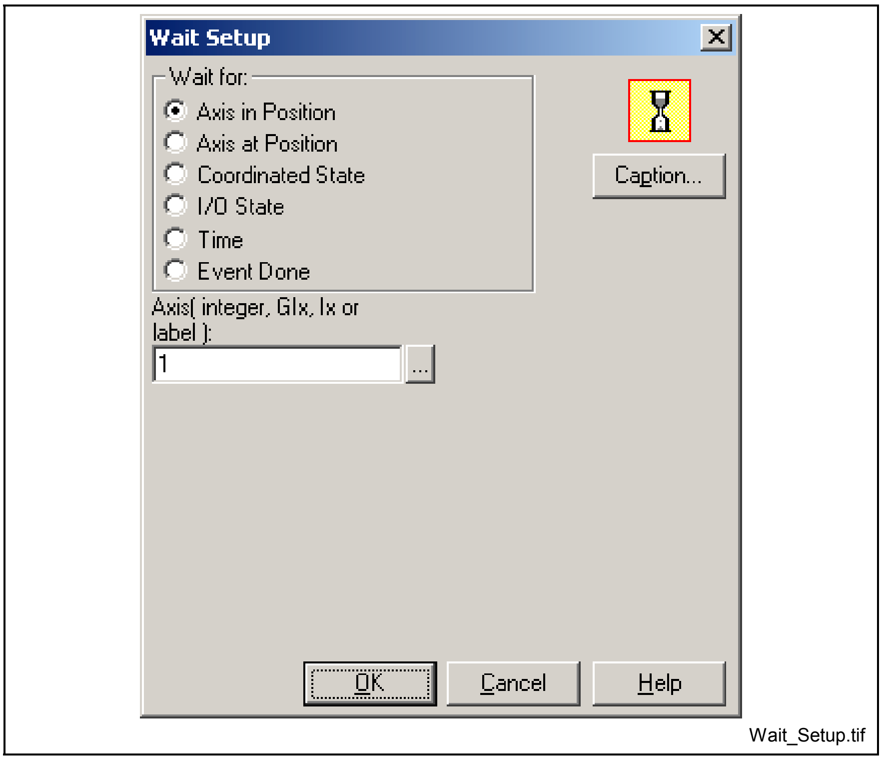


Fig. 4-90: Wait for Axis in Position

16. Place the **Branch** icon to the right of the Wait (Axis in Position) icon. This icon re-directs the program flow based on a true/false logical value. This creates a continuous loop within the program depending on the value of register 100 bit 9.

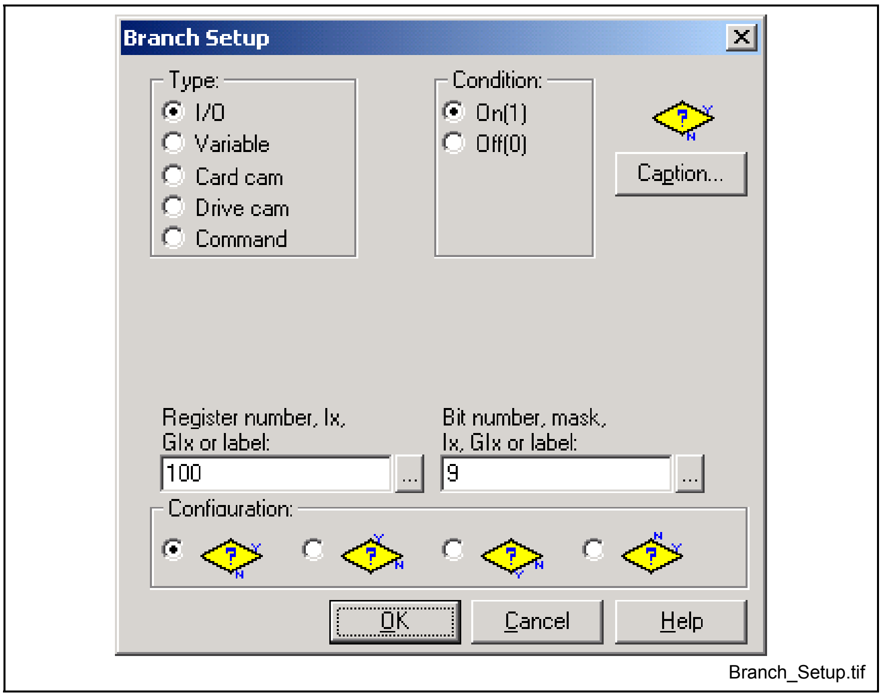
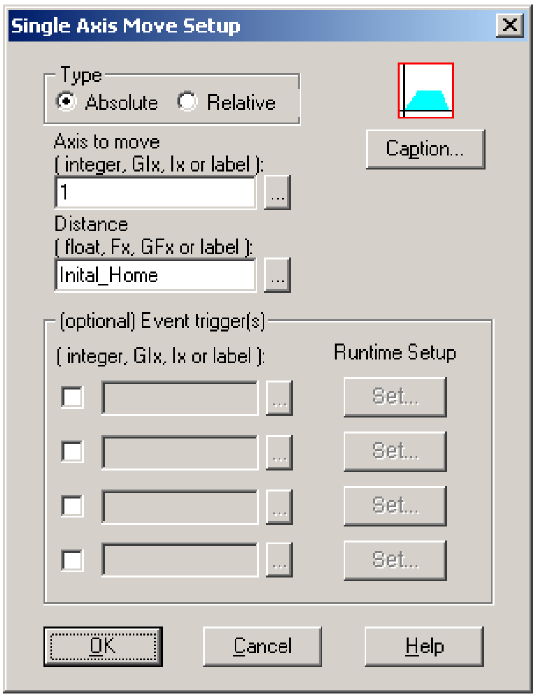


Fig. 4-91: Branch Setup

Note: The Branch icon will loop back to a specified icon until the branch condition is true. In this example, the Finish icon will not be encountered until register 100 bit 9 has the value 1.

---

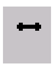
17. Place a second **Move** icon just below the previous Move icon. This icon sets the return distance for the specified axis.
18. Select **Absolute** for the move **Type**.
19. Enter a **1** to specify the axis number.
20. Use the browse button to select the float for the **Distance** field. In this example, the **Initial_Home** float represents the distance the axis will move.



Move2_Setup.tif

Fig. 4-92: Single Axis Move Setup

21. Place a **Wait** icon to the left of the second Move icon and select **Axis in Position**.

- Use the  icon to connect the icons.

To connect the icons, click once with the left mouse button on the first icon and then click on the next icon in the program flow. A line will join the icons with an arrow indicating program flow.

Note: If an error is made while connecting two icons, use the scissors  icon to remove the created connection line. Re-select the Line icon to continue connecting icons.

The completed project should appear as shown in Fig. 4-86.

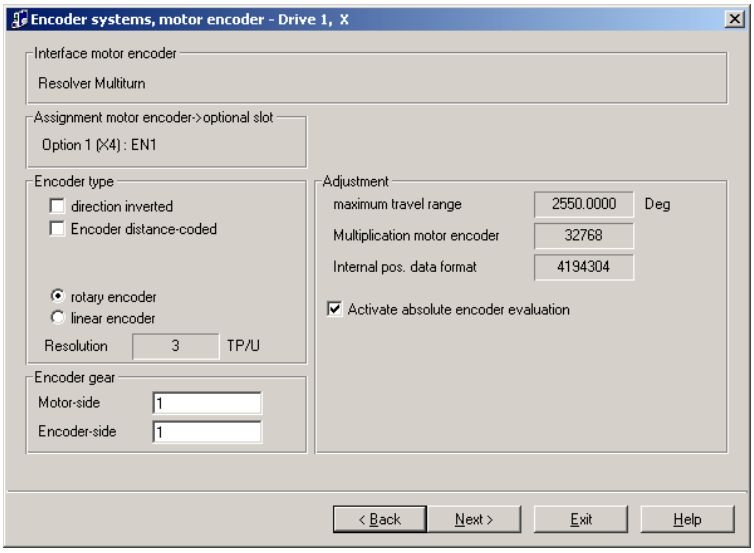
Determining an Encoder Type

Depending on the type of encoder on the motor, single turn or multi turn, a Home icon or Move icon will be used to issue the home command.

Use the following steps to determine encoder type on motor:

1. Close your VisualMotion project and reopen in Service Mode.
2. Select ***Commission ⇒ Drive Overview…*** from the main menu.
3. From the *Drive – Project* window, double click on the relevant drive.
4. From the menu in the *DriveTop Drive Status* window, select ***Drive Functions ⇒ Encoder systems ⇒ Motor encoder…***.

The *Encoder Systems, motor encoder* window will indicate the type of encoder in your system.



Encoder_Type.tif

Fig. 4-93: Encoder Systems, Motor Encoder Window

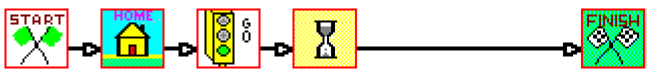
Homing a Drive

The homing procedure is an internal function of Rexroth's intelligent digital drives and requires only that VisualMotion send a home command to the drive. Refer to the drive help system for information about drive homing.

Note: Drive help system for supported drive firmwares can be optionally installed during the initial installation of VisualMotion.

Single Turn Encoder

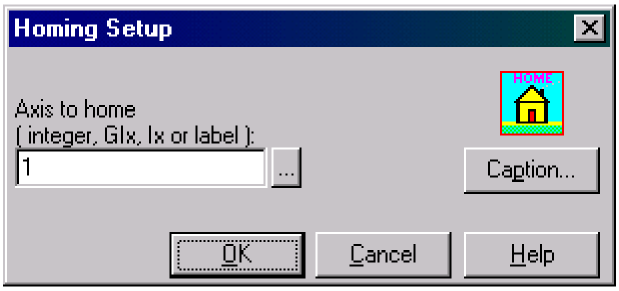
For a PPC with a single turn encoder, set the homing signal in the icon program using the following sequence of icons.



Single_Encoder_Homing.tif

Fig. 4-94: Single Turn Encoder Homing Sequence

1. Place a **Home** icon and enter a **1** in the **Axis to home** field.



Homing_Setup.tif

Fig. 4-95: Homing Setup Window

2. Place a **Go** icon to the right of the Home icon, set the motion type to **Non-Coord**, and identify the axis as **1**.
3. Place a **Wait** icon to the right of the Go icon, set *Wait for* to **Axis in Position** and identify the axis as **1**.

Note: A Wait icon is used to allow the axis to perform the homing routine before the Move command is executed.

Multi-turn Encoder

For multi-turn encoders, use the following icon setup for the homing routine. The sequence of icons is shown in the Fig. 4-86.

1. Place a **Go** icon to the right of the Start icon, set the motion type to **Non-Coord**, and identify the axis as **1**.
2. Place a **Move** icon to the right of the Go icon, select **Absolute** for the move type and enter the axis number **1** and distance of **0**.
3. Place a **Wait** icon to the right of the Move icon, set *Wait for* to **Axis in Position** and identify the axis as **1**.

Step 5 Import Default IndraLogic Project

An IndraLogic PLC project is a logic task that allows external devices to access user-defined control and status registers. The IndraLogic project is necessary for controlling external I/O devices through VisualMotion or to use the I/O Box utility for program control.

In this sample program, a default IndraLogic project will be used for controlling system registers through the I/O Box. With the default IndraLogic project, register 100 is mapped to the System_Control and TaskA_Control registers for program control.

Default I/O Mapping Project

During the installation of VisualMotion 11, a default IndraLogic project file (*Default_IOmapper.exp)* was installed under the folder structure *Rexroth\VisualMotion11\Param\Def100_IndraLogic*. In addition, register and bit label files (*Register.lbl* and B*it.lbl*) were installed for importing in the VisualMotion project.

Use the following steps to import the default IndraLogic I/O Mapping project into VisualMotion and IndraLogic:

1. Open the target VisualMotion project in offline.
2. Open the VM Data Table and import the "register.lbl" file into the Registers tab and the "bit.lbl" file into the *Bits* tab. To import, right click in the relevant tab window and select ***Import…***
3. Select ***Tools*** ⇒ ***Options…*** and open the *IndraLogic* tab.

4. Click the **Select Items to Share with IndraLogic** button and choose the following data types:
   a) System register and bits
   b) Task A-D control registers and bits
   c) Task A-D status registers and bits
   d) User Registers: IO_Box_inputs and IO_Box_outputs

Items can be selected individually or in groups by right clicking on the item name and selecting *Check items and subitems*.

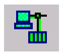
Note: For the default I/O mapping POU to work, **<u>Do</u>** **<u>Not</u>** *Prepend Data Type to Label* available under the *Options* menu.

---

5. Save the VisualMotion project.
6. Start ***IndraLogic*** from the ***Commission*** menu.
7. From IndraLogic, select ***Project*** ⇒ ***Import,*** locate and open the file, "default_IOmapper.exp". This file include the "Default_IOM" POU.
8. From IndraLogic, select the *Resources* tab and double-click on ***Task Configuration***.
9. In the Task Configuration window, right click on ***Task Configuration*** and select **Append Task** to add a NewTask.
10. Rename the task and enter the following information:
    a) Configure the task type as **cyclic**
    b) Enter an interval of **T#2ms** for the 2ms cycle time
    c) **Activate watchdog** with the same time period as in step b)
11. Right click on *NewTask* tree selection, choose **Append Program Call** and enter the POU "***Default.IOM***" as a program call.
12. Select ***Online*** ⇒ ***Login*** to download the IndraLogic project to the control.
13. Next, select ***Online*** ⇒ ***Run*** to start the PLC program.
14. Finish by creating a boot project. Select ***Online*** ⇒ ***Create boot project*** to save the PLC program to flash.

Switching a Project to Online Mode

With the project completed, you need to download the project file to the control. Project download occurs when you go online and involves synchronizing the project and control data. To download your project data:

1. Click the Toggle Online/Offline Mode button  in the VisualMotion toolbar or by selecting ***File*** ⇒ ***Online*** from the VisualMotion Toolkit's main menu.
2. Click ***Yes*** when prompted to save modifications to your program.
3. Click ***OK*** in the *Synchronize Project Data* window to allow VisualMotion to synchronize the data in the control with data in the project.

Note: If a difference is detected between the axes that are setup in the project with those on the Sercos ring, the *Offline/Online Drive Difference Detected* window opens. Refer to section 13.2, Online/Offline …F9, in volume 2 for details.

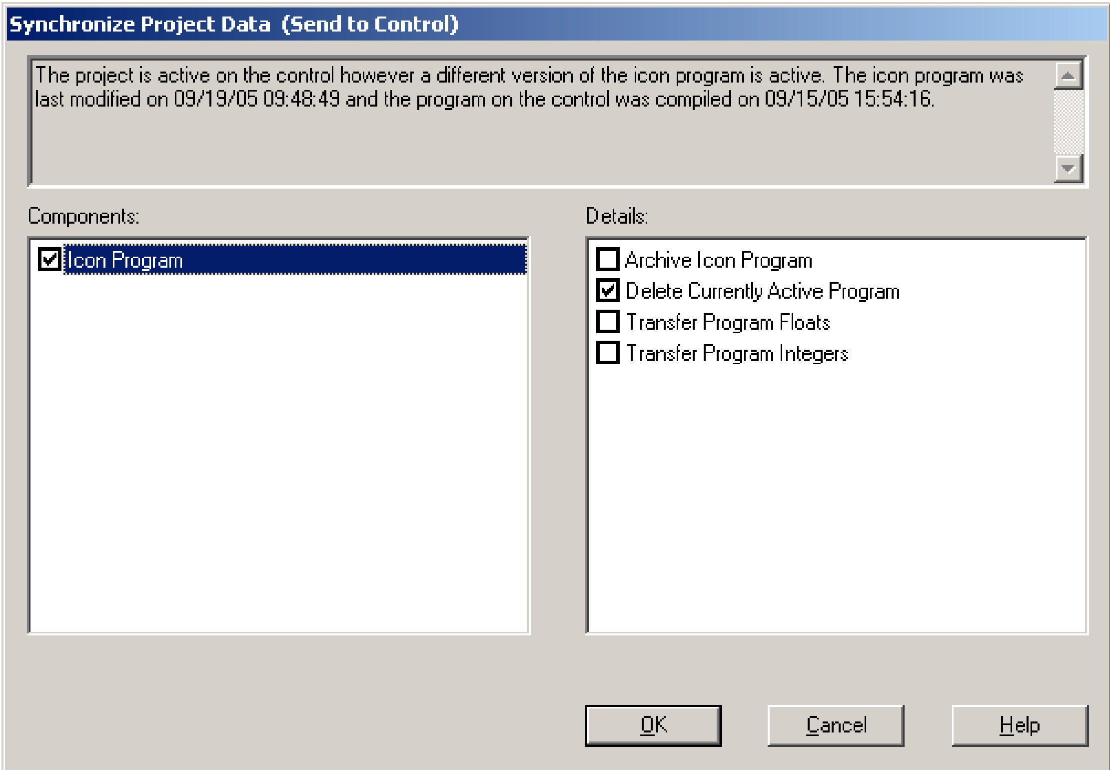


Fig. 4-96: Synchronize Project Data

The program should now be online, as indicated by the online symbol in the lower right corner of the *VM Project* window.

Saving a Project

Icon programs are saved in the project folder on the computer. When a project goes online, the icon program is automatically saved to the project folder on the computer. To save an icon program when offline, use the ***Save Program*** option in the VisualMotion ***File*** menu.

4.11 Activating a Project

For this sample project, the I/O Box tool (optionally installed) is used to perform the basic activation of control registers to start the VisualMotion project. The I/O Box will be using the default I/O mapping project imported and started in IndraLogic.

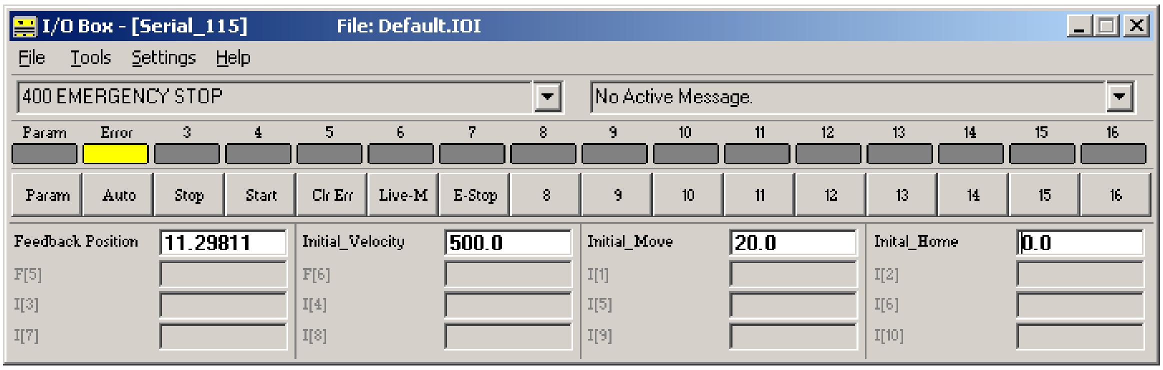
I/O Box

The I/O Box is a VisualBasic® interface designed for activating and monitoring an icon program. It is installed by default during the VisualMotion installation and is accessed through the main menu in VisualMotion Toolkit.

To open the I/O Box tool:

1. Select ***Tools ⇒ Registered Tools ⇒ IoBox***
2. Click **Accept** in the *Load I/O Box* window.

The I/O Box window opens with 400 EMERGENCY STOP displayed, as shown in Fig. 4-97.



IOBox_New.tif

Fig. 4-97: I/O Box Window

To activate the icon program with the I/O Box, use the following sequence:

3. Clear the error by selecting the **E-Stop** button and then the **Clr Err** button. The control should be in Manual Mode.
4. Select the **Auto** button to put the control in auto mode.
5. Select the **Stop** button.
6. Select the **Start** button to activate the icon program.

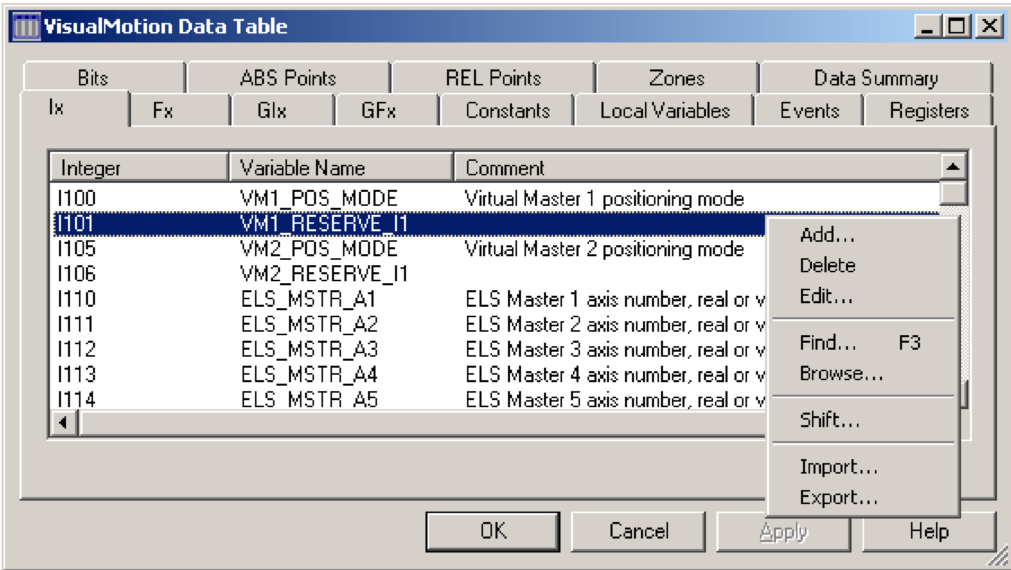
Note: The system can be switched to parameter mode by selecting the *Param* button.

4.12 VM Data

The VM Data Table is used to Add, Delete, Edit, Find, Browse, Shift, Import and Export the following VisualMotion data types.

- Program Integer (Ix) and Floats (Fx)
- Global Integer (GIx) and Floats (GFx)
- Constants
- Local Variables
- Events
- Registers
- Bits (register)
- Points (ABSolute and RELative)
- Zones

Note: The VM Data Table can also be opened by clicking on the VM Data icon ( ). The VM Data Table can be resized by clicking and dragging any corner.



VMdataTable.tif

Fig. 4-98: VisualMotion Data Table

Add and Modify VM Data

Right clicking in the display area of any VM Data type, opens the pop-up window in Fig. 4-99. From this window, the functions displayed can be performed under any tab, except for the **Local Variables** and **Data Summary** tabs.

Multiple Item Selection

Items displayed in the VM Data Table can be selected using the following standard Windows™ keyboard and mouse selection methods.

- *Ctrl + A* - selects all the items in the current window.
- *Ctrl + Left Mouse button* – selects items one at a time while holding down the Ctrl key.

- *Shift + Left Mouse button* – selects a group of items while holding down the Shift key.. Using the left mouse button; select the first items, scroll down and then select the last item.

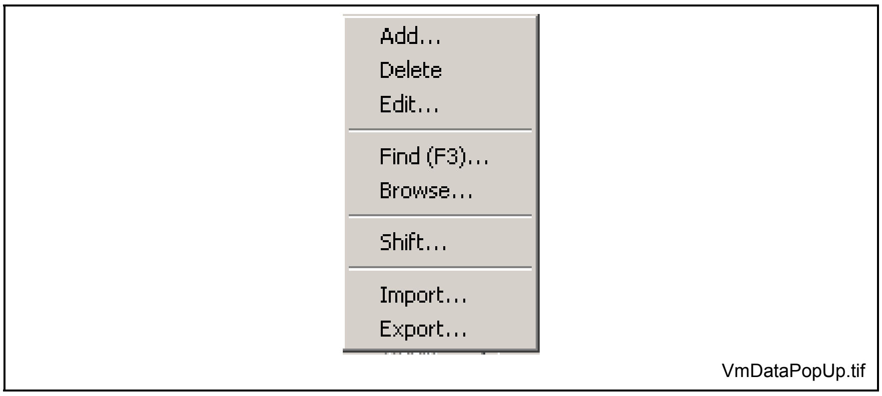


Fig. 4-99: VM Data Pop-up Window

Add, Delete and Edit

These selections allow the user to **Add** a new item in the VM Data Table, **Delete** or **Edit** an existing item.

Find (F3)

Selecting **Find** or pressing the **F3** key opens the *Find* window. From the open window, any items can be found in the current VM Data window by entering a matching or partially matching **Number**, **Variable Name** or **Comment**.

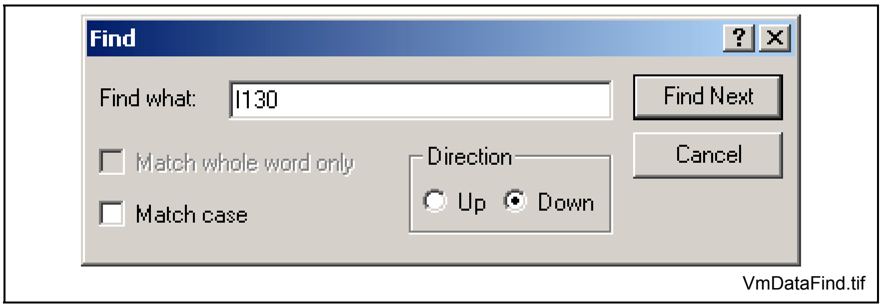


Fig. 4-100: VM Data Find

Browse

Selecting Browse opens the Browse Project for Data Usage window. Refer to section 10.2 regarding *Browsing a Project for Data Usage and Icon Flow* for details.

Shift

The *Shift Indexes* window is used to shift a single item or a group of items **Up (-)** or **Down (+)** by the amount entered in the **Increment** field. To shift an item(s), select the item(s), enter the number of increments to shift and press the ***OK*** button. The **Up** shift selection, subtracts the **Increment** value from the current index number. The **Down** shift selection, adds the **Increment** value to the current index number.

Note: If the new location is already occupied by an existing item, VisualMotion issues an error and shift request is canceled.

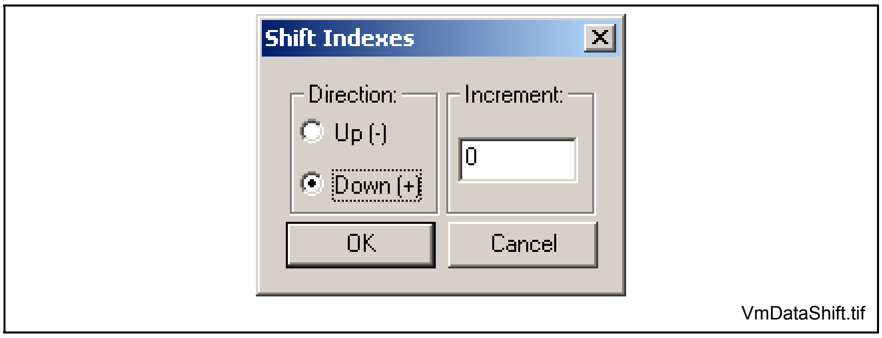


Fig. 4-101: VM Data Shift

Import and Export

This feature is used to import or export the selected items from the VM Data table to a file with a label (*.lbl) extension.

Sorting Numbers, Names or Comments

Any VM Data Table type *Number, Name* or *Comment* can be sorted in either ascending or descending order by clicking on the column header.

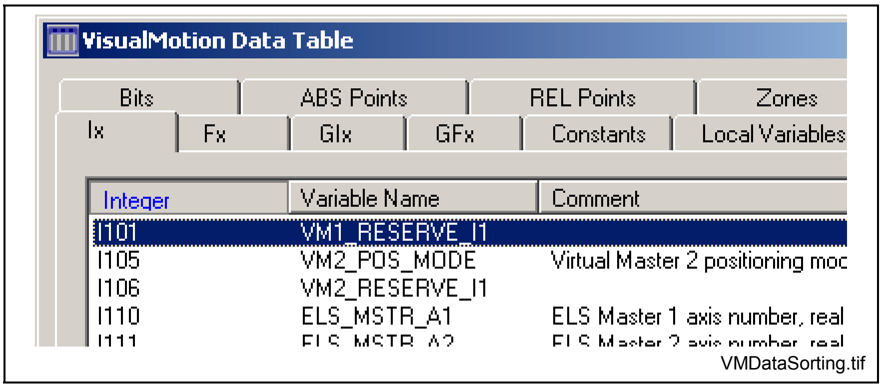


Fig. 4-102: Ascending or Descending Listings

Using VisualMotion Keywords as Names

VisualMotion has a number of keywords, which are used for command instructions. These keywords cannot be used as names (labels). If a keyword is used as a name, VisualMotion will issue the error "Name is a Keyword!" when the user tries to save the name. Following is a list of the VisualMotion keywords:

Current VisualMotion Keywords				
AA_XFER	CALL	EVENT	MOVER_PATH	SET_PARAM
ABORT	CALL_FUNC	EVENT_DONE	MSG_DIAG	SET
ABORT_PATH	CAP_ENABLE	EVENT_ENABLE	MSG_STATUS	START
ABS_INIT	CAP_SETUP	EVENT_END	PARAM_BIT	STOP
ACCEL	CLEAR	EVENT_START	PARAM_INIT	STOP_PATH
AXES	CNC_CIRCLE	EVENT_WAIT	PATH_CALC	Task_A
AXES_GROUP	COMP	EVT_INIT	PATH_HOLD	Task_B
AXES_GROUP1	CONVEYOR_INIT	FUNC_ARG	PID_CONFIG	Task_C
AXIS_ATPOSITION	CONVEYOR_LIST	FUNC_END	PID_CONFIG1	Task_D
AXIS_EVENT	CONVEYOR_PICK	FUNC_START	PLS_WRITE	TEST
AXIS_WAIT	DATA_SIZE	GET_PARAM	PLS_INIT	V_MASTER
BNE	DEC	GO	PLS1_INIT	VAR_INIT
BEQ	DECEL	HOME	POSITION	VEL
BGT	DRVCOM	INC	RA_XFER	WAIT
BLT	DRVCOM_STATUS	INIT	RATIO	WAIT_IO
BGE	EE_XFER	INITIALIZATION	READ	WAIT_PATH
BLE	ELS_ADJUST	KINEMATIC	REGISTRATION	WRITE
BRA	ELS_ADJUST1	KINEMATIC1	REL_INIT	ZONE_INIT
BROADCAST	ELS_GROUPM	LOCAL_VAR	RESUME_PATH	ZZ_XFER
CALC	ELS_GROUPS	MESSAGE	RETURN	
CAM_ADJUST	ELS_GROUPS1	MODE	ROBOT_ORIGIN	
CAM_ACTIVATE	ELS_INIT	MOVE_JOINT	ROBOT_TOOL	
CAM_BUILD	ELS_MODE	MOVEA_AXIS	ROTARY_EVENT	
CAM_BUILD1	ELS_MASTER	MOVEA_CIRCLE	RR_XFER	
CAM_ENGAGE	ELS_MASTER1	MOVEA_PATH	SEQ_LIST	
CAM_INDEX	ELS_STOP	MOVER_AXIS	SEQ_STEP	
CAM_STATUS	END	MOVER_CIRCLE	SEQUENCER	

Table 4-30: Current VisualMotion Keywords

Integers (Ix) and Floats (Fx)

Program integers and floats are displayed listing the *Number*, *Variable Name* and *Comment* of all defined integers and floats in the current program. Integers and floats are variables that can be used within any task and are associated with the project for which they were created.

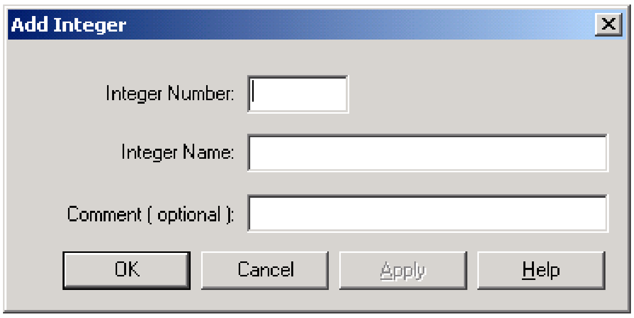
Note: The value of all integers and floats is saved with the project even during a power lost condition.

When creating a new project, VisualMotion automatically allocates memory for a minimum of 50 integers and 50 floats. These integers and floats are not displayed in the VM Data Table until a number and name (label) has been assigned.

System default integers and floats, such as those used for ELS, will appear in the VM Data Table if assigned in their corresponding icons.

Add an Integer or Float

Right click and select **Add…** to open the *Add Integer(Float)* window. Enter a number and name for the integer or float. The **Comment** field is optional. Since VisualMotion allocates memory for the first 50 integers and floats, any number entered that is greater than 50 will become the top-level number with additional memory allocated.



AddInteger.tif

Fig. 4-103: Add Integer or Float

For example:

If the number 200 is entered, VisualMotion will allocate memory for 200 integers. Even if only one integer or float is displayed in the VM Data Table. Each integer and float is allocated 4 bytes of memory. The total memory allocated for integers and floats can be viewed by selecting the *Data Summary* tab.

Note: Edit or Delete an integer or float by first select the item.

Since the VM Data Table only displays the number and name (label) of an integer or float, the actual value of each integer or float in a project can be viewed and defined by selecting ***Data ⇒ Variables…*** (In online mode) and selecting the appropriate *Type*.

Global Integers (GIx) and Floats (GFx)

Global integers and floats are displayed listing the *Number*, *Variable Name* and *Comment* of all defined global integers and floats. Global integers and floats are variables that can be used in any task across different programs resident in the control. The names (labels) assigned to global variables are saved with each program while their values are saved to the control's memory.

Note: The value of global variables is not saved if the control loses power unless the values are flashed using the *Save Global Variables Command* (**C-0-0082**). Refer to section 8.2, *Data Editor*, regarding *Variables* for details.

System default global integers (GIx) are assigned, with fixed memory, for all programs. These global integers are reserved for counter, timer and shift register output functions.

The maximum number of global variables in a program is fixed and defined by the following control parameters:

- **C-0-0080**, Maximum number of global integers
- **C-0-0081**, Maximum number of global floats

The default numbers are 512 for global integers and 256 for global floats.

Add a Global Variable

- Right click and select **Add…** to open the *Add Global Integer(Float)* window in offline mode. Enter a number and name for the global variable. The **Comment** field is optional.

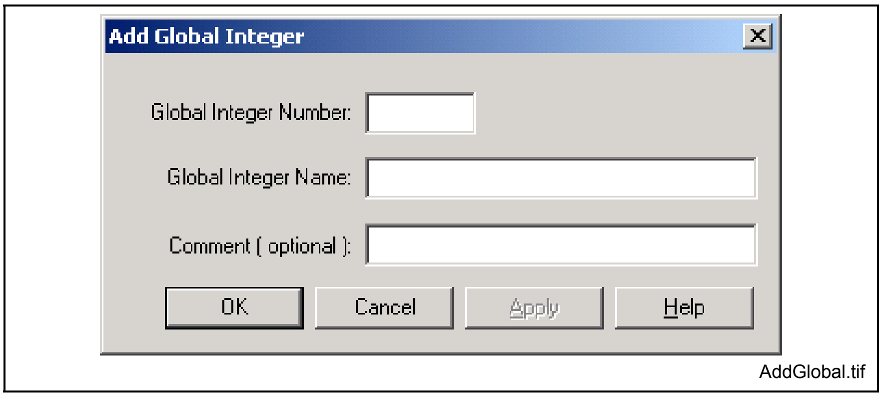


Fig. 4-104: Add Global Variable (GIx or GFx)

Since the VM Data Table only displays the number and name (label) for global variables, the actual value for each global variable in a program can be viewed and defined by selecting ***Data ⇒ Variables…*** (In online mode) and selecting the appropriate *Type*.

Constants

Constants are displayed listing the **Constant Value**, **Constant Name** and **Comment** of all defined constants in a program. Constant names are created to represent a fixed numeric value in an icon program.

The user can use a name assigned to a value instead of the value itself. The benefit of this would be; for example, if the same numeric value is used within multiple icons, the user need only modify the value in the VM Data Table instead of a value in each icon.

Add a Constant

Right click and select **Add…** to open the *Add Constant* window in offline mode. Enter a numeric value and name for the constant. The **Comment** field is optional.

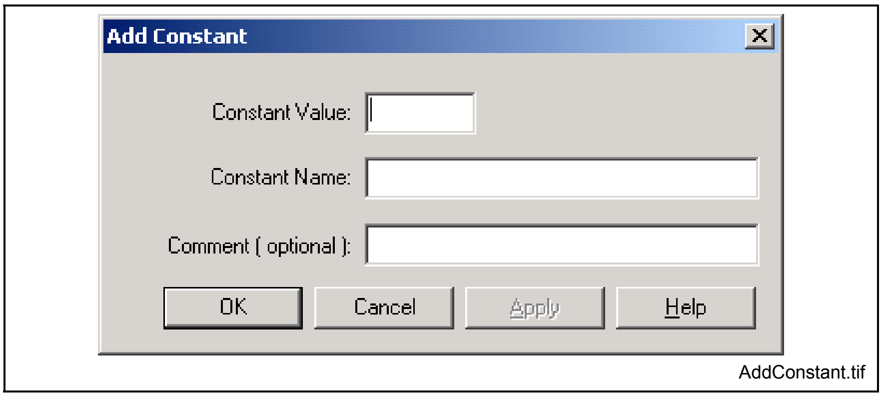
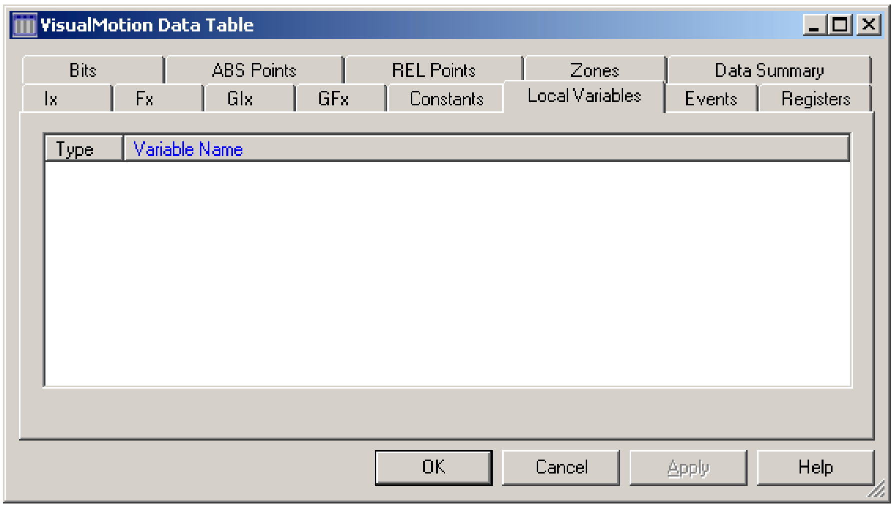


Fig. 4-105: Add a Constant

Local Variables

Only local variables that are used in the current task, subroutine or event are displayed in the VM Data Table. Local variables ***cannot*** be added in the VM Data Table. Local variables are added in the *Start* icon.



LocalVar.tif

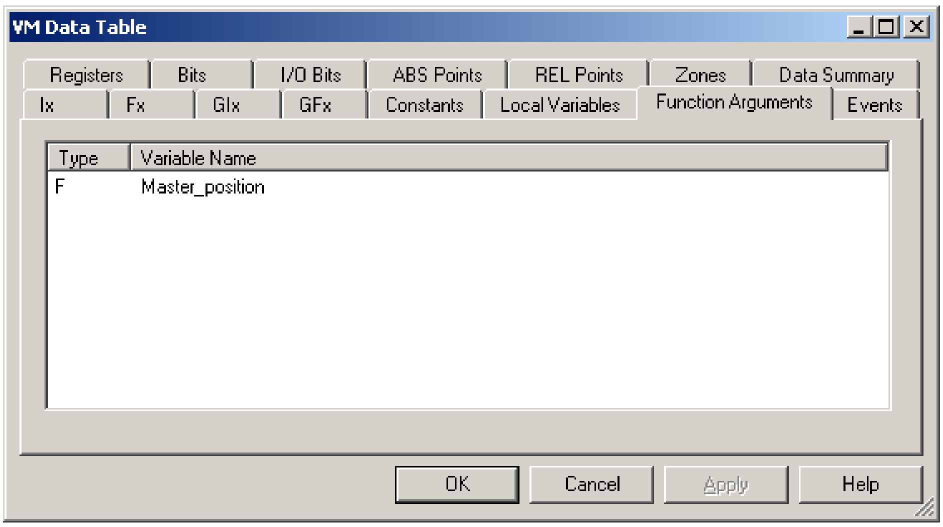
Fig. 4-106: Local Variables

Since the VM Data Table only displays the number and name (label) for local variables, the actual value for each local variable in a task, subroutine or event can be viewed and defined by selecting ***Data ⇒ Variables…*** (In online mode) and selecting the *Type: (Task A Local Variables)*.

Function Arguments

Only function arguments that are used in the current subroutine are displayed in the VM Data Table. Function Arguments ***cannot*** be added in the VM Data Table. Function arguments are added in the subroutines' *Start* icon.

Note: Function arguments are only displayed if they exist in the currently selected subroutine.



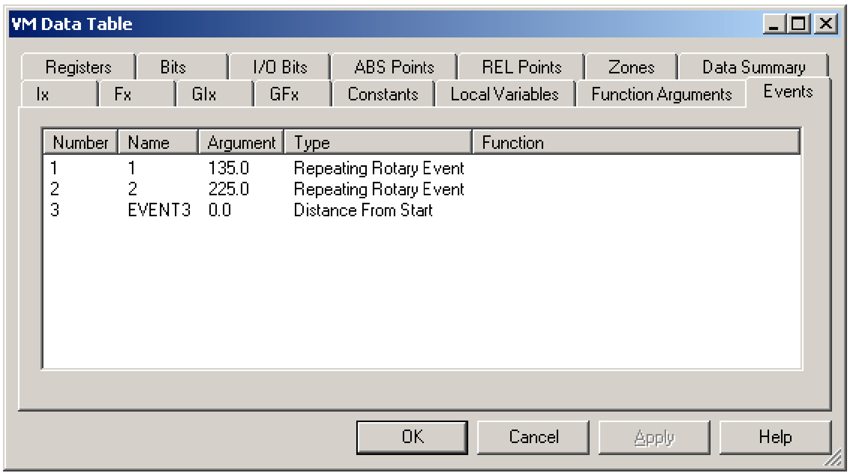
FuncArgu.tif

Fig. 4-107: Function Argument

Since the VM Data Table only displays the number and name (label) for function arguments, the actual value for each function argument in a program can be viewed and defined in the Start icon of the subroutine where it resides.

Events

Events are displayed listing the *Number*, *Name*, *Argument*, *Type* and *Function* of all events in the current project.

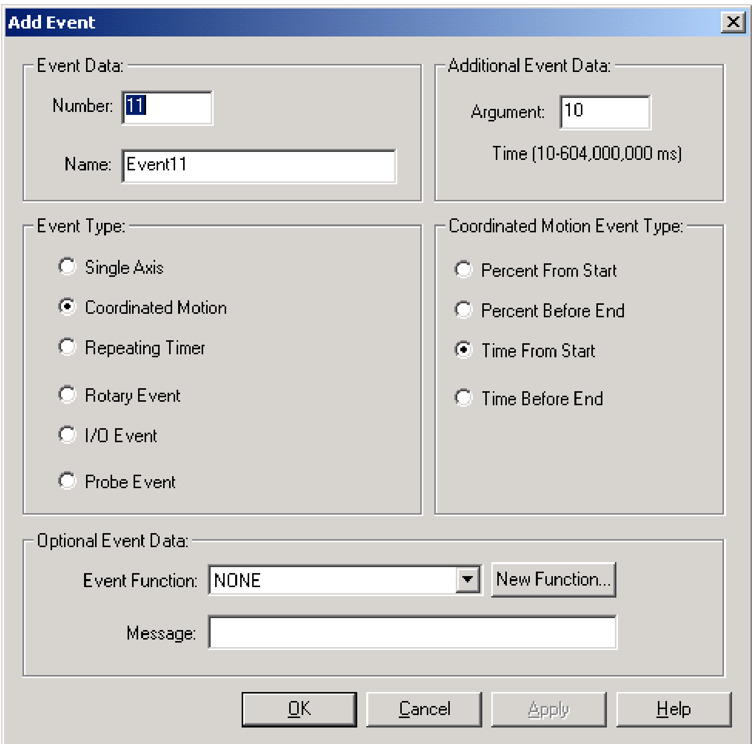


Events.tif

Fig. 4-108: Events

Add an Event

Right click and select **Add…** to open the *Add Event* window in offline mode. From this window, the user can create events, select the event type, initialize an argument value and assign an event function. For detailed examples on creating events and event functions, refer to chapter 5.2, Events.



AddEvent.tif

Fig. 4-109: Add an Event

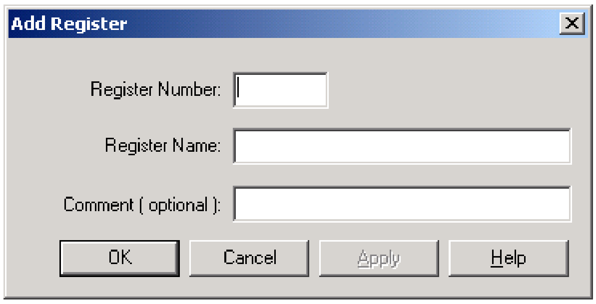
Rexroth
Bosch Group

Registers

Registers are displayed listing the **Register Number**, **Register Name** and **Comment** field for all registers that have been assigned a name other than default name (Reg_070). The first 100 registers are reserved for control and system functions, while others are recommended as defaults for applications such as ELS.

Add a Register

Right click and select **Add…** to open the *Add Register* window in offline mode. Enter a number and name for the register. The **Comment** field is optional. A VisualMotion system can contain up to a maximum of 1024 registers.



AddRegister.tif

Fig. 4-110: Add a Register

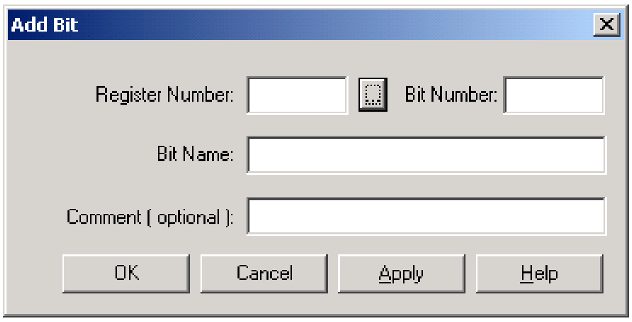
Since the VM Data Table only displays the number and name (label) for assigned registers, the actual value of each register in a program can be viewed and defined by selecting ***Data ⇒ Registers*** (In online mode).

Bits

The Bits window displays the **Register-Bit Number**, **Bit Name** and **Comment** field for all register bits that have assigned a name other than the default name (Bit_01). Only those bits that have been assigned names will be displayed.

Add a Bit

Right click and select **Add…** to open the *Add Bit* window in offline mode. A bit name is assigned to a register number. Enter a Register Number, Bit Number and Bit Name. The **Comment** field is optional. A VisualMotion register can contain up to 16 bits. Only those bits assigned to a register number and given a name will be displayed in the VM Data Table.



AddBit.tif

Fig. 4-111: Add a Register Bit

The VM Data Table only displays the number and name (label) for assigned register bits. The actual value of each register bit in a program can be viewed and defined by selecting ***Data ⇒ Registers*** (In online mode) and double-clicking a register number to open the bits window.

Points (ABS and REL)

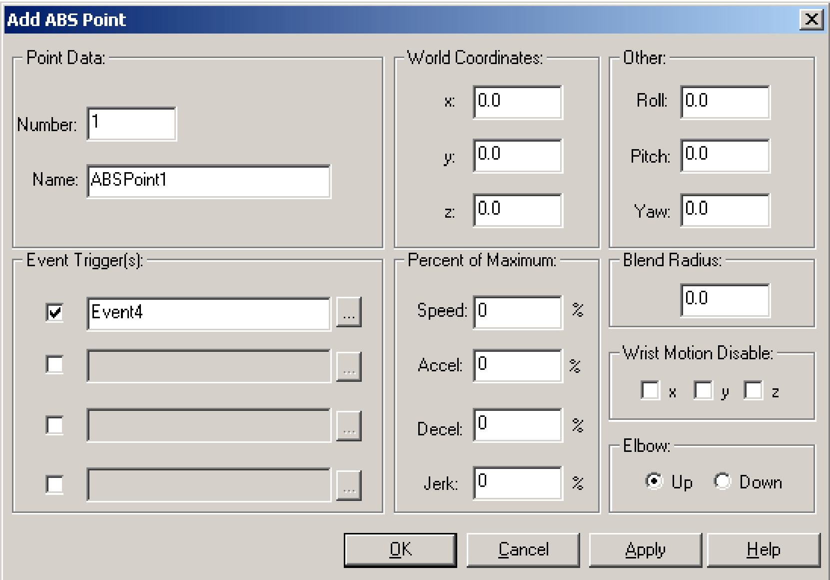
Absolute and Relative points are displayed for all defined points in the current project. Points are not displayed in the VM Data Table until a number and name (label) has been assigned.

Refer to section 8.2, *Data Editor*, regarding *Points* for details

Add an Absolute or Relative Point

Right click and select **Add…** to open the *Add ABS(REL)* window. From this window, the user can assign a number and name, enable and disable events, initialize coordinate and movement values.

Note: VisualMotion allocates memory for points based on the largest number assigned to a point.

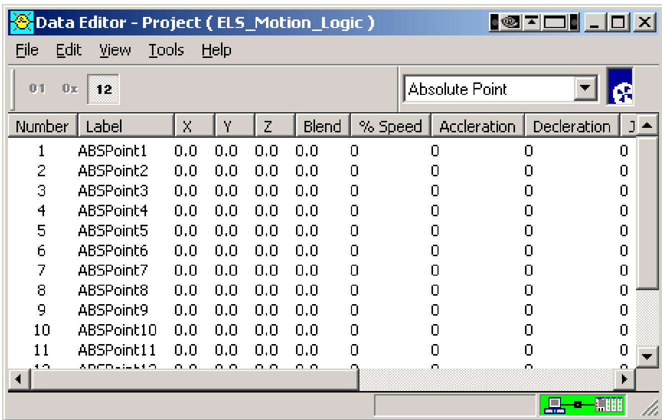


AddPoints.tif

Fig. 4-112: Add ABS(REL) Point

For example:

If the number 200 is entered, VisualMotion will allocate memory for 200 points. Even if only one point is displayed in the VM Data Table. Viewing points by selecting ***Data ⇒ Points*** will display a list of 200 points. Each ABS or REL point is allocated 44 bytes of memory. The total memory allocated for points can be viewed by selecting the *Data Summary* tab.



DataPoints.tif

Fig. 4-113: Data Editor – Absolute or Relative Points

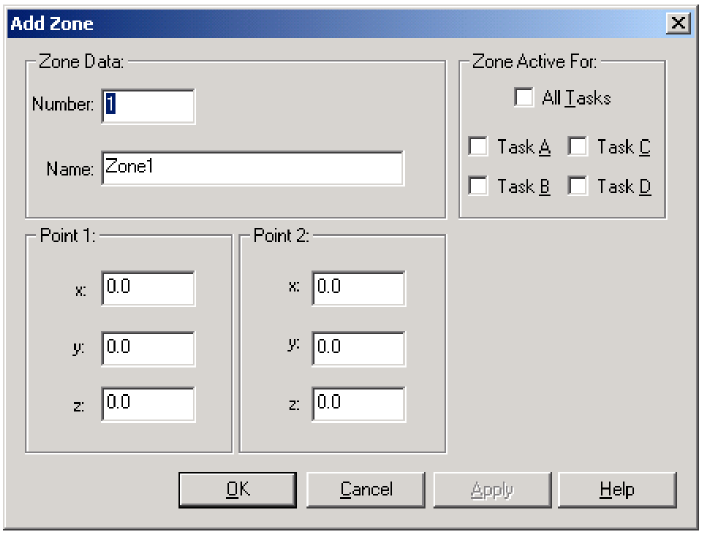
Zones

Zones are displayed for all defined zones in the current project. Zones are not displayed in the VM Data Table until a number and name (label) has been assigned.

Add a Zone

Right click and select **Add…** to open the *Add Zone* window. From this window, the user can assign a number and name, active zones for specific task and initialize (x, y, z) coordinate values for points 1 and 2.

Note: VisualMotion allocates memory for zones based on the largest number assigned to a zone.

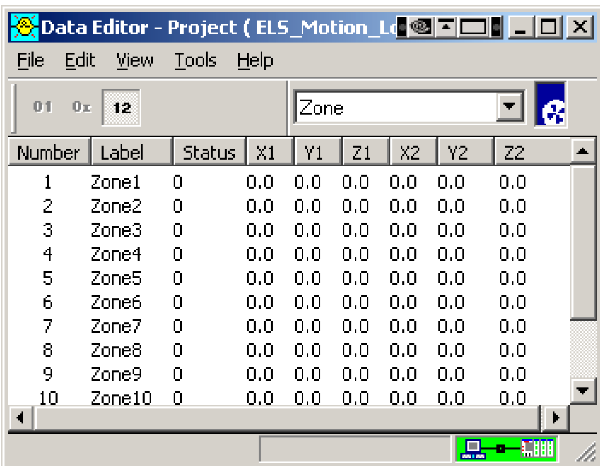


AddZone.tif

Fig. 4-114: Add a Zone

For example:

If the number 15 is entered, VisualMotion will allocate memory for 15 zones. Even if only zone number 15 is displayed in the VM Data Table. Selecting ***Data*** ⇒ ***Zones*** displays a list of 15 zones. Each zone is allocated 28 bytes of memory. The total memory allocated for zones can be viewed by selecting the *Data Summary* tab.



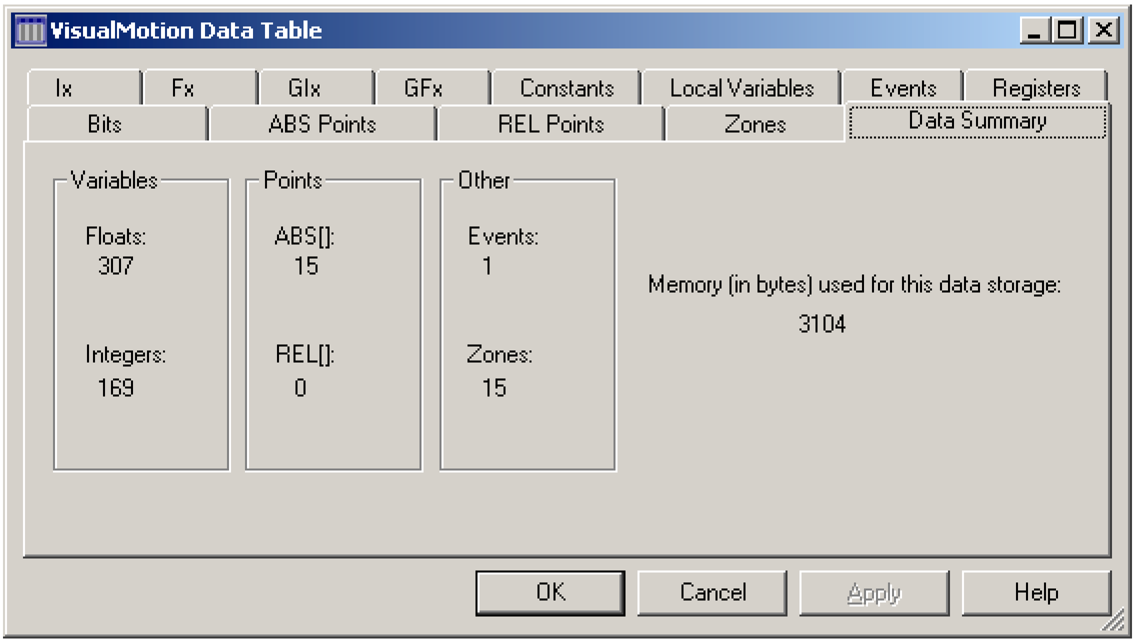
Number	Label	Status	X1	Y1	Z1	X2	Y2	Z2
1	Zone1	0	0.0	0.0	0.0	0.0	0.0	0.0
2	Zone2	0	0.0	0.0	0.0	0.0	0.0	0.0
3	Zone3	0	0.0	0.0	0.0	0.0	0.0	0.0
4	Zone4	0	0.0	0.0	0.0	0.0	0.0	0.0
5	Zone5	0	0.0	0.0	0.0	0.0	0.0	0.0
6	Zone6	0	0.0	0.0	0.0	0.0	0.0	0.0
7	Zone7	0	0.0	0.0	0.0	0.0	0.0	0.0
8	Zone8	0	0.0	0.0	0.0	0.0	0.0	0.0
9	Zone9	0	0.0	0.0	0.0	0.0	0.0	0.0
10	Zone10	0	0.0	0.0	0.0	0.0	0.0	0.0

DataZones.tif

Fig. 4-115: Selecting Data → Zones

Data Summary

The Data Summary tab displays the total memory allocation by VisualMotion for Integers, Floats, Global Integers and Floats, Points and Zones.



DataSummary.tif

Fig. 4-116: Data Summary

4.13 Labels

Note: The Labels menu selection is only available in service mode.

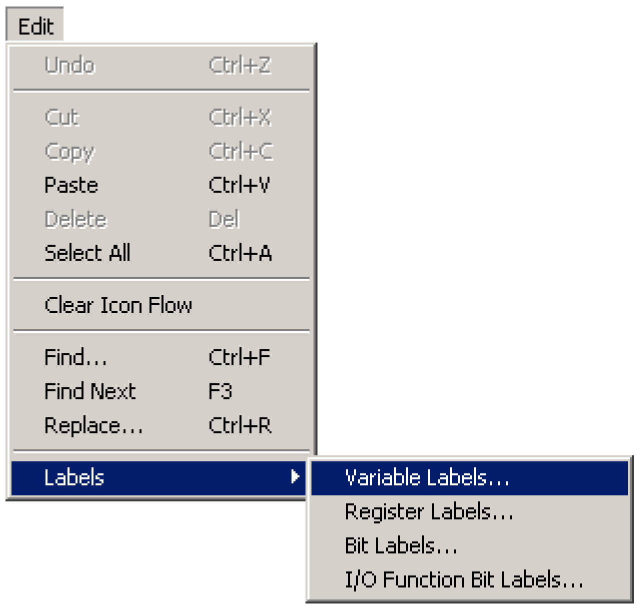
Labels are symbolic names assigned to system resources, such as axes, drives, etc.. Labels may also be used for absolute or relative point names, or in place of "literal" constant or variable values in expressions. For example, once assigned, the label "PI" can be used throughout a program, instead of repeatedly entering the literal value 3.14159.

Labels may use up to twenty case-sensitive ASCII characters. Blank spaces are not allowed within a symbol. Use a printable character as a separator if it is required for clarity. For example, "next_move," rather than "next move". The first character of a label must be an alpha character.

The control compiler, used for both icon and text language programming, allows the use of a literal integer value (i.e., a number such as "1" or "5"). Provided it is within the range of integers that are valid for the specified argument. Integers used to specify system devices, such as an axis or drive, must be within the range permitted by the complete VisualMotion system and installed software.

For example: a VisualMotion system with eight digital drives installed can specify an axis or drive using an integer from 1 to 8. The compiler must be able to resolve a symbol used as a table index argument to an integer index within the range, or size, of the table.

VisualMotion has a number of keywords, which it uses for command instructions. These keywords cannot be used as labels. If a keyword is used as a label, VisualMotion will issue the error "Label is a Keyword!" when the user tries to save the label. Refer to Using VisualMotion Keywords as Names on page 4-91 for a listing.



EditLabels.tif

Fig. 4-117: Edit Labels

Variable Labels

Selecting ***Edit*** ⇒ ***Edit Labels*** ⇒ ***Variable Labels*** opens the **User Defined Labels** window, used to provide symbolic ASCII names for Float, Integer, Global Float, Global Integer, Absolute or Relative point values.

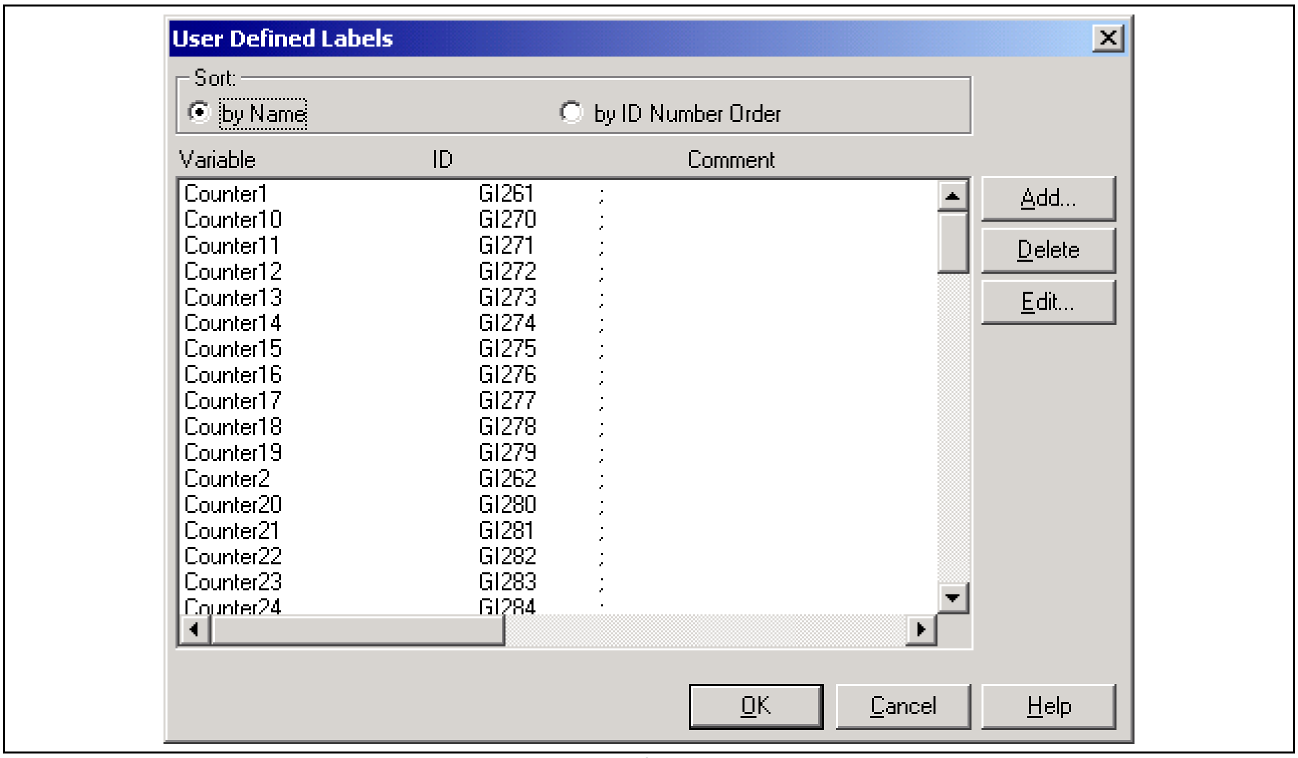


Fig. 4-118: "User Defined Labels" Window

The **User Defined Labels** window allows you to assign an ASCII name to a value or system component, as previously described. To sort the list alphabetically by label, select the radio box **by Name**. . To sort the list alphabetically by ID, select the radio box **by ID Number Order**. After labels are defined, instead of explicitly entering a value or redefining a system component, the label can be entered by accessing the **User Defined Labels** window and selecting the appropriate label.

To add a new label:

1. Click **Add…** to open the **Add Variable Label** window below.

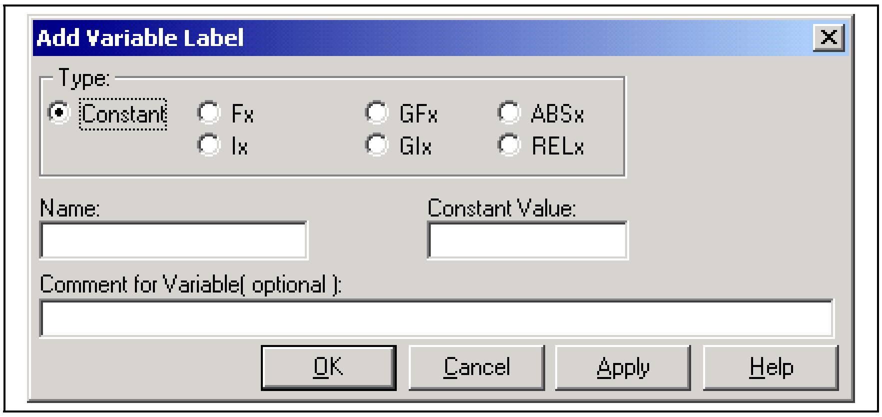


Fig. 4-119: "Add Variable Label" Window

2. Enter the desired **Type**, **Name**, **Constant Value** (if applicable) and **Comment** (up to 80 characters).
3. Click **Apply** or **OK**.

Rexroth Bosch Group

To delete an existing label:

Highlight a desired label and click **Delete**.

To edit an existing label:

1. Highlight a desired label and click **Edit…** .

   The **Edit User Label** window opens just below. The label's **Type**, **Name**, **Global Integer** and **Comment** are filled in as they were originally designated.

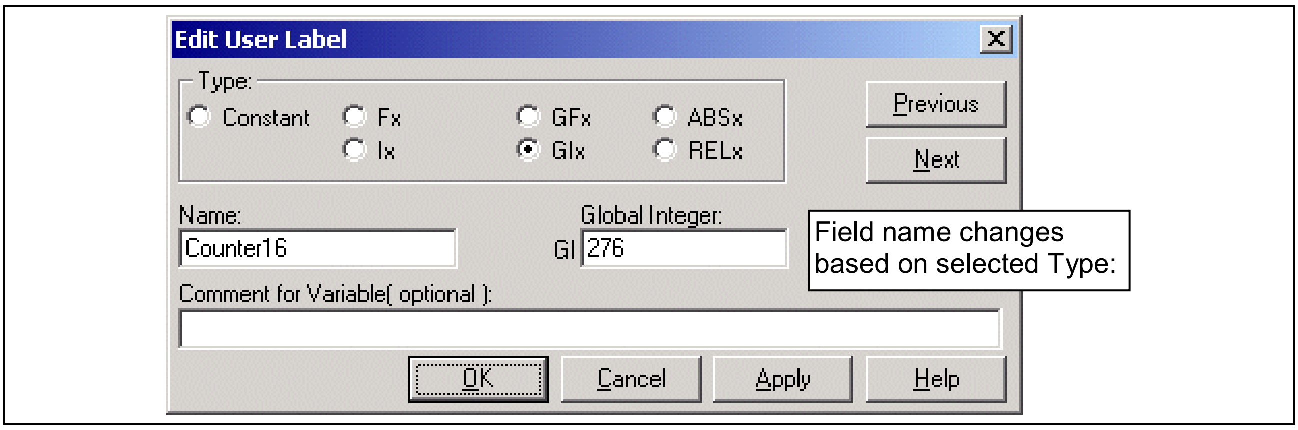


Fig. 4-120: "Edit User Label" Window

2. Enter the desired **Type**, **Name**, **Global Integer** (if applicable) and **Comment** (up to 80 characters).
3. Click **Apply** or **OK**.

Register Labels

Selecting ***Edit*** ⇒ ***Edit Labels*** ⇒ ***Register Labels*** opens the **Register Labels** window, used to provide symbolic ASCII names for the VisualMotion control status and I/O registers.

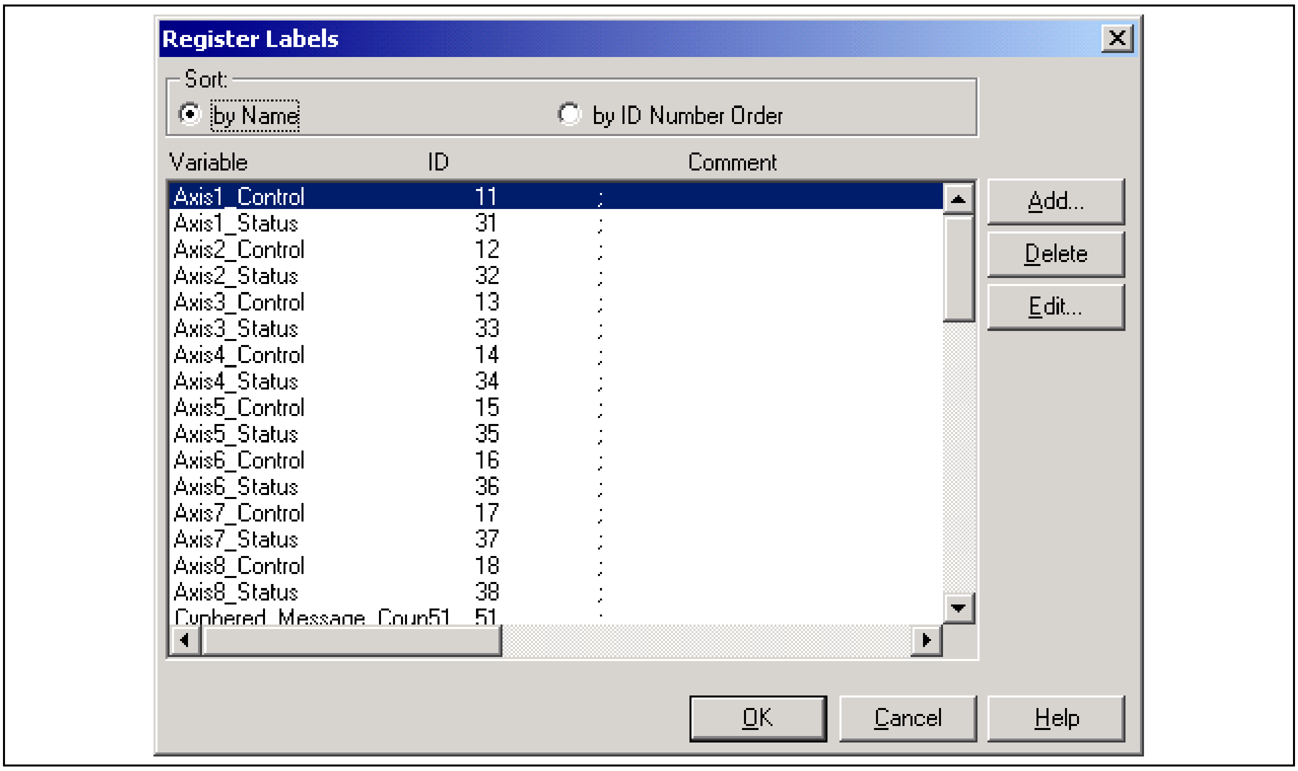


Fig. 4-121: "Register Labels" Window

To sort the list alphabetically by label, select the radio box **by Name**. To sort the list alphabetically by ID, select the radio box **by ID Number Order**. After register labels are assigned and the program is saved, the labels are embedded in the motion program (the .str file) and will not be lost if the program is later transferred to a different VisualMotion system. New programs are loaded with the default register names. Refer to chapter 16 in volume 2 of the *VisualMotion Functional Description*, Registers, for default register names.

To add or edit a register label:

1. Click **Add…** or select the desired register and click **Edit…** .

   An Add/Edit Register Labels window opens. The **Name** must be explicitly entered. (When editing an existing label, the current **Name**, **Register Number** and **Comment** are automatically filled in.)

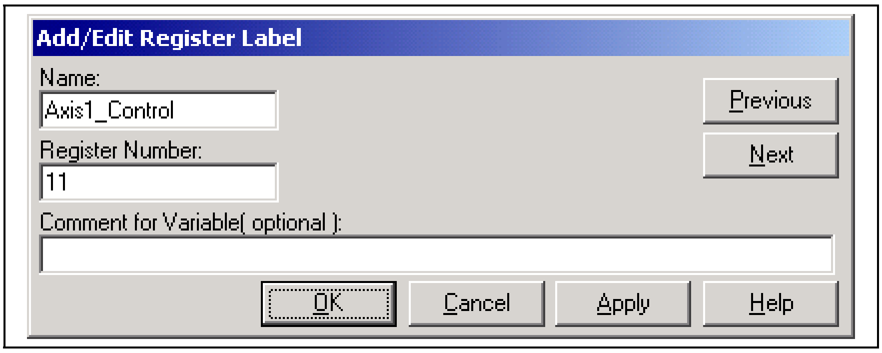


Fig. 4-122: "Add/Edit Register Label" Window

2. Enter the desired **Name**, **Register Number** and **Comment** (up to 80 characters).
3. Click **Apply** or **OK**.
   The **Previous** and **Next** buttons scroll through the defined labels.

Bit Labels

Selecting ***Edit*** ⇒ ***Edit Labels*** ⇒ ***Bit Labels*** opens the **Bit Labels** window, used to provide symbolic ASCII names for individual bits within VisualMotion control and I/O registers.

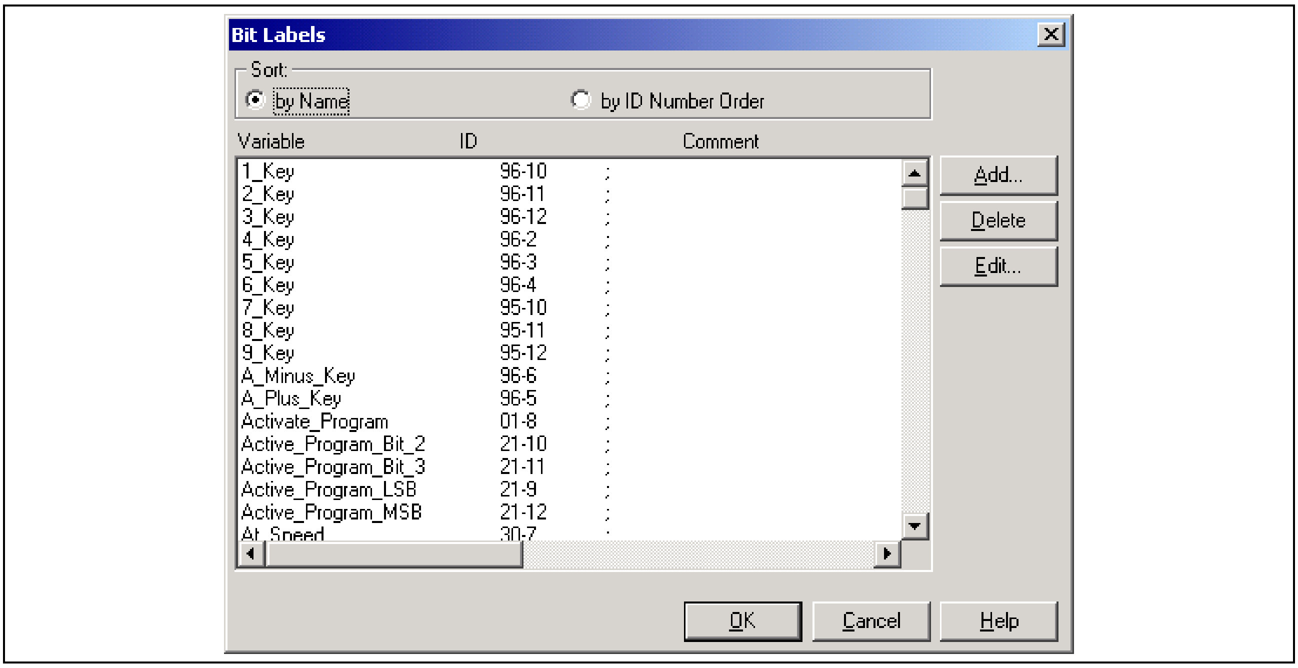


Fig. 4-123: "Bit Labels" Window

To sort the list alphabetically by label, select the radio box **by Name**. To sort the list alphabetically by ID, select the radio box **by ID Number Order**.

After bit labels are assigned and the program is saved, the labels are embedded in the motion program (the .str file) and will not be lost if the program is later transferred to a different VisualMotion system. New programs are loaded with the default bit names.

To add or edit a bit label:

1. Click **Add…** or select the desired bit and click **Edit…** .

   An **Add/Edit Bit Label** window opens. The **Name** must be explicitly entered. (When editing an existing label, the current **Name**, **Reg-Bit Number** and **Comment** are automatically filled in.)

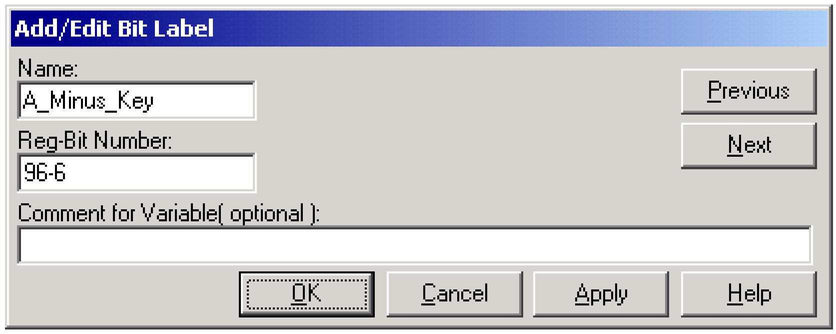


Fig. 4-124: "Add/Edit Bit Label" Window

2. Enter the desired **Name**, **Reg-Bit Number** and **Comment** (up to 80 characters).
3. Click **Apply** or **OK**.
   The **Previous** and **Next** buttons scroll through the defined labels.

I/O Bit Function Labels

Selecting ***Edit*** ⇒ ***Edit Labels*** ⇒ ***I/O Bit Function Labels*** opens the **Global Integer Bit Labels** window, used to provide default ASCII names for I/O Mapper functions.

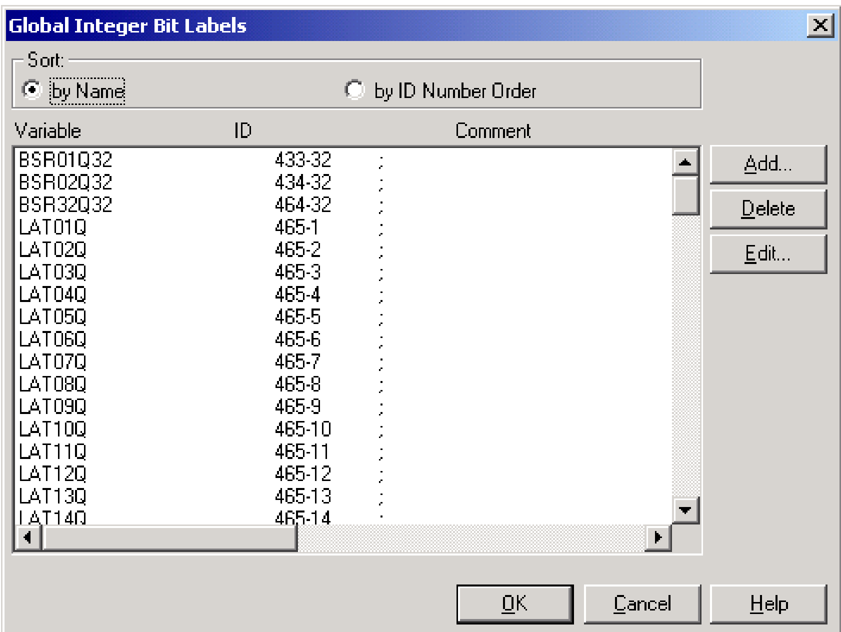


Fig. 4-125: “Global Integer Bit Labels” Window

To sort the list alphabetically by label, select the radio box **by Name**. To sort the list alphabetically by ID, select the radio box **by ID Number Order**.

After bit labels are assigned and the program is saved, the labels are embedded in the motion program (the .str file) and will not be lost if the program is later transferred to a different VisualMotion system. New programs are loaded with the default bit names. Refer to chapter 16 in volume 2 of the *VisualMotion Functional Description*, Registers, for default bit names.

To add or edit a bit label:

1. Click **Add…** or select the desired bit and click **Edit…** .

   An **Add/Edit Bit Label** window opens. The **Name** must be explicitly entered. (When editing an existing label, the current **Name**, **Reg-Bit Number** and **Comment** are automatically filled in.)

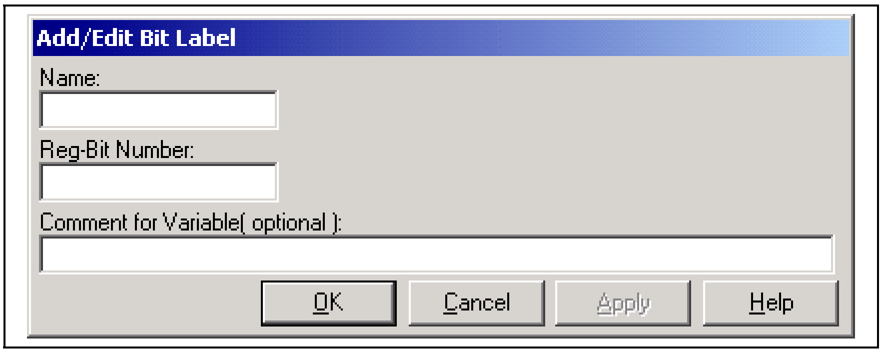


Fig. 4-126: "Add/Edit Bit Label" Window

2. Enter the desired **Name**, **Reg-Bit Number** and **Comment** (up to 80 characters).
3. Click **Apply** or **OK**.

5 Programming Capabilities

5.1 Overview

This chapter presents supplementary tools available for programming with VisualMotion 11. *Events* provide a method for executing special programs based on a specific condition in a program task. *Position Monitoring Group* (GMP), *Proportional Integrative Derivative* (PID), and *Programmable Limit Switch* (PLS) function in the background of a motion program to provide monitoring and signal output.

5.2 Events

Events are special subroutines that run only after they have been armed and a specific predefined condition has been met. This provides an advanced method for programming greater accuracy to an icon program for controlling the execution of a task. An event consists of a trigger and an event function.

Trigger

The trigger is the information used when arming an event and contains the following components:

Trigger Component	Description
Event Number	Numerical value identifying the event trigger in the project
Event Name	User defined label for the event trigger in the project
Event Argument	The value or condition that triggers (executes) the special subroutine (event function) to run
Event Type	This could be one of the following: coordinated motion, single-axis motion, Repeating Timer, Rotary, Probe, or I/O. Refer to Event Types on page 5-7 for details.
Event Function Name	The name for the special subroutine called by the trigger

Table 5-1: Trigger Elements

Event Function

An event function is a special subroutine that runs when the trigger condition is met.

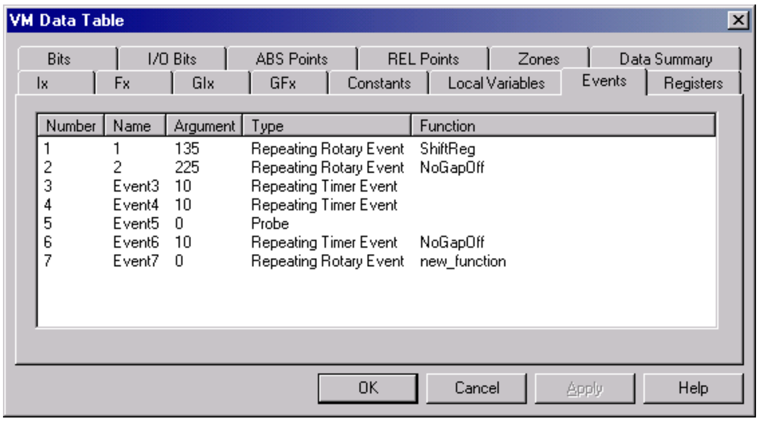
Event Triggers

VisualMotion Toolkit offers multiple methods for adding event triggers to an icon program. The *Events* tab in the *VM Data Table* window displays all the data associated with each event trigger in the project and allows data to be edited, deleted, and created in offline or online mode.

Note: Event trigger information can be created prior to creating the icon program.

The *VM Data Table* can be opened by clicking this toolbar button or by selecting ***Edit ⇒ VM Data Table…***

Rexroth Bosch Group



VM_Data_Table_Events.tif

Fig. 5-1: *VM Data Table* Events Tab

Event triggers can be enabled through event arming icons such as, *Event*, *Move*, *Circle*, and *Path* icons. Event types such as, Single Axis or Coordinated Motion are predefined and selected based on the icon. Refer to Event Types on page 5-7 for details.

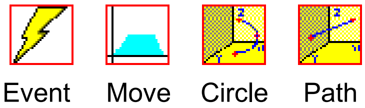


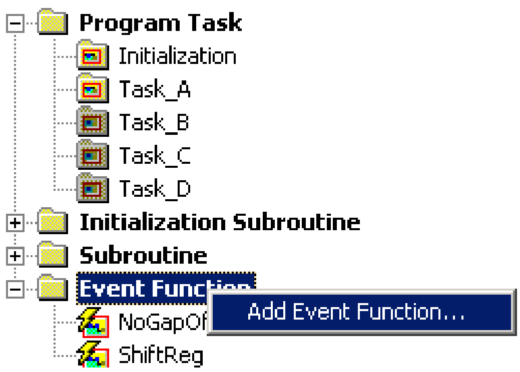
Fig. 5-2: Event Arming Icons

Note: The *Calc* icon can also be used to configure an existing event trigger in the icon program. Memory must be allocated using the *VM Data Table* before the Calc icon can be used to configure an event trigger. Refer to VM Data Table on page 5-6 for details.

For complete details on the icons described, refer to chapter 14 (Icons) in volume 2 of the *VisualMotion 11 Functional Description Manual*.

Event Functions

An event function is a subroutine that runs when the configured trigger condition is met. Event functions can be added by selecting ***Edit ⇒ VM Data Table…*** or ***Insert ⇒ Event Function…*** from the VisualMotion main menu. An event function can also be added by right-clicking the ***Event Functions*** folder in the *Project Navigator* tab.



Add_Event_Function.tif

Fig. 5-3: Add Event Function in Project Navigator

Note: No arguments can be passed to an event function.

This opens the *Event Function Control Block* window where the event name is entered. After the event function name is entered, a new icon programming workspace opens with a *Start* and *Finish* icon in place.

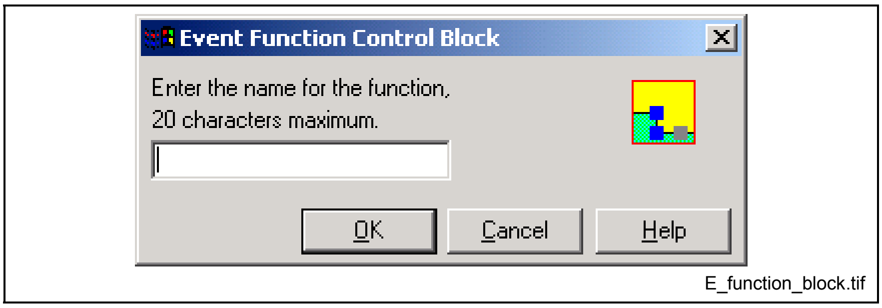


Fig. 5-4: Event Function Control Block window

Create the event function that will run when the condition in the icon program is true.

Note: Icon instructions that require a long time to process, such as the Home, Wait, Branch, and Parameter Transfer icons, should be avoided within event functions. Task response time and execution can be adversely affected if the event function requires too much time to process.

Relationship of Event Components

More than one event arming icon can be assigned to the same event trigger as long as the event trigger types match, as illustrated in Fig. 5-5. For example, a rotary event arming icon must be assigned to a rotary event trigger. It is also possible to have more than one event trigger call the same event function. The event function is not dependent on the type of event trigger; any type of event trigger can call an event function.

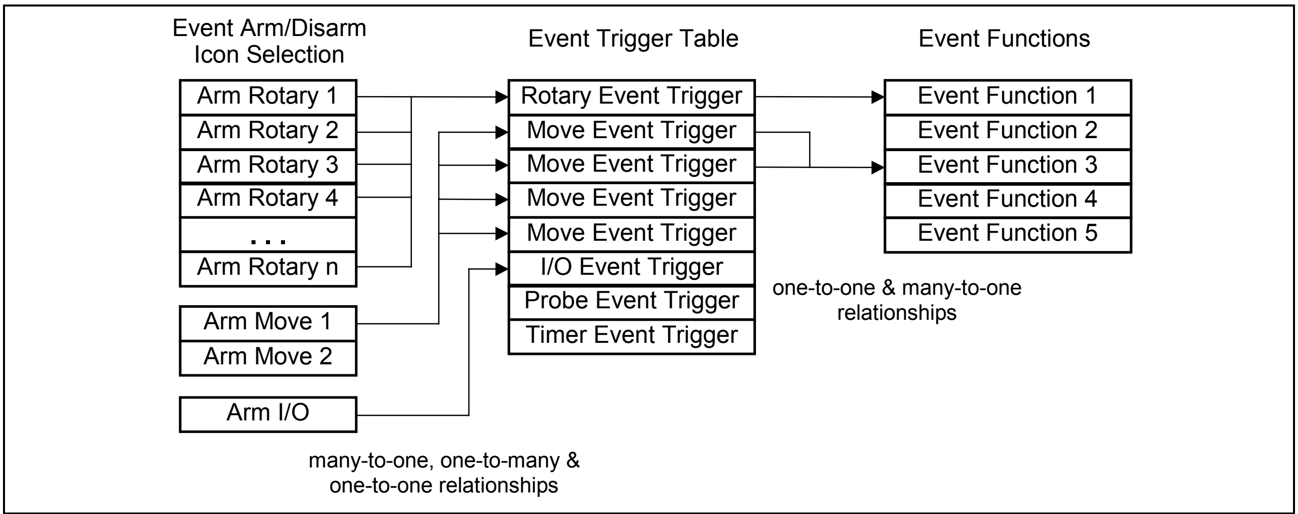


Fig. 5-5: Event Arming, Trigger, and Function Relationship

Rexroth
Bosch Group

Event Processing

If more than one event is used in a program, the events are prioritized according to the following designation:

Highest	Priority 1 - PPC-R X1 Input Event
	Priority 2 - Events from Task A
	Priority 3 - Events from Task B
	Priority 4 - Events from Task C
	Priority 5 - Events from Task D
	Priority 6 - Timer Task (repeating events)
Lowest	Priority 7 - User Tasks A, B, C, and D

The following conditions should be considered when creating a VisualMotion project with events:

- Events interrupt user tasks
- Events **(except the repeating timer event)** are assigned to a user task
- Each user task has a separate event queue (stack) , which can store up to 25 events
- The repeating timer event stack is separate from user task stacks and can store up to 16 events

Note: An event queue (stack) is a storage area where events are placed as they are triggered. The events are accumulated in the stack and are activated in the order in which they are received. Once an event has been executed, it is cleared from the stack and the remaining events move up in the stack. VisualMotion will issue a *Stack Overflow* error when the maximum number of event in a queue is exceeded.

- A higher priority event takes precedence over a lower priority event. If lower priority events are queued and a higher priority event is armed, the higher priority event is placed above a lower priority event as long as the lower priority event is not running. Refer to Table 5-4 for details.
- The stack handles events with the same priority in a First In First Out (FIFO) order.
- The PPC-R X1 input events are considered “fast input events” and will always be placed at the top of the FIFO stack. Refer to PPC-R X1 Input Event on page 5-19 for details.

Event Programming

Event triggers and functions in a project can be created or modified in either offline or online mode. VisualMotion 11 provides the following methods for programming events in a project.

Note: Service mode can also be used to edit event triggers and functions. However, when using service mode, only the data on the control will be modified. It is the responsibility of the programmer to ensure that all project related data is properly imported in to the project files. Refer to *Importing Project Data* in chapter 10 of the *VisualMotion 11 Functional Description* for details.

Method	Description
Event Arming Icons	The Event, Move, Circle, and Path icons can be configured to arm event triggers
VM Data Table	Used to name an event trigger, select an event type, configure the argument (trigger condition) and select the event function that will run.
Data Editor	In online mode, argument values can be modified. In offline mode, the event type and argument can be modified.
Event Function folder in Project Navigator	Used to add, delete, and edit event functions

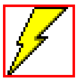
Table 5-2: Event Programming Methods

The steps for configuring or modifying events are the same in offline, online or service mode. The only difference is the target where the data will be saved. When programming in offline mode all data is saved to the project files on your hard drive. When programming in online mode, only event trigger arguments can be modified. All other event trigger data, although it can be created or modified in online mode, must be saved and synchronized with the project in order for the changes to take effect. Refer to *Synchronize Project Components* in chapter 10 of the *VisualMotion Functional Description* for details.

Event Arming Icons

The event arming icons are the *Event*, *Move*, *Path*, and *Circle* icons. These icons can be used for arming event triggers in any of the four main tasks (A-D), in subroutines, and in event functions. Event arming icons are not available in the initialization task and in an initialization subroutine.

Event Icon



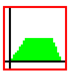
The Event icon is used to arm the following event types:

- Repeating Timer
- Rotary
- I/O
- Probe

Refer to Table 5-4: Overview of Event for details.

The Event icon can also be used for disarming auto rearming event triggers such as, repeating timer, rotary repeating axis, I/O register, or task input transition.

Move Icon

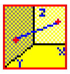
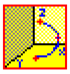


The *Move* icon is used primarily for single axis, non-coordinated absolute or relative distance moves. This icon also allows the user to configure up to 4 event triggers for the defined axis.

Rexroth Bosch Group

Event functions associated with the event triggers are executed based on the distance set in the *Move* icon and not from the actual feedback position. Refer to Single Axis Motion Events on page 5-9 for details.

Path and Circle Icons

The *Path* and *Circle* icons are used for absolute or relative coordinated motion. Event triggers for the *Path* and *Circle* icons are not associated directly with the icon but with the ABS or REL point assigned to the icon. Absolute or Relative points can have up to 4 event triggers configured. Refer to Coordinated Motion Events on page 5-7 for details.

VM Data Table

The configuring and allocation of memory for event triggers is done in the *VM Data Table* under the *Events* tab.

Allocating Event Triggers in a Project

Memory allocation in the *VM Data Table* is calculated based on the largest event trigger number configured.

For Example:

If only one event trigger is added and identified as number 5, then the control allocates memory for up to 5 event triggers. Afterwards, the *Calc* icon can be used, as part of the icon program, to configure event triggers 1, 2, 3, and 4. An error will be issued if the *Calc* icon is used to configure an event trigger numbered 6.

Note: Changes to event trigger arguments made in the *VM Data Table* do not effect a running program. The project must be saved and synchronized in order for changes to take effect.

Data Editor

The Data Editor can be used to modify existing event trigger data such as, argument value and event type. As an online tool, an event argument for a specific event trigger can be modified for a running program. Also, the status of an event can be monitored. The editor provides the following status indications:

Event Trigger Status	Description
Inactive	not triggered
Pending	triggered but not queued
Queued	triggered and awaiting execution of event function
Executing	event function is running
Done	complete
Repeat	retriggered and armed

Table 5-3: Data Editor Event Trigger Status

The *Data Editor* under menu selection ***Data ⇒ Events*** is used for modifying event arguments in a running program.

Event Types

The following table provides an overview of the supported event type in VisualMotion 11:

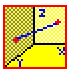
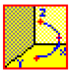
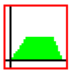
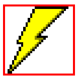
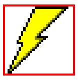
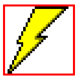
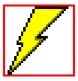
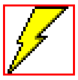
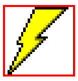
Event Type	Examples	Arming Mechanism	Auto Rearm	Maximum Number	Priority*
Coordinated Motion	• Start a move from another axis • Control a glue gun	 	No	4 events per move segment	2 – 4 Based on the assigned task
Single Axis Motion	• Start a move from another axis • Control a glue gun		No	4 events per single axis move	2 – 4 Based on the assigned task
Repeating Timer	• Switch a pump on every hour • Calculate statistics every minute		Yes	16 in a project	6 Lowest priority Runs independent of any task
Rotary Repeating Axis	• Control a valve • Change the H-Factor • Start a move from another axis		Yes	4 for each axis, group, or master	2 – 4 Based on the assigned Task
Probe	• Latch motor position of drive for registration function • Detect product position on a belt		No	2 per drive	2 – 4 Based on the assigned Task
I/O Register	• Move SERVO to a predefined position after opening a door		Yes	16 in a project	1 for Task A
Task Input Transition	• Start a calculation • Move SERVO to a predefined position after opening a door		Yes	1 per task	2 – 4 Based on the assigned Task
PPC-R X1 Input	• Time critical measurements • Latch virtual master position		No	3 absolute	1 for Task A Highest priority

Table 5-4: Overview of Event Types

Coordinated Motion Events

Multi-axis coordinated motion requires a trigger distance specified as a percentage of the total length of the segment or time (in ms) from the start or end of the segment. Event triggers for coordinated motion are associated with the points table that defines the path. Up to 4 event triggers can be configured for each point in a coordinated motion path. The point is set in either the Path or Circle icon.

Note: When using a jerk filter (non-zero jerk value) within a path segment, the triggering of an event function within the path segment may not occur as expected.

Rexroth
Bosch Group

Use the following steps to add a coordinated motion event to a project:

1. Place a Path or Circle icon in the icon workspace.

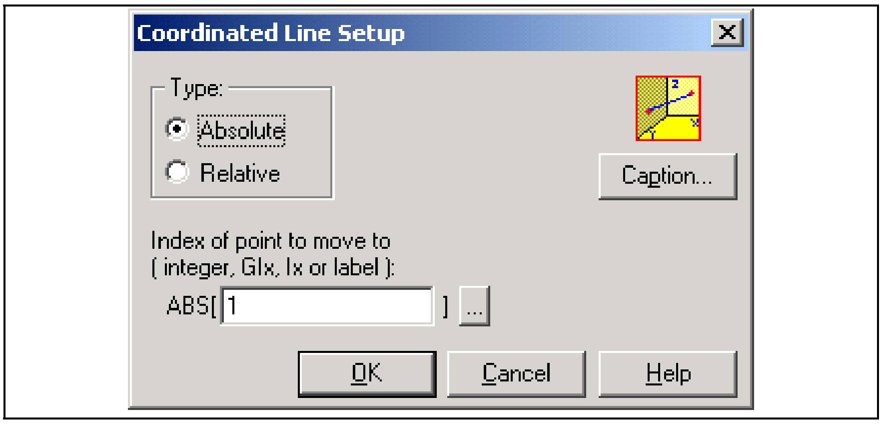


Fig. 5-6: Path Icon Setup for Percentage of Coordinated Path Event

2. From the setup window of a Path or Circles icon, select the browse button ... to open the *VM Data Table* window.
3. Select the *ABS* or *REL Points* tab.
4. If selecting an existing point, right-click on the desired point and select ***Edit ...*** If no points exists or a new point is to be created, right-click in the empty table and select ***Add …***
5. Check a box in the *Enable Event* section of the *Point* window to enable the browse button.
6. Click the browse button to open the *VM Data Table* displaying only events.
7. Select an event trigger that will be assigned to the point. If an event trigger does not exists, right-click the empty window, click ***Add …*** and create a new event trigger.
8. Name the event trigger, enter an *Argument* value and select the desired *Coordinated Motion Event Type*.

The following table lists the event types available for the Path or Circle icon along with their arming behavior, maximum number and description.

Event Type	**Auto Rearming**	**Maximum Number**	**Description**
Percent from Start	No	4 events per point	A position-based event that executes an event function at the set percentage of the overall travel from the start.
Percent before End	No	4 events per point	A position-based event that executes an event function at the set percentage of the overall travel before the end.
Time from Start	No	4 events per point	A time-based event that executes an event function at a set time from the start of the move.
Time before End	No	4 events per point	A time-based event that executes an event function at a set time before the end of the move.

Table 5-5: Coordinated Motion Event Types

9. Select an existing *Event Function* from the drop-down menu or create a new event function by clicking the ***New Function…*** button.

Note: Only the name of a newly created function will appear in the *Event* window. The actual event function can be program by expanding the ***Event Function*** folder in the Project Navigator window and double-clicking on the event function name.

The following figure illustrates an overview of the previous steps described above.

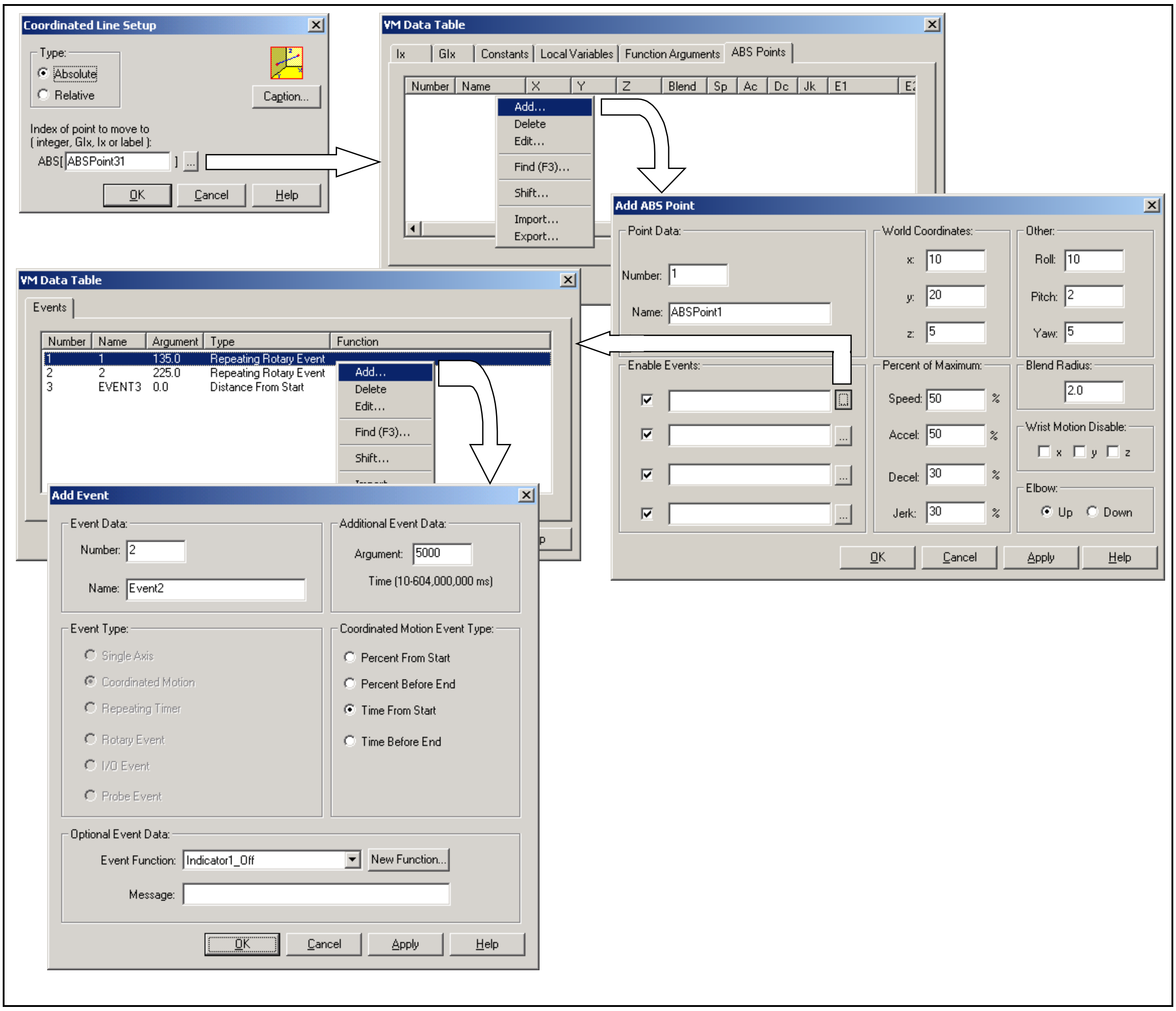


Fig. 5-7: Coordinated Motion Event Setup

Single Axis Motion Events

For single axis non-coordinated motion, an event function can be triggered at a set distance from the start of a move or before the end of a move. The event trigger is set within the Move icon. Up to 4 event triggers within one Move icon can be armed which can call up to 4 different event functions.

Note: Single axis, non-coordinated, motion icons use the internal positioning intelligence of Rexroth's digital drives. Because the rate profile for single axis motion is developed within the drive, the control uses the drive's feedback position. Consequently, a time jitter of 1-2 Sercos cycles may occur.

To add a single axis event to an icon program:

1. Place the Move icon in the icon workspace, select the axis and set the move distance.
2. Check a box in the *Event triggers* section to enable the browse button.

3. Click the browse button to open the *VM Data Table* and select the *Events* tab.
4. Select an event trigger that will be assigned to the move. If an event trigger does not exists, right-click the empty window, click ***Add …*** and create a new event trigger.
5. Name the event trigger, enter an *Argument* value and select the desired *Single Axis Event Type*.

The following table lists the event types available for the Move icon along with their arming behavior, maximum number and description.

Event Type	Auto Rearming	Maximum Number	Description
Distance From Start	No	4 events per move	A position-based event that executes an event function after the start of the move.
Distance Before End	No	4 events per move	A position-based event that executes an event function before the end of the move.

Table 5-6: Single Axis Event Types

6. Select an existing *Event Function* from the drop-down menu or create a new event function by clicking the ***New Function…*** button.

Note: Only the name of a newly created function will appear in the *Event* window. The actual event function can be program by expanding the ***Event Function*** folder in the Project Navigator window and double-clicking on the event function name.

The following figure illustrates an overview of the previous steps described above.

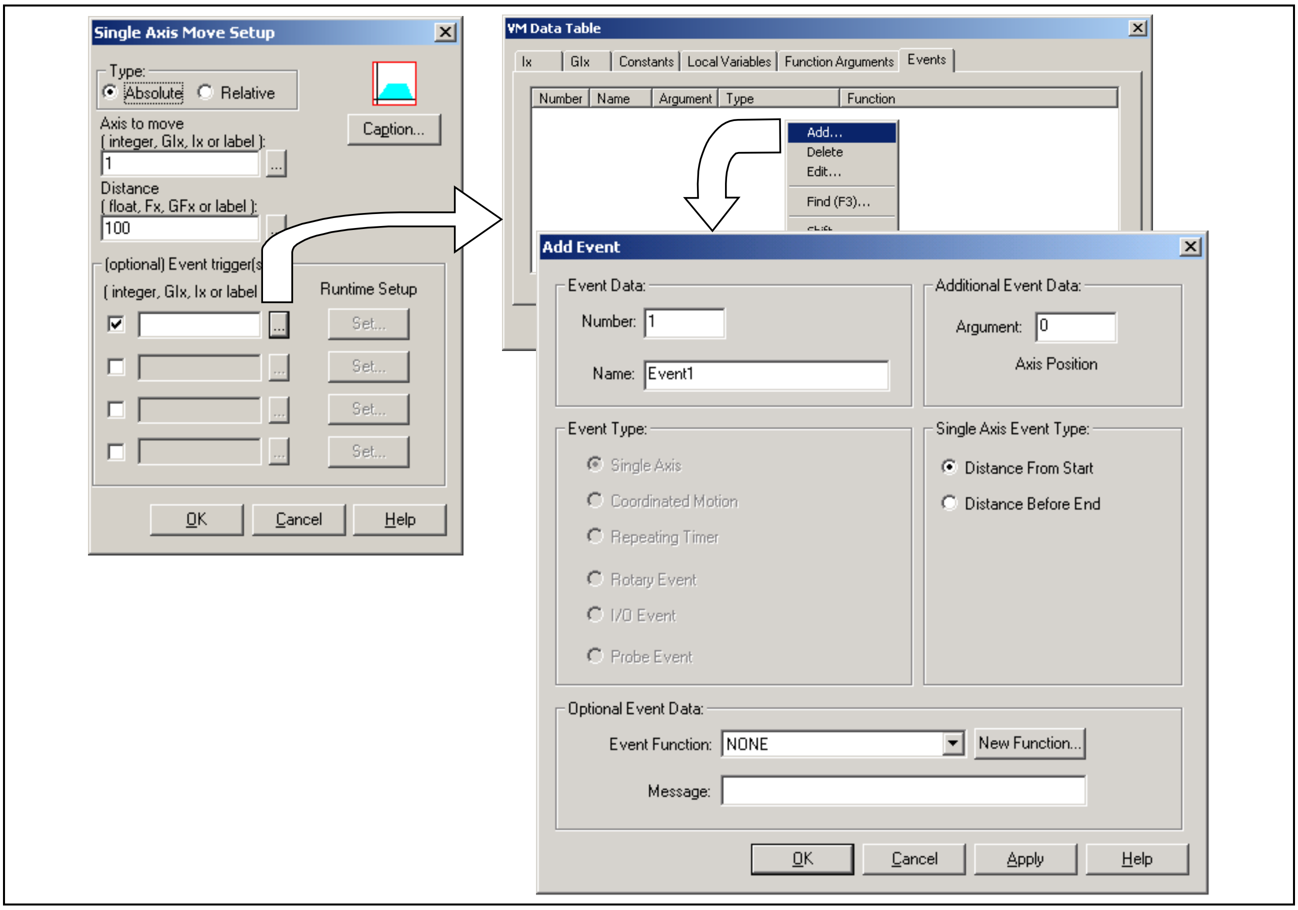


Fig. 5-8: Single Axis Event Setup

Special Cases for Single Axis Events

The following examples provide additional information regarding the functionality behavior of single axis event triggers.

Note: Single axis motion event triggers are executed when the Move icon is encountered in the icon flow, regardless of the move type (Absolute or Relative).

Special Case for Distance from End of Move

When an event trigger argument is larger than the remaining distance, at the moment the Move icon is encountered, ***the event function will still execute***. VisualMotion will calculate a new event trigger as follows:

Target Position – [0.99 * (Target Position – Current Position)]

Example:

Target Position	20mm
Single Axis Event Type	Distance from End of Move
Argument	15mm
Current Position	10mm (non-Modulo)
Axis Positioning Mode	Linear

In this example, 20mm – 0.99 * 10mm = 10.1mm (event trigger position). Therefore, instead of the programmed 15mm (Distance from End), VisualMotion will internally use an argument of 9.9mm, as illustrated in Fig. 5-9.

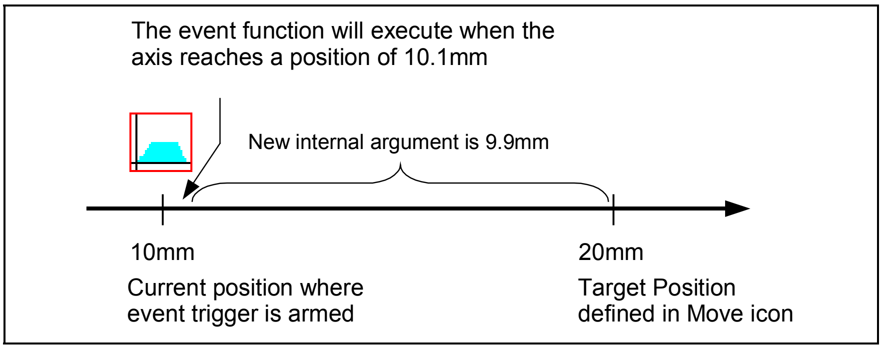


Fig. 5-9: Special Case for Distance from End of Move Event Type

Special Case for Distance from Start of Move

When an event trigger argument is larger than the remaining moving distance, at the moment the Move icon is encountered, ***the event function will still execute***. VisualMotion will calculate a new event trigger as follows:

Current Position + [0.99 * (Target Position - Current Position)]

Example:

Target Position	12mm
Single Axis Event Type	Distance from Start of Move
Argument	10mm
Current Position	5mm
Axis Positioning Mode	Linear

In this example, 5mm + [0.99 * (12mm-5mm)] = 11.9mm (event trigger position). Therefore, instead of the programmed 10mm (Distance from

Start), VisualMotion internally will use an argument of 6.9mm, as illustrated in Fig. 5-10.

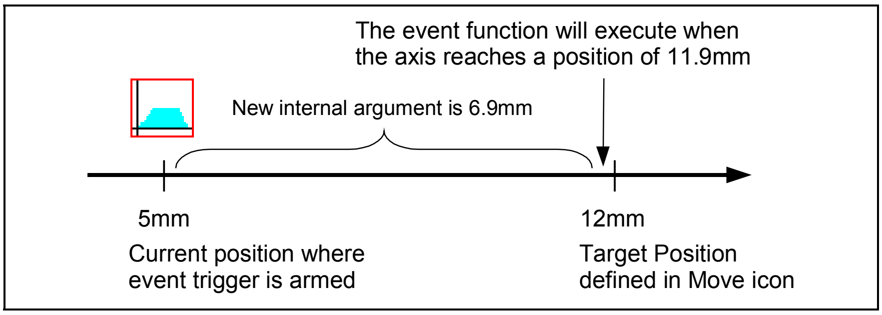


Fig. 5-10: Special Case for Distance from Start of Move Event Type

Special Case Single Axis Event Triggering Behavior

Special care must be taken when using multiple Move icons to trigger events for the same axis. The Move icon can arm up to 4 event triggers for the same axis. However, if the event trigger within one Move icon does not execute the associated event function before a second Move icon is armed, the first event function will be overwritten by the second event function.

To prevent any overlapping of event triggers when using multiple Move icons for the same axis, place the event triggers at different levels within the Move icons. Refer to Fig. 5-11 for details.

Note: Event triggers can be placed at the same level within Move icons as long as the timing between event triggers allows enough time for the event functions to execute.

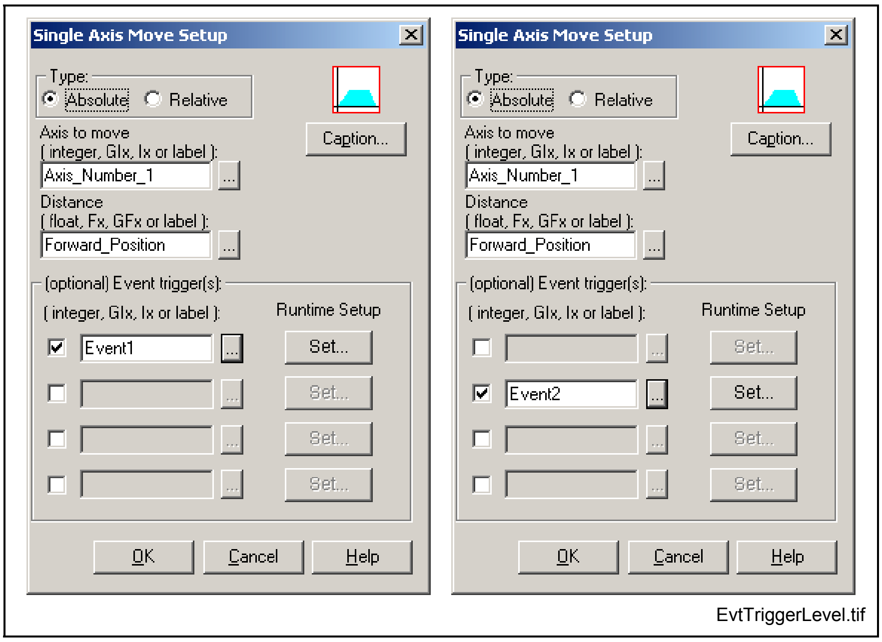


Fig. 5-11: Using Different Trigger Levels

If a Stop icon (Abort and Stop exhibit the same behavior) interrupts a Move icon procedure before its event function executes, the event trigger for the event function remains in the stack and will be executed when the axis reaches the event trigger position.

If a Move icon arms an event trigger and then an ELS Mode Change icon changes the operation mode to Velocity or ELS before the axis reaches the event trigger position, the event stays pending in the stack and will still be executed.

Special Case for Single Axis Rotary Positioning

The following example describes the single axis event trigger behavior when using rotary positioning with a single axis Move icon. The following should be taken in to consideration:

- S-0-0393, Command Value Mode, is used with the event trigger position calculation. The shortest path, positive and negative direction settings are part of the calculation.
- When using SGP03 and SGP20 firmware on EcoDrive 03, the axis can be configured for a relative move beyond one modulo value. In this case, all single axis event functions will trigger within the first modulo value of 360 degrees.

Example:

Target Position	720 degrees
Single Axis Event Type	Distance from End of Move
Argument	5 degrees
Current Position	200 degrees
Axis Positioning Mode	Rotary

When the event is triggered, the expected reaction would be for the event function to execute 5 degrees before the end of the move (after 715 degrees of travel). However, because single axis event triggers are executed within the first modulo value (360 degrees), the event function will actually execute after 355 degrees from the point where the Move icon is encountered. Refer to Fig. 5-12 for details.

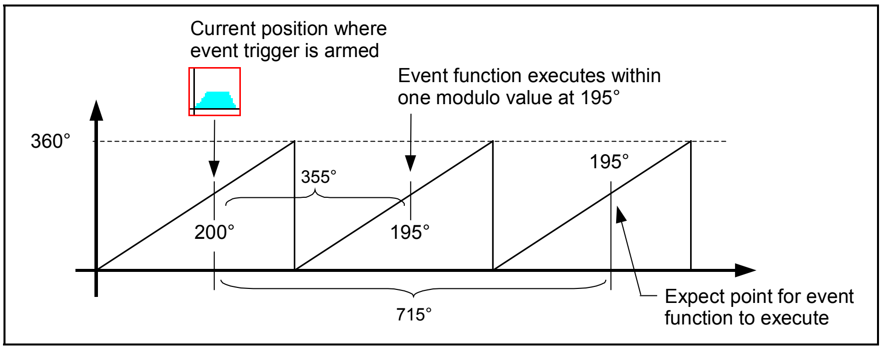


Fig. 5-12: Modulo Axis Event Trigger Behavior

Rexroth Bosch Group

Repeating Timer Event

The repeating timer event is triggered by an internal clock that runs independently of the event's associated task. The argument for the event is a value that represents the time interval between each triggering of the event.

Note: Repeating Timer events are auto rearming. This means that it will continuously run, until it is disarmed, even after its associated task has ended.

To add a repeating timer event to a project:

1. Place an Event icon in the icon workspace.
2. In the *Event Arm/Disarm* window, select **Arm Event** for the Event Action and **Repeating Timer** for the Event Type.

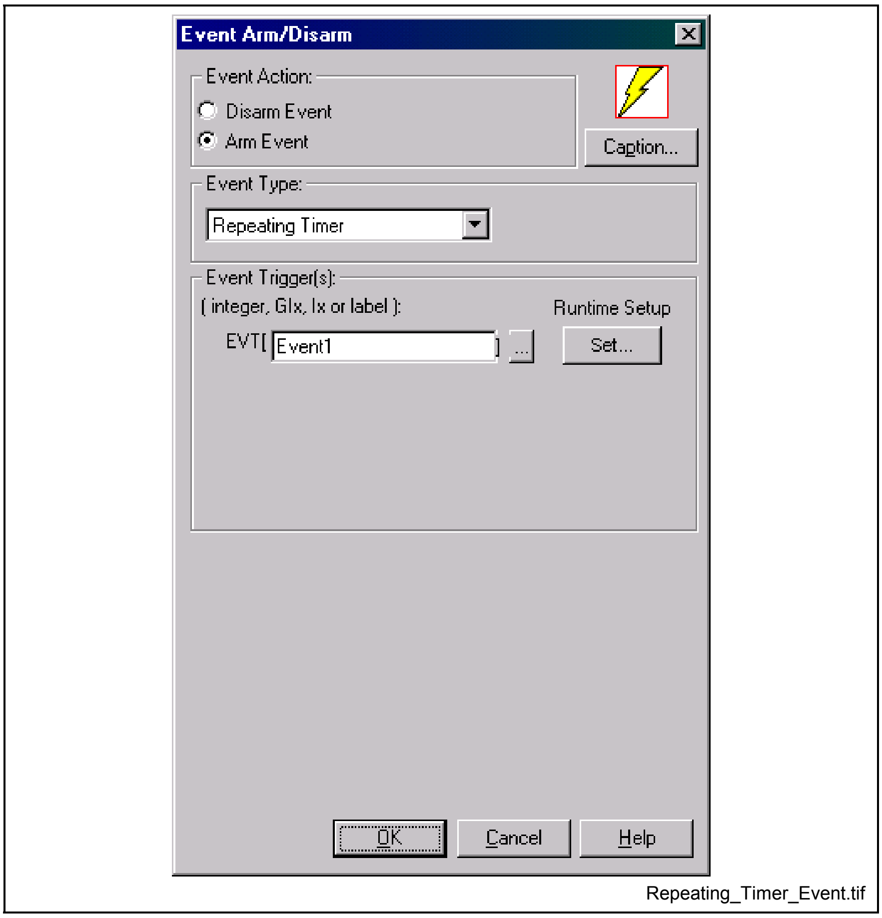


Fig. 5-13: Event Arm/Disarm Window

3. Type an event name in the **Event Trigger** field, select the event name from the *Events* tab in the *VM Data Table* window, or create a new event name by selecting ***Add*** to open the *Add Event* window.

Note: To disarm the repeating timer event, place a second Event icon in the program and select **Disarm Event** for the event action.

Rotary (Repeating Axis Position) Event

Rotary events are triggered each time the axis encounters an absolute position. The axis motion type can be single-axis, ELS, ratio, or velocity mode and configured for modulo or non-modulo positioning.

The event function will be triggered each time the axis reaches the position (from either direction) set in the argument. Because rotary motion uses the shortest path to reach the next specified position, verify that the axis will travel through the position specified in the argument.

Note: Rotary events are auto rearming. This means that it will continuously run, until it is disarmed, even after its associated task has ended.

To add a Rotary event to a project:

1. Place an Event icon in your icon program.
2. Select **Arm Event** for the Event Action.

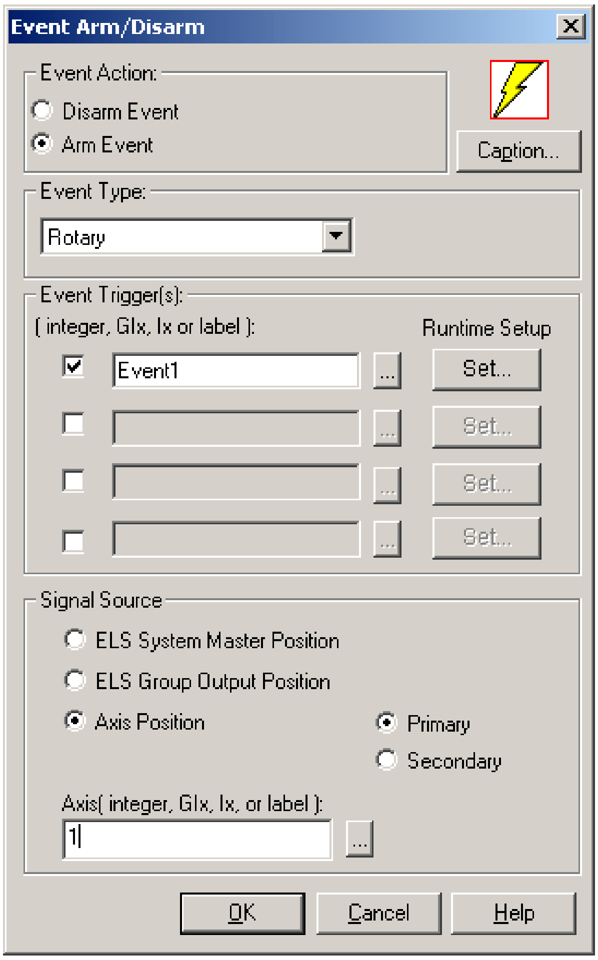


Fig. 5-14: Event Arm/Disarm Window

3. Select **Rotary** for the Event Type.
4. Enable the event trigger by placing a check in the box. Type an event name in the *Event Trigger* field or select the event name from the *Events* tab in the *VM Data Table* window.
5. Select a signal source, depending on whether the event function will be triggered by a single axis, ELS Group, or ELS Master, position.

Probe Event

VisualMotion supports Sercos probe functionality and real-time bits along with the event system. Typically, probes are used to detect registration marks on material. When the position is captured, the drive signals the control with a real-time bit in the Sercos cyclic data telegram (AT). When the control detects a change in the real-time bit, it can execute an optional event function.

IndraDrive, Diax 04, and EcoDrive 03 digital drives support up to two probe inputs that can be used for capturing feedback positions and/or time values. The following table lists the drive hardware and firmware types that support probe inputs.

Drive	Hardware	Firmware
IndraDrive	CSH control section	MPH-02
Diax 04	DSS 02.1	SSE-03 or ELS-05
EcoDrive 03	Sercos Interface	SGP 01, SGP 03, SGP 20, SMT-02, or MGP-01

Table 5-7: Probe Input Hardware Support

Note: The EcoDrive 03 cyclic data telegram has a limitation of 16 bytes. When the MUX channel is enabled (multiplexing) in the drive, only one probe input can be placed in the cyclic data telegram. Single axis probe inputs are not supported when the MUX channel is enabled. Refer to the *Sercos Drive Telegram Utility* in chapter 7 of the VisualMotion 11 Functional Description for details.

The inputs are wired in each drive according to the following figure:

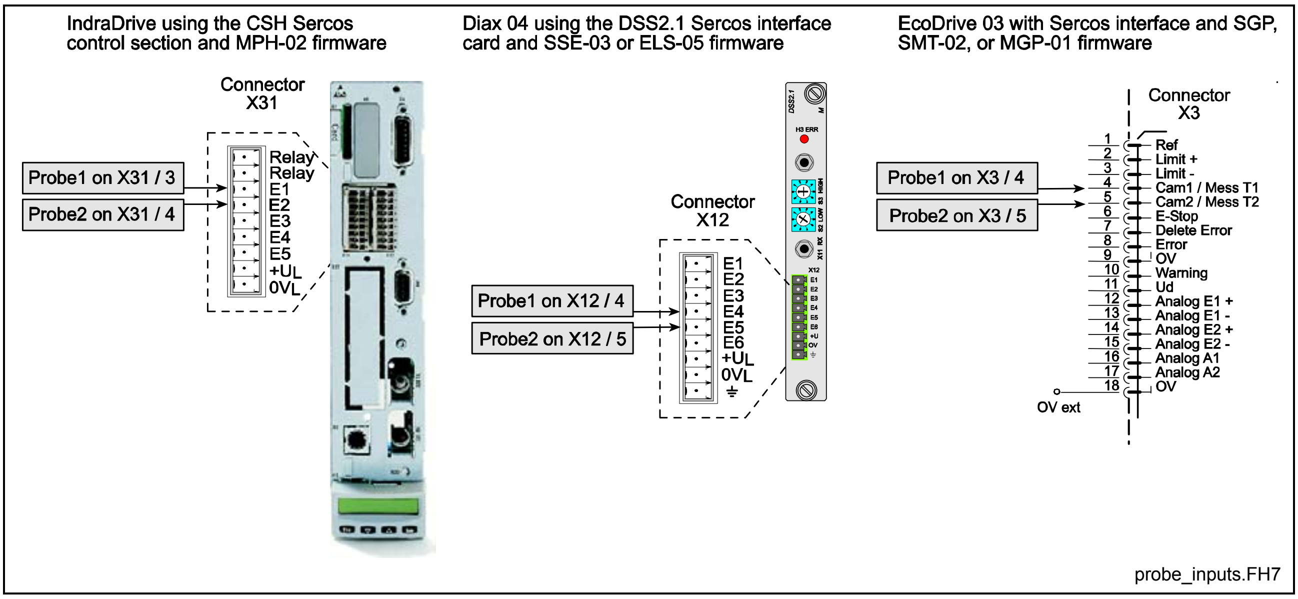


Fig. 5-15: Connecting the Probe Inputs

Note: For IndraDrive: If both probes are used, the 4 drive CAM support will be transmitted over the service channel.

The probe inputs are scanned every µs. Upon either a positive or negative transition of a probe input, the drive captures and places the position into the cyclic data telegram.

Note: It is also important that the value being read in the feedback event matches the parameter in the amplifier telegram (AT). If feedback is requested from a probe and its associated parameter is not in the AT, the service channel is used to transmit the data.

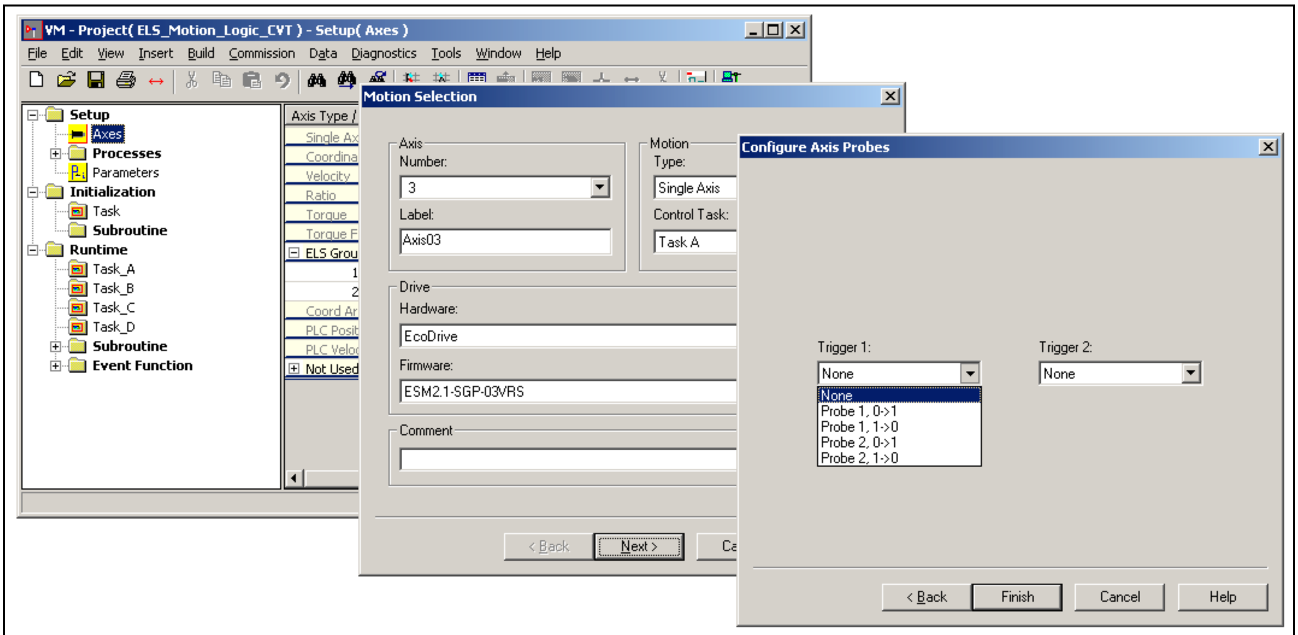
To add a Probe event to an icon program:

1. Add an axis under ***Setup*** ⇒ ***Axes*** of the desired motion type.

Note: Probes can be configured for the following motion types available in the drop-down menu of the *Task Axes Setup* window:

- ELS Axis
- Single Axis
- Coordinated
- Velocity
- Ratioed Axis
- Torque Mode
- Torque following Mode

2. Specify the Axis, Motion, and Drive in the *Motion Selection* window.
3. Click the **Next** button, setup the *Axis Properties* and continue to the *Configure Axis Probe* window.



Probe_Config.tif

Fig. 5-16: Configure Probe Window

4. Select the probes that will be associated with each event trigger in the *Configure Axis Probe* window.

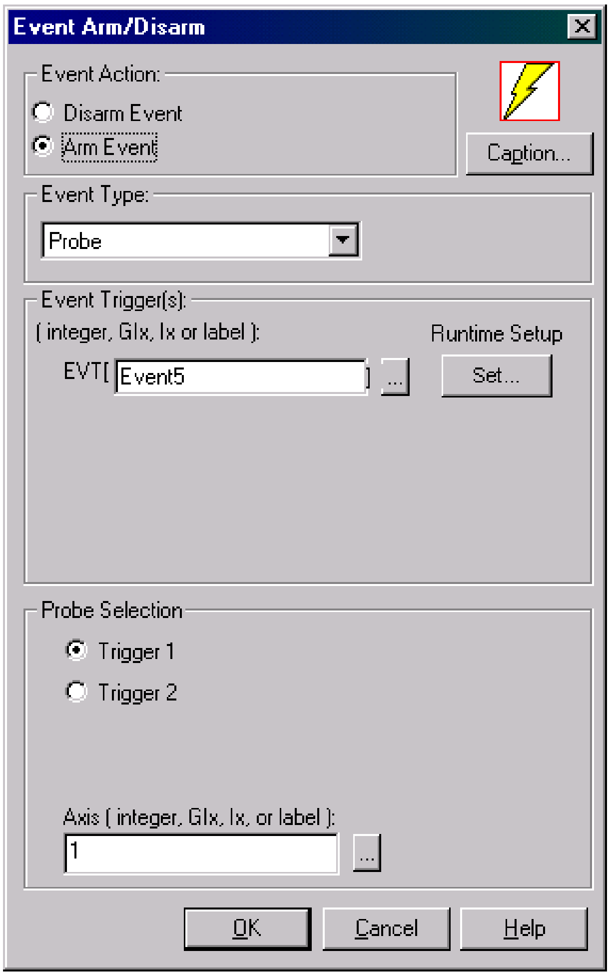
The following table lists the four probe trigger sources and their associated Sercos and control parameters:

Probe Trigger	**Sercos Parameter**	**Control Parameter**
Probe 1, 0->1	S-0-0130	A-0-171
Probe 1, 1->0	S-0-0131	A-0-172
Probe 2, 0->1	S-0-0132	A-0-173
Probe 2, 1->0	S-0-0133	A-0-174

Table 5-8: Available Probe Triggers

Note: Configuring the data that will be captured by the probes is done using DriveTop. Refer to drive documentation for additional information on configuring drives.

5. Select and place an Event icon in the icon program and select the **Arm Event** radio button.



Probe_Event.tif

Fig. 5-17: Event Arm/Disarm Window

6. Select **Probe** for Event Type.
7. Type the event trigger name or select it from the *VM Data Table*.
8. Select the probe trigger number. This number corresponds to the trigger number configured in the axis setup.
9. Select the same **Axis** that was setup under ***Setup*** ⇒ ***Axes***.

I/O Events

Events can be triggered through an I/O register or bit. There are four types of I/O Events:

- Task Input Transition
- I/O Register Event
- PPC-R Input Event (0->1)
- PPC-R Input Event (1->0)

Task Input Transition Event

Bit 9 of each Task Control Register (002-005) is reserved as an event interrupt input for the task. Every low-to-high transition of this input can trigger an event function. Following a high-to-low transition of the input, the event trigger is rearmed automatically. To disarm the event trigger, an additional event icon in the program flow is necessary to disarm the event.

The control scans the input every 2ms and queues an event trigger upon a low-to-high transition. The event function will take priority over the user tasks, allowing quick response to an external input.

IndraLogic can be used to invert the logic of the interrupt input, or to direct other external inputs to the Task Control Register's Event Interrupt bit.

I/O Register Event

An I/O Register event uses the bits of register 88 (USER_XI_REG) as inputs for up to 16 separate event triggers. Once an I/O Register event trigger is initially armed, every low-to-high transition of the bit will cause the event function to run. The event trigger is rearmed automatically after the high-to–low transition of the bit. An Event icon must be used in the icon program to disarm an I/O register event.

The bits in register 89 (USER_XO_REG) show the event status of their corresponding bit in register 88. This provides an external output to indicate the event is armed and ready for operation. Once an I/O register event is armed in the icon program, the corresponding bit in register 89 is set high. The bit in register 89 is set low again when the event is triggered and is set high again when the event is rearmed. The register 89 output bits are also set low if the event is disarmed using the Event icon.

PPC-R X1 Input Event

The PPC-R X1 High Speed Inputs can be used for high priority events. When pins 3, 4, or 5 on connector X1 of the PPC-R detect a positive or negative rise, the associated event function is triggered. If another event is currently running and a PPC-R X1 input event is triggered, it will run immediately after the current event has finished and before the next event in the stack.

Note: High speed input events are not automatically rearmed. An Event icon must be used to rearm the event.

The PPC-R X1 input event is triggered by a physical switch that is connected to pin 3, 4, or 5 of connector X1. The event is triggered by latching the switch to provide 24V to the input. The event could be triggered by either a latch or unlatch of the switch.

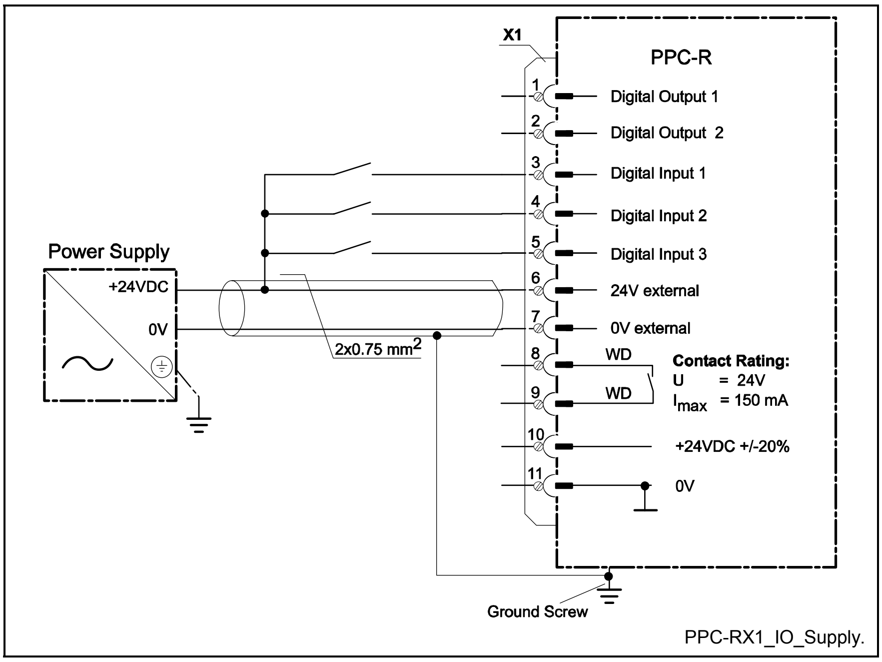


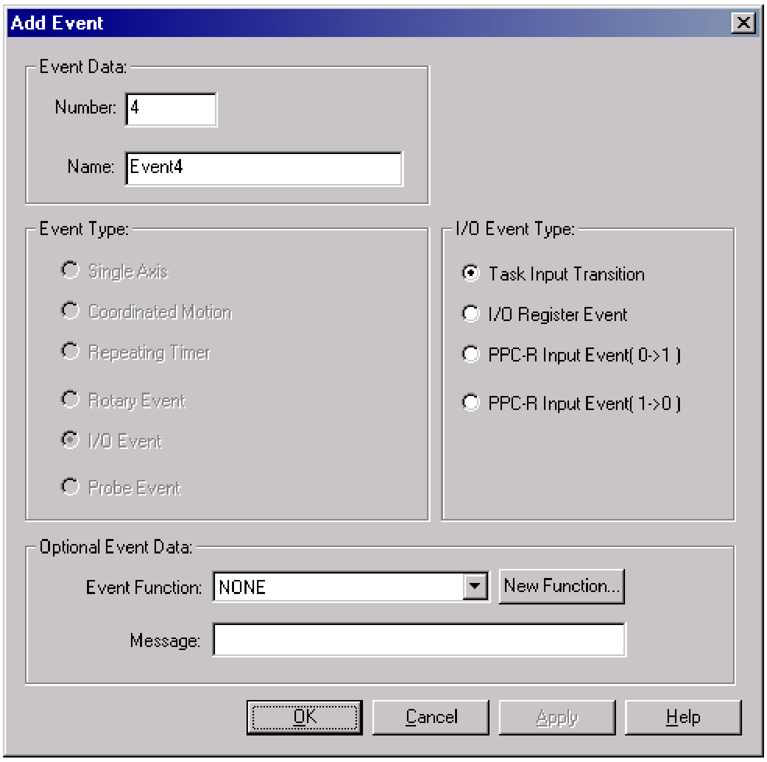
Fig. 5-18: Wiring Diagram for Digital Input/Output

Note: The digital inputs and outputs found on connector X1 of the PPC-R22.1 are not functional unless 24V are supplied to pins 6 and 7. Digital inputs I1, I2 and I3 are mapped the bits 1, 2, and 3 of register 44. Digital outputs Q1 and Q2 are mapped the bits 1 and 2 of register 45.

Adding an I/O Event

I/O Events are added using the Event icon in VisualMotion. To add an I/O Event to a project:

1. Place an Event icon in the task where the event will occur.
2. In the *Event Arm/Disarm* window, select Arm Event.
3. Select I/O for the Event Type.
4. Type the name of the event function or select or create one by selecting the *VM Data Table* button.
5. In the *VM Data* window, select the *Events* tab and select an event or click the ***Add*** button to create a new event.
6. In the *Add Event* window, enter the event number and name.
7. Select the I/O Event Type and enter the event function name or create a new one with the ***New Function…*** button.



Add_Event_IO.tif

Fig. 5-19: Add Event Window

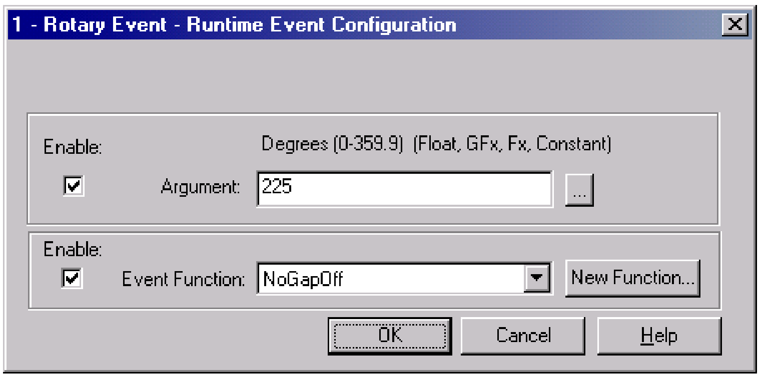
Runtime Setup

The Runtime setup allows you to dynamically change the event argument and/or event function for an existing event. The Runtime setup values for an event will overwrite the values for that same event in the *VM Data Table*.

Runtime changes can be set in the **Runtime Setup** field in the Event and Move icons.

Note: The functionality of the Runtime Setup was accomplished using the Calc icon in earlier versions of VisualMotion. The Calc icon can still be used to change event values in VisualMotion 11 as in previous versions of VisualMotion. Refer to the events topic in VisualMotion online help for information about configuring the Calc icon for and event.

The Runtime Setup field in the Event and Move icons has a ***Set…*** button which opens the *Runtime Event Configuration* window.



Runtime_Setup.tif

Fig. 5-20: Runtime Event Configuration Window

Rexroth
Bosch Group

Note: If a disarm function is used for an event, neither the argument or event function can be changed in runtime. In the *Runtime Event Configuration* window, the event function should be disabled by removing the check in the Enable check box.

A list of all the events in a project can be viewed by selecting ***View ⇒ Event Function…*** in the VisualMotion Toolkit main menu.

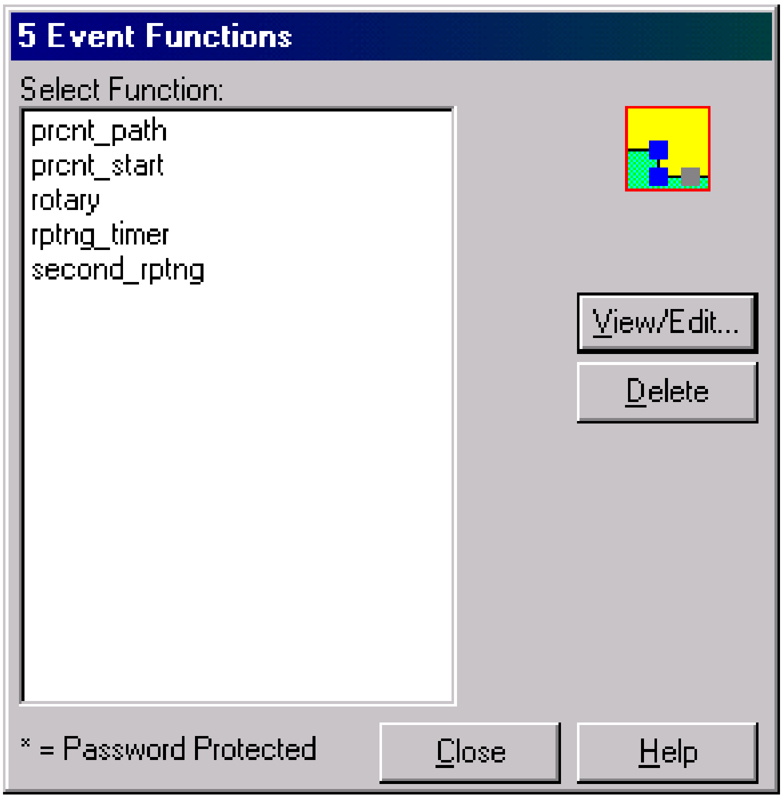


Fig. 5-21: Event Functions Window

In the *Runtime Event Configuration* window, the **Enable** checkbox must be selected to enable the **Argument** and **Event Function** fields.

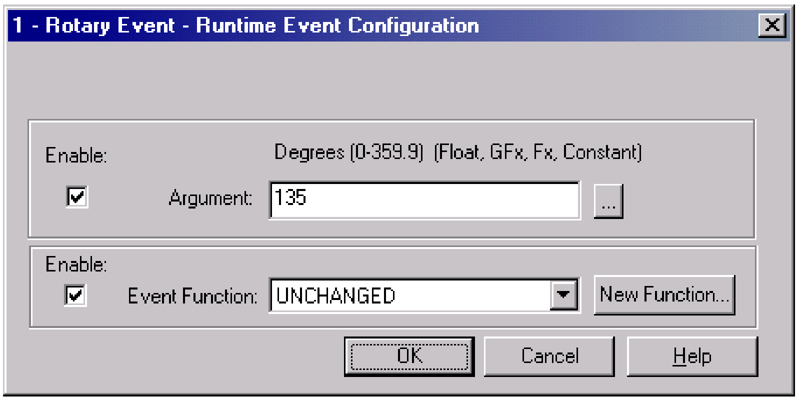


Fig. 5-22: Runtime Event Configuration Window

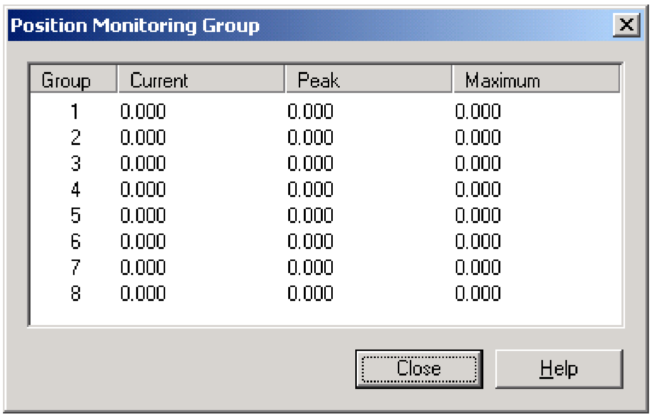
The checkbox can also be used to deactivate either setting so that the program will continue to use the original settings in the event trigger.

5.3 Position Monitoring Group

Certain applications require position deviation monitoring between multiple axes on a machine when following the same position command signal. Should one of the drives in a group deviate from a set-monitoring window, an error reaction can be set to warn the user of the deviation or stop all drives (issuing a fatal error). This function provides independent monitoring of multiple groups of axes including the X, Y and Z of a coordinate system, simultaneous single axes moves and axes in an ELS group or an ELS system master. The *Position Group Monitoring* window allows the configuration of up to 8 independent groups containing a maximum of 6 axes (1 primary and 5 slaves) per group. All configured axis groups are monitored every Sercos cycle.

Selecting ***Commission ⇒ Position Monitoring Group*** opens the window in Fig. 5-23.

Note: In offline mode, the Position Group Monitoring window is used to configure up to 8 groups. In online mode, the window is used to view the status of current, peak and maximum deviation values for all groups.



PMGconfig.tif

Fig. 5-23: Axis Position Monitor

Relevant Parameters for Axis Group Monitoring

The following control parameters are used when configuring Axis Position Monitoring.

Parameter	Description
C-0-3201, C-0-3211, C-0-3221, C-0-3231, C-0-3241, C-0-3251, C-0-3261, C-0-3271	PMG # Maximum Allowed Deviation Window (# identifies groups 1-8)
C-0-3202, C-0-3212, C-0-3222, C-0-3232, C-0-3242, C-0-3252, C-0-3262, C-0-3272	PMG # List of Axis (# identifies groups 1-8)
C-0-3203, C-0-3213, C-0-3223, C-0-3233, C-0-3243, C-0-3253, C-0-3263, C-0-3273	PMG # List of Position Offsets (# identifies groups 1-8)
C-0-3204, C-0-3214, C-0-3224, C-0-3234, C-0-3244, C-0-3254, C-0-3264, C-0-3274	PMG # Current Peak Group Deviation (# identifies groups 1-8)
C-0-3205, C-0-3215, C-0-3225, C-0-3235, C-0-3245, C-0-3255, C-0-3265, C-0-3275	PMG # Maximum Deviation (# identifies groups 1-8)
C-0-3206, C-0-3216, C-0-3226, C-0-3236, C-0-3246, C-0-3256, C-0-3266, C-0-3276	PMG # Configuration (# identifies groups 1-8)

Table 5-9: Position Monitoring Groups

System/Position Initialization

Before position monitoring can become active, the PMG parameters are processed. This process dynamically creates and initializes internal data structures with valid settings. The processing of the *PMG # List of Axis* and *PMG # LIST of Position Offsets* runs during every occurrence of system initialization (Phase- 2 → Phase-4). If any parameter is set incorrectly, errors are reported before the system is able to run in phase 4. In general, all position signals must have the same data format (in, mm, deg) and likewise, mechanical system (M/N ratios, feed constants/modulo, etc…). The system automatically detects if the position master is a modulo axis (e.g. reads S-0-0076, bit 7) and calculates the deviation window accordingly. The deviation window parameter will be scaled in the units of the master signal.

Note: If an axis requires drive homing, monitoring can only become activated after the homing process has been completed on all the axes in a grouping.

Configure Monitoring Group

A monitoring group is configured by double-clicking on a group number row in the *Position Monitoring Group* window. From the *Position Monitoring Group Settings* window, the user selects the Deviation Method, Primary Signal, Error Reaction on Maximum Deviation and Maximum Deviation Window for the group.

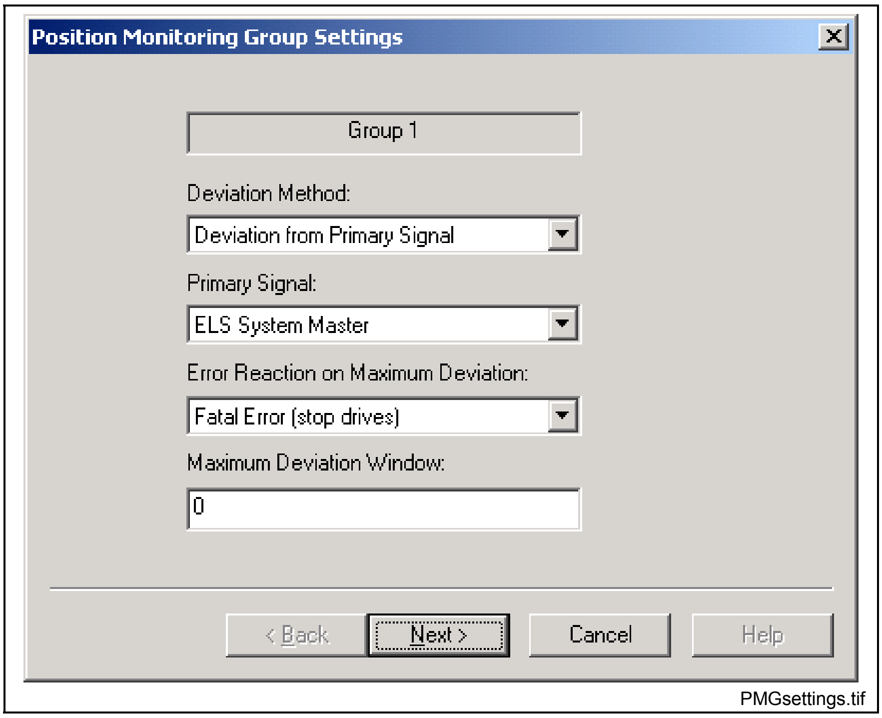


Fig. 5-24: Position Monitoring Group Settings

Deviation Method

This selection sets the method of deviation that will be used in the system. The selections include:

- Deviation from Primary Signal
- Min/Max Group Deviation Window

Deviation from Primary Signal

The system monitors the maximum allowable position difference between the selected primary (master) signal and the group of slave signals.

How it works:

During runtime, the group's corresponding enabled bit (PMG#_ENABLE, REG. 86) is checked. If enabled, the group's maximum deviation window is calculated. This calculated value is compared to each of the slave's position, including offset, to determine if it exceeds the maximum deviation window value. If a slave's position is outside of the maximum deviation window, the selected error reaction is executed.

Min/Max Group Deviation Window

The system monitors the maximum allowable position difference between the slaves in a group and compares it to the set maximum deviation window.

How it works:

During runtime, the group's corresponding enabled bit (PMG#_ENABLE, REG. 86) is checked. If enabled, each slave's position, including offset, in a group is analyzed. The slaves are sequentially checked based on their order. Next, the Min/Max value is derived by taking the absolute difference between the current Min and Max value and comparing it to the group's maximum deviation window. If the group's Min/Max value is greater than the maximum deviation window, the selected error reaction is executed.

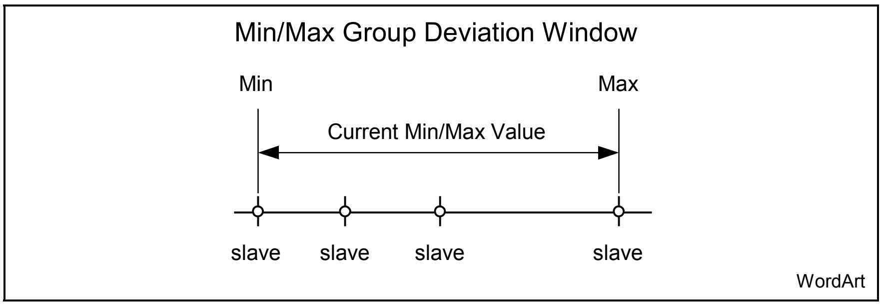


Fig. 5-25: Analyzing Min/Max Value

Primary Signal

The primary signal is the master signal that is used when the deviation method is set to "Deviation from Primary Signal". The available primary signal sources are listed in the table below.

Primary Signal	Description
Feedback Position (A-0-0102)	An axis' feedback position is used as a master signal
Command Position (A-0-0101)	The axis' command position value is used as a master signal
ELS System Group Position	An ELS Groups (1-8) output position is used as a master signal
ELS System Master	An ELS System Master (1-6) is used as a master signal

Table 5-10: Primary Signals

Rexroth
Bosch Group

Error Reaction on Maximum Deviation

This selection sets the error reaction that is used when a slave axis is outside the set maximum deviation window. When set to *Fatal Error (stop drives)*, all slaves axis will stop and the control will display the error, “554 Excessive Deviation in PGM%d, see ext. diag”. When set to *Warning (user defined)*, motion to all slaves continues and the control will display the warning, “220 Excessive Deviation in PGM%d. see ext. diag”.

Maximum Deviation Window

The value in this field represents the maximum positional deviation between the primary signal and slave axes, when using *Deviation from Primary Signal,* and between a minimum and maximum axes positions in a group, when using *Min/Max Group Deviation Window*. This value is stored in control parameter C-0-32x1 (PGM # Maximum Allowed Deviation).

Note: The maximum allowable value that can be entered is 90. The PMG uses the current system unit settings.

The following figure illustrates how the maximum deviation value is applied to both Deviation Methods.

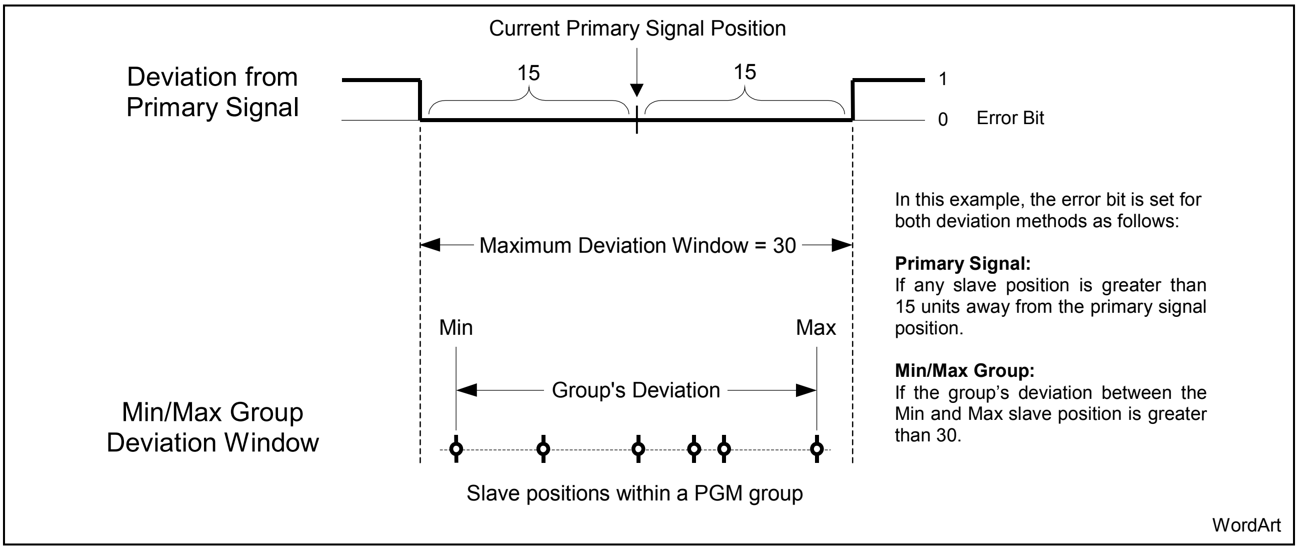


Fig. 5-26: Maximum Deviation Window

Note: Changes to the maximum deviation window, while a group is enabled, has no immediate effect. The group’s control register enable bit (Reg. 86 bits 1-8) must be disabled and then set before a deviation window change can take effect.

Error Detection and Reset Monitoring Groups

The system monitors the maximum allowable position difference between any number of drives identified in control parameter C-0-32x2 (PGM # List of Axis). Should one or more axes exceed the maximum deviation window value, the error bit(s) (Reg. 87, bits 9-16) will be set.

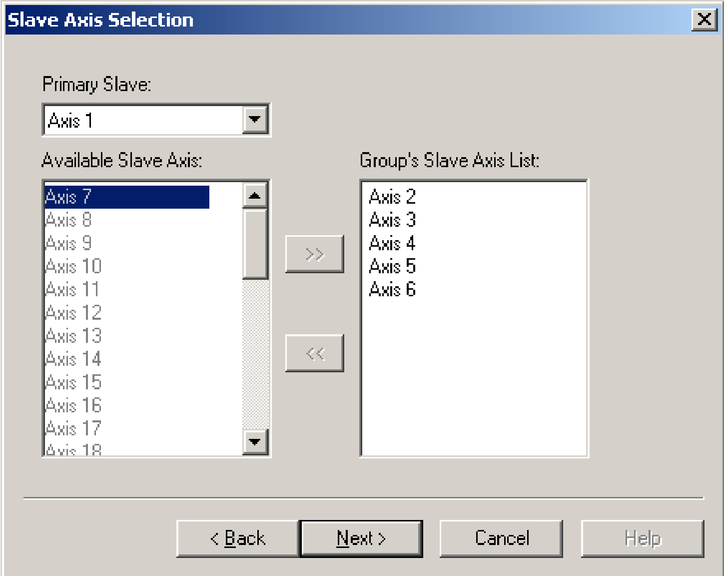
Note: PMG error bits are set to 0 when a PMG group is disabled.

Note: The programmer must react to these conditions and program a corrective routine based on the machine’s design and/or requirements.

One method could be to arm an Extended Input or Task Interrupt event to fire during the error. After the error has been detected and corrected, the programmer must rearm monitoring by transitioning the group's control register bit (Reg. 86, bits 1-8) from Low (0) to High (1). At this point, if all axes are within the deviation window, the status register bit (Reg. 87, bits 1-8) will be set High (1) and the error bit set Low (0). If any of the axes is outside of the deviation window, then the status register bit will remain Low (0) and the error bit will remain High (1).

Slave Axis Selection

The Slave Axis Selection window is used to select the primary and group slave axis for each group. A maximum of 5 group slave axes can be set for 1 primary slave.



SlaveAxisSelect.tif

Fig. 5-27: Slave Axis Selection

Primary Slave

The available selections within the **Primary Slave** field are dependent upon the type of Primary Signal selected in the previous window. The selected primary slave is stored in control parameter C-0-32x2 (PGM # List of Axis) as the first entry. The allowable primary slave types are listed based on selected primary signals.

Primary Slave	Primary Signal
Axis 1-64	Feedback Position (A-0-0102) or Command Position (A-0-0101)
Groups 1- 8	ELS System Group Position
ELS System Master 1- 6	ELS System Master

Table 5-11: Available Primary Signals

Note: When an axis number is selected as a primary slave, it is removed from the **Available Slave Axis** field.

Rexroth Bosch Group

Adding Slave Axes to a Group

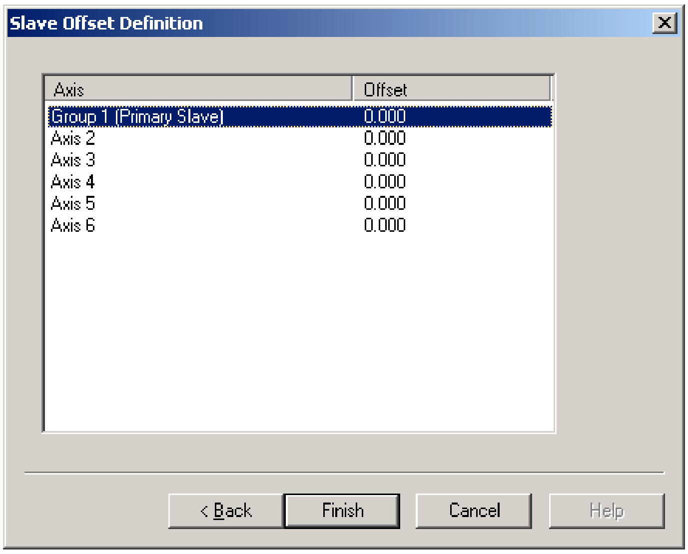
A maximum of 5 slave axes can be added to a *Groups Slave Axis List*. Axis are added and removed from the group's list by selecting the desired axis number and clicking on the following buttons:

>> Add a Slave Axis

<< Remove a Slave Axis

Slave Offset Definition

The *Slave Offset Definition* window is used to manually set an offset position between the Primary Slave and up to 5 additional slave axes in a group.



SlaveOffsetDef.tif

Fig. 5-28: Slave Offset Definition

Note: Slave offsets must be set after all axes in the system have been referenced for the application.

Manually Set a Slave's Offset

To set an offset, double-click on any axis to open the *Axis Offset* window. The primary slave offset is normally set to zero. All other axes are assigned an offset value to account for the difference in position from the primary slave.

For example:

If the primary slave's actual feedback position were 15 units, then the slave's offset would be set as follows:

Axis	Feedback Position	Offset
Group 1 (Primary Slave)	15	0
Axis 2	20	-5
Axis 3	16	-1
Axis 4	11	4
Axis 5	09	6
Axis 6	10	5

Table 5-12: Slave Offset

Automatically Set a Slave's Offset

Slave offsets can be automatically calculated by the system to ensure accurate values. This calculation is necessary for axes that have a fixed phase deviation in an application.

Note: Offset calculations are performed before a group is enabled and used to adjust all axes positions to zero. The actual feedback position for each slave is not physically moved, but the system uses the feedback + offset position to zero out all axes positions. From a zero starting value, the PMG feature can monitor all axes positions from a zero value and issue an error if any slave is outside the Maximum Deviation Window.

Control Register 86 (PMG_Control), bits 9-16 are used to calculate the required slave axis offset for each group. After a PMG is commissioned, the offset bits can be programmed as follows:

- Group offset bits can be set using an I/O Setup icon
- A minimum Wait of 2 ms must follow the setting of an offset bit before enabling the group. This provides sufficient time for the bit to set.

Note: Changes to a group's position offset, while a group is enabled, has no immediate effect. The control register enable bit (Reg. 86 bits 1-8) for the group must be disabled and then re-enabled before a position offset change takes effect.

5.4 PID

VisualMotion can support up to 32 PID (Proportional, Integral, Derivative) control loops with each program. These PID's are parameterized with program variables or registers, and have a minimum update rate of 8ms. A choice of optional filters (Low-pass or Butterworth) may be applied to the feedback signal.

The PID causes corrective action to be taken before a problem becomes unmanageable. For instance in a machine dispensing chocolate onto a conveyer the chocolate must be kept at a certain temperature so as to hold it's form once dispensed but not thicken too early restricting the flow rates. The PID function would keep the temperature within the too hot and too cold limits much more precisely than a simple on off switch controlled by a thermostat.

The PID instruction is activated at program activation with communication in Sercos phase 2 or greater and it's control register "PID Enable" (bit 5) set. The tasks do not need to be running. PIDs are configured under ***Setup ⇒ Processes ⇒ PID Loops***. Refer to section 4.7, *PID Loops Setup*, for details. The runtime tool is available in online mode under menu selection ***Data ⇒ PID…***

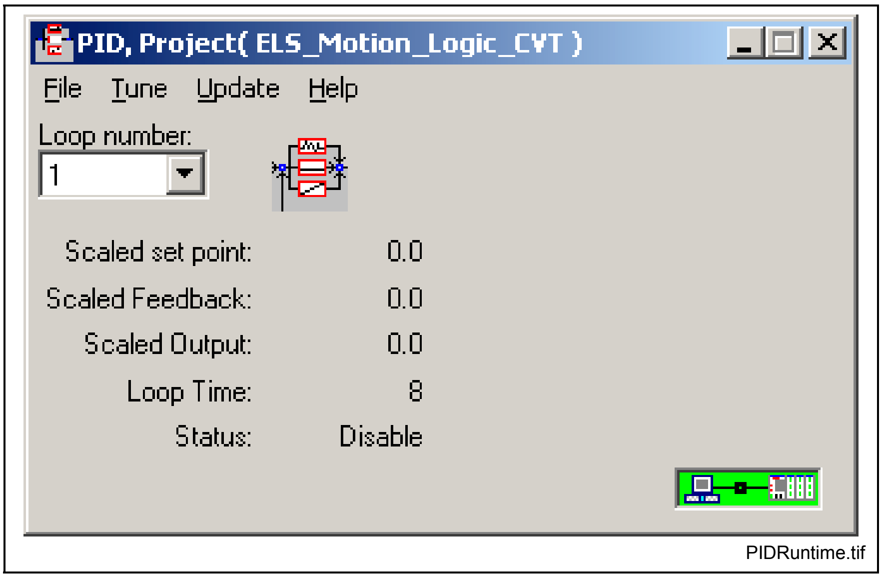


Fig. 5-29: PID Runtime Tool

PID Properties

Selecting ***File ⇒ Properties*** in the PID runtime tool opens the PID Properties window below. This window displays the variables, registers and Control Block start float that was setup under ***Setup ⇒ Processes ⇒ PID Loops***.

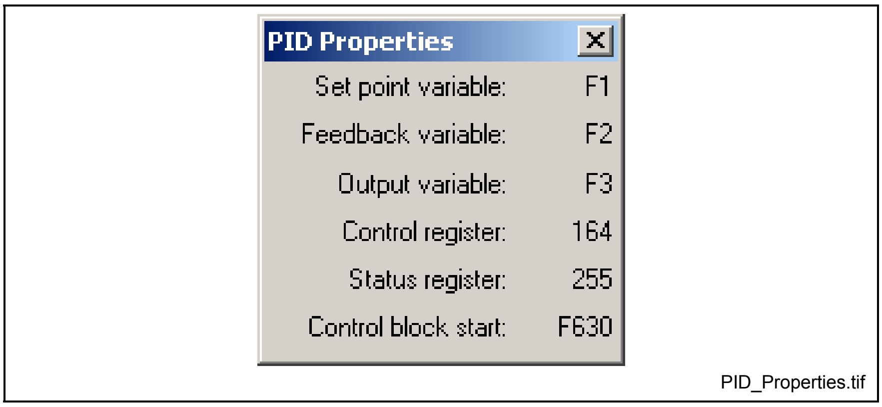


Fig. 5-30: PID Properties

Tune PID Control Block

Selecting **Tune** opens the *Tune PID Control Block* window. The tuning screen is used to adjust the selected PID loop while monitoring its values on the main screen. This screen has several grouping for scaling and adding offset to the process variables.

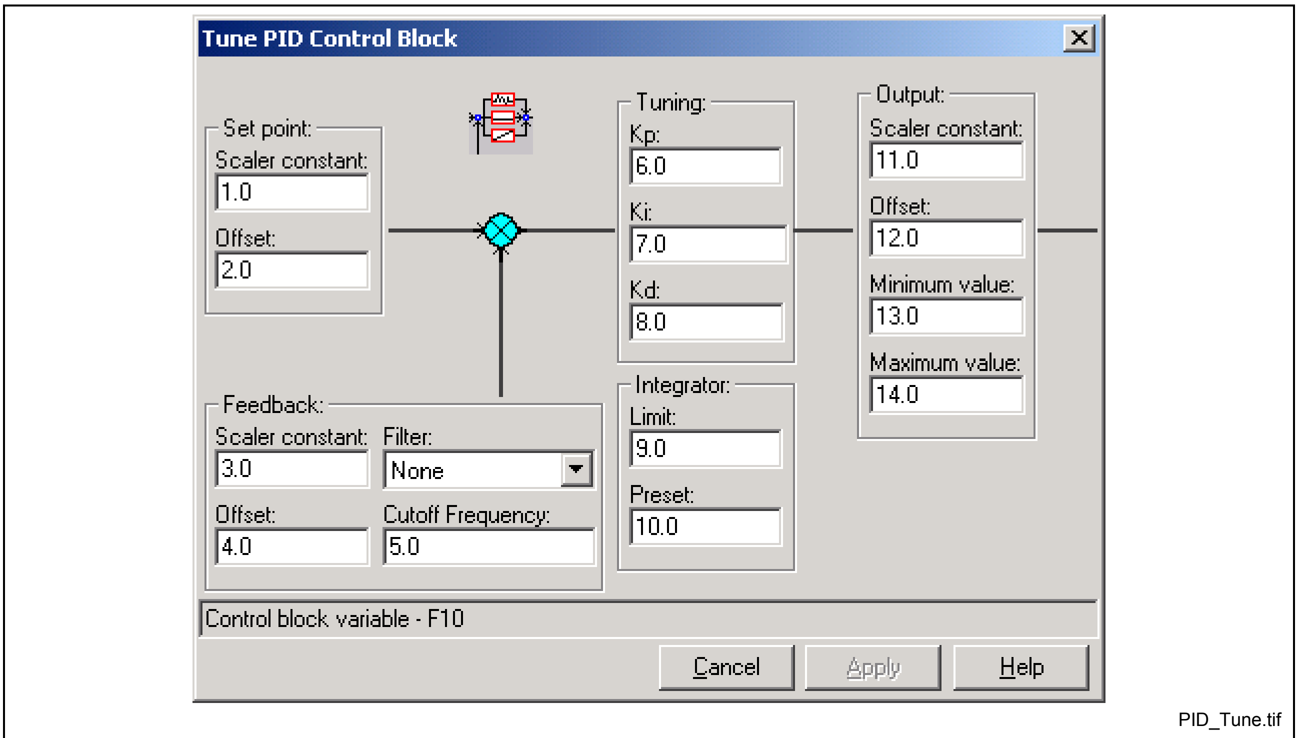


Fig. 5-31: Tuning PID

A grouping adjusting the Kd, Kp, Ki, integral preset, and integral limit are provided. The output grouping also has min, max limits for it. The feedback has an optional digital filter to condition the signal.

Refer to section 4.7, *PID Loops Setup*, for details

PID Instruction

The PID instruction configures a PI or PID control loop. The set point, feedback, and output variables can be registers, integers, floats, or parameters; appropriate conversions are supplied. Control factors (Ki, Kp, Kd, Last_I_Result_Preset) and limits (min., max.) can be constants or variables. Minimum loop update time is 8 milliseconds. In operation, the PID instruction only needs to be executed once in the program flow. The label for the PID loop is its control registers label.

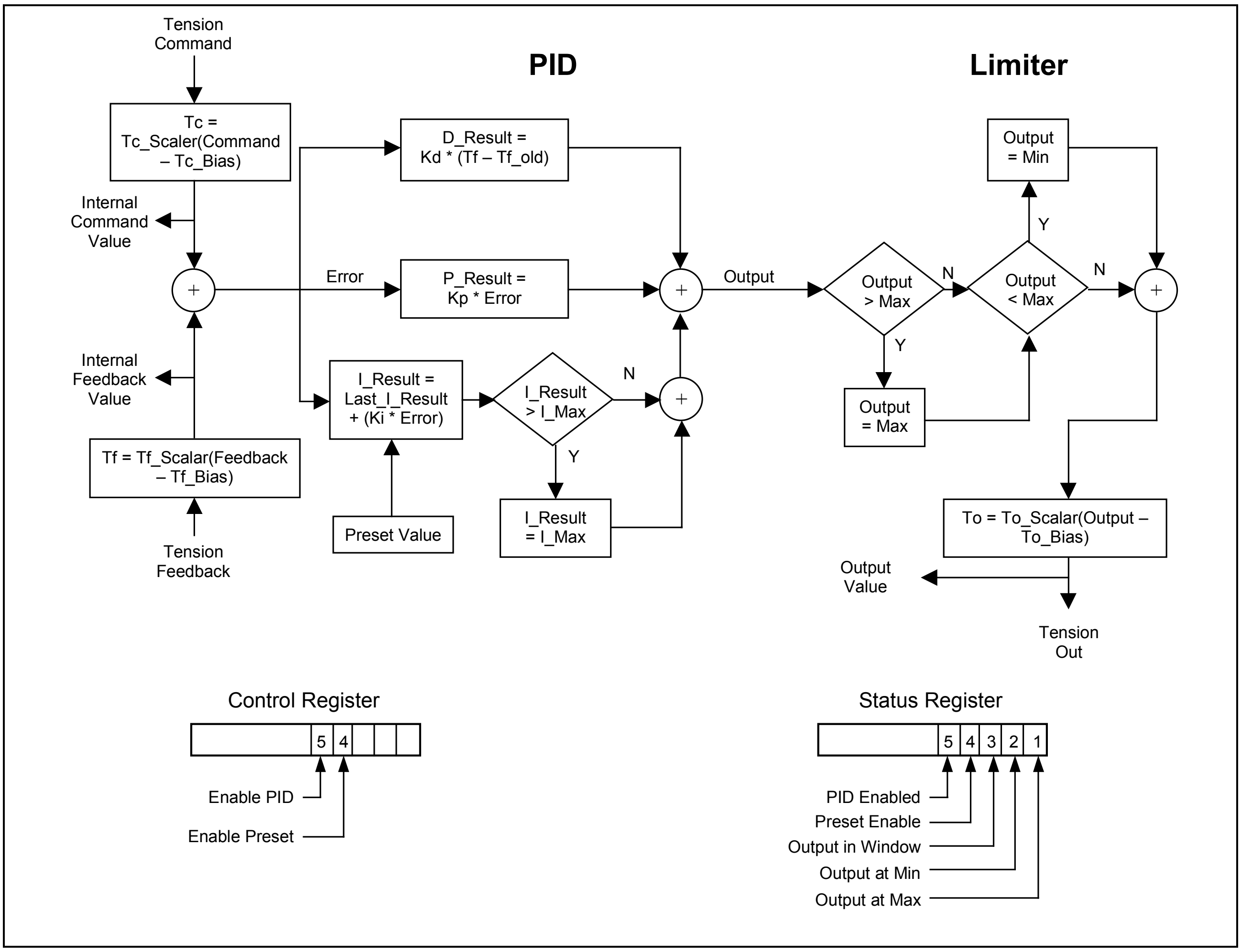


Fig. 5-32: PID Instruction

5.5 Programmable Limit Switch Functionality

A Programmable Limit Switch (PLS) is used to switch on and off digital outputs based on the input position of an associated axis or master. A PLS can be parameterized in offline, online or service mode. The basic component of a PLS is referred to as a PLS object.

PLS Object

A PLS object receives an input position from one associated axis or master and switches on and off digital outputs based on the programmed on and off positions.

The following figure illustrates a basic PLS object.

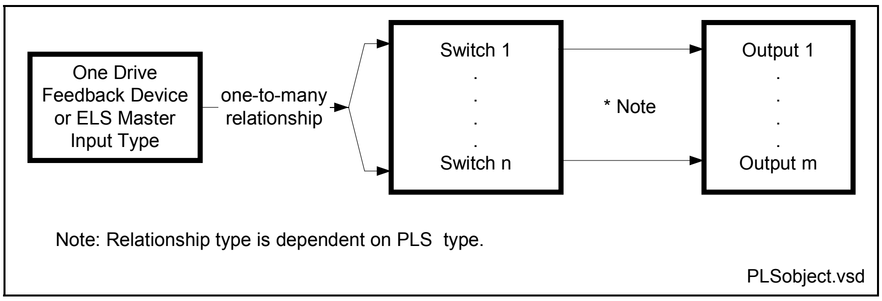


Fig. 5-33: Basic PLS Object

VisualMotion 11 supports the following three PLS object types:

- 2 Control PLS objects
- 1 Drive PLS object per drive (up to 64 drives total)
- Up to 8 Option Card PLS objects
  (one Option Card PLS hardware per control)

Control PLS Object

Up to 2 Control PLS objects are available in a VisualMotion project. The following table provides detailed specification for a Control PLS object:

PLS Type	Switches	Outputs	Input Type
Control PLS	16 switches per PLS object. 1 switch per output.	16 outputs per PLS object assigned to a single VisualMotion register. Register bits are automatically assigned starting from bit 1.	ELS System Master (1-6) ELS Group (1-8) Drive 1-64 - Motor encoder (S-0-0051) - External encoder (S-0-0053)

Table 5-13: Control PLS Object

Control PLS Object Relationships

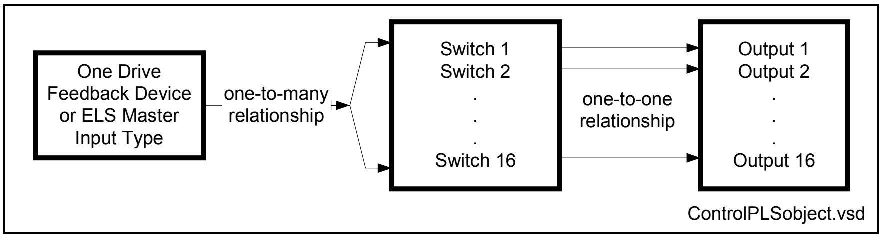


Fig. 5-34: Control PLS Object Relationships

One drive feedback device or master input position is assigned to all 16 switches. This is known as a one-to-many relationship.

The switches and outputs have a one-to-one relationship. Each switch number corresponds to the same output number.

The user assigns all 16 outputs to a 16-bit VisualMotion register. The 16 outputs are automatically assigned to each bit. Output 1 is assigned to bit 1, output 2 is assigned to bit 2, and so on.

Once parameterized and downloaded to the control, the status of each output, via the VisualMotion register, can be used by the user program.

Control PLS Data Storage

Unlike a Drive or an Option Card PLS that use parameters for data storage, two control PLS data structures exist in each user program with initial values of zero.

Drive PLS Object

Rexroth Diax 04 and EcoDrive 03 digital drives support 1 PLS object per drive in a Sercos ring. There can be a total of 64 Drive PLS objects to match the number of drives in a system. The following table provides detailed specifications for a Drive PLS object

PLS Type	Switches	Outputs	Input Type
Drive PLS	Up to 8 switches for Diax 04 Up to 16 switches for EcoDrive 03 1 switch per output.	8 outputs for Diax 04 16 outputs for EcoDrive 03 Outputs are assigned to a single VisualMotion register. Register bits are automatically assigned starting from bit 1.	Drive 1-64 - Motor encoder (S-0-0051) - External encoder (S-0-0053)

Table 5-14: Drive PLS Object

Drive PLS parameter support is dependent on the firmware version installed on the drive. Drive PLS parameters can also be written via the icon program and activated during runtime.

Diax 04 Drive

Up to 8 switches can be assigned to a Diax 04 Drive PLS. Each switch can also be assigned an individual lead-time. The following Diax 04 drive firmware supports 8 switches:

- FWA-DIAX04-ELS-04VRS-MS
- FWA-DIAX04-ELS-05VRS-MS
- FWA-DIAX04-SMT-01VRS-MS

Drive-based I/O (i.e., DEA4.2M card) can be configured for a PLS output. Refer to “PLS Register Assignment” on page 5-44 for details.

EcoDrive 03 Drive

Up to 16 switches can be assigned to an EcoDrive 03 Drive PLS. Each switch can also be assigned an individual lead-time. The following EcoDrive 03 drive firmware supports 16 switches:

- FWA-ECODR3-SGP-01VRS-MS
- FWA-ECODR3-SGP-03VRS-MS
- FWA-ECODR3-SGP-20VRS
- FWA-ECODR3-SMT-02VRS-MS
- FWA-ECODR3-MGP-01VRS-MS

IndraDrive using MPx firmware

Up to 8 switches can be assigned to a IndraDrive drives using MPx03 firmware. Each switch can also be assigned an individual lead-time. The following IndraDrive firmware supports 8 switches:

- FWA-INDRV*-MPx-03VRS-D5-1-SNC-ML
- FWA-INDRV*-MPx-03VRS-D5-1-SNV-ML
- FWA-INDRV*-MPx-04VRS-D5-1-SNC-ML
- FWA-INDRV*-MPx-04VRS-D5-1-SNV-ML

Drive PLS Object Relationships

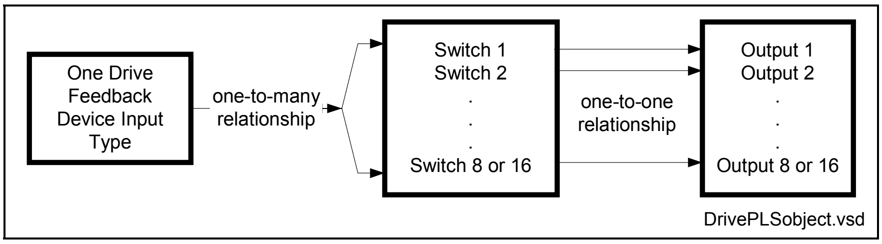


Fig. 5-35: Drive PLS Object Relationships

Depending on drive firmware, up to 8 or 16 switches can receive the input position from one drive feedback device. This is known as a one-to-many relationship.

The switches and outputs have a one-to-one relationship. Each switch number corresponds to the same output number.

The user assigns up to 8 or 16 outputs to a 16-bit VisualMotion register. The outputs are automatically assigned to each bit. Output 1 is assigned to bit 1, output 2 is assigned to bit 2, and so on.

Once parameterized and downloaded to the control, the status of each output, via the VisualMotion register, can be used by the user program.

Parameters Relevant to a Diax 04 and EcoDrive 03 Drive PLS

The following table contains a list of parameters valid for Diax 04, EcoDrive 03, and IndraDrive MPx03/04 drive-based PLSs. Refer to the relevant Digital Drive Functional Description manuals for details.

Parameters		Description	Supported Drives		Updated
			Diax 04 / EcoDrive 03	IndraDrive MPx03/04	
Drive Parameters	P-0-0130	Position Switch Signal Selection List		X	read-only
	P-0-0131	Position Switch Signal Select ion	X	X	Phase 2
	P-0-0132	Position Switch On Threshold	X	X	Phase 4
	P-0-0133	Position Switch Off Threshold	X	X	Phase 4
	P-0-0134	Position Switch Lead Time	X	X	Phase 4
	P-0-0135	Position Switch Status Word	X	X	Phase 4
Control Parameters	A-0-0009	Output Register Value	X	X	Phase 2

Table 5-15: Drive PLS Parameters

Note: IndraDrive drive-based PLS is only available in the Closed Loop/Servo (SRV) and the Closed Loop/Synchronization (SNC) packages.

IndraDrive PLS Signal Selection List

IndraDrive drive parameter P-0-0130 contains a list of valid PLS input signals. The desired input signal is written to parameter P-0-0131 in Sercos phase 2. The following input signals are supported:

- S-0-0000 No assignment
- S-0-0051 Position feedback 1
- S-0-0053 Position feedback 2
- P-0-0052 Actual position measuring encoder
- P-0-0434 Position command value of controller
- P-0-0775 Resulting master axis position
- P-0-0776 Effective master axis position
- P-0-0778 Synchronous position command

Option Card PLS Object

The Option Card PLS is optional hardware that can support up to 8 PLS objects. The hardware interface can be ordered with up 2 digital output modules (16 outputs per module) for both the PPC-R22.1 and PPC-P11.1. A maximum of 96 switches can be divided up among the 8 PLS objects. No switch can be shared between PLS objects or outputs.

Note: Refer to the VisualMotion 11 Project Planning manual for available hardware and firmware configurations.

PLS Type	Switches	Outputs	Input Type
Option Card PLS	96 total switches available for up to 8 PLS objects. Switches can be assigned to a single output or divided among configured outputs.	Number of outputs is dependent on hardware configuration. Up to 2 output modules (32 outputs) can be ordered for a control. 16 outputs per module. Switch to output assignment is user-defined.	ELS System Master (1-6) ELS Group (1-8) Drive 1-64 - Motor encoder (S-0-0051) - External encoder (S-0-0053)

Table 5-16: Option Card PLS Object

Rexroth
Bosch Group

Note: The maximum speed for any Option Card PLS input master is 3500 rpm. Only one Option Card PLS can be ordered for a PPC control.

Option Card PLS Object Relationships

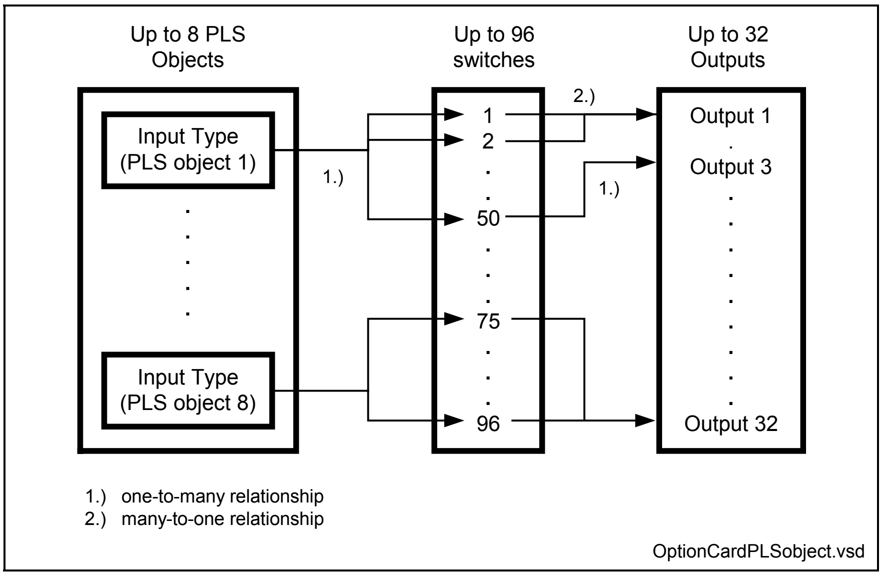


Fig. 5-36: Option Card PLS Object Relationships

Up to 96 switches can be used for up to 8 PLS objects. Switches contained in a PLS object can receive the input position from one drive feedback device or ELS Master. This is known as a one-to-many relationship.

Note: The same input type can be used for more than one PLS object. For example, ELS Master 1 can be the configured input type for 2 or more PLS objects.

The switches and outputs can have a one-to-one relationship or a many-to-one relationship. Any number of switches, up to a total of 96, can be assigned to one output or divided up among the 32 available outputs. However, once a switch is assigned to an output it can not be used by a different output.

The user can assign up to 16 outputs (one output module) to a 16-bit VisualMotion register. If the control is ordered with 2 output modules (32 outputs), the user assigns the first register and the second consecutive register is automatically assigned to the second output module. The outputs are automatically assigned to each bit. Output 1 is assigned to bit 1, output 2 is assigned to bit 2, and so on.

Once parameterized and downloaded to the control, the status of each output, via the VisualMotion register, can be used by the user program. The output module(s) installed with the control provide high-speed access to the available 32 outputs.

Parameters related to the Option Card PLS

The parameters for an Option Card PLS can be grouped into four categories:

- General Parameters (C-0-2901 through C-0-2910)
- Switch Settings (C-0-2920 through C-0-2922)
- Output Settings (C-0-2930 through C-0-2935)
- PLS Input Settings (C-0-2940 through C-0-2943)

The following tables list each parameter with a description and the control phase in which it is updated.

General Parameters

Parameter	Description	Updated
C-0-2901	PLS1 Start Output Register	Phase 4 (read/write)
C-0-2902	PLS1 Start Mask Register	Phase 4 (read/write)
C-0-2903	PLS1 Build Table Command	Phase 4 (read/write)
C-0-2904	PLS1 Build Table Status	Phase 4 (read)
C-0-2905	PLS1 Switch Table Command	Phase 4 (read/write)
C-0-2906	PLS1 Switch Table Status	Phase 4 (read)
C-0-2907	PLS1 Error Code	Phase 2 (read) / Phase 4
C-0-2908	PLS1 Extended Error Code	Phase 2 (read) / Phase 4
C-0-2909	PLS1 Hardware ID	Phase 2 (read) during power up
C-0-2910	PLS1 Software ID	Phase 2 (read) during power up

Table 5-17: General Parameters for an Option Card PLS

Switch Parameters

Parameter	Description	Updated
C-0-2920	PLS1 Switch On List	Phase 4 (read/write)
C-0-2921	PLS1 Switch Off List	Phase 4 (read/write)
C-0-2922	PLS1 Switch Output List	Phase 4 (read/write)

Table 5-18: Switch Parameters for an Option Card PLS

Output Parameters

Parameter	Description	Updated
C-0-2930	PLS1 Output Master List	Phase 2 (read/write)
C-0-2931	PLS1 Output Lead Time	Phase 4 (read/write)
C-0-2932	PLS1 Output Lag Time	Phase 4 (read/write)
C-0-2933	PLS1 Output One Shot (PT) List	Phase 4 (read/write)
C-0-2934	PLS1 Output Mode List	Phase 4 (read/write)
C-0-2935	PLS1 Output Direction List	Phase 4 (read/write)
C-0-2936	PLS1 Output Hysteresis	Phase 2 (read/write)

Table 5-19: Output Parameters for an Option Card PLS

PLS Master Parameters (Input Type)

Parameter	Description	Updated
C-0-2940	PLS1 Master Type List	Phase 2 (read/write)
C-0-2941	PLS1 Master Number List	Phase 2 (read/write)
C-0-2942	PLS1 Master encoder List	Phase 2 (read/write)
C-0-2943	PLS1 Master Phase Offset List	Phase 4 (read/write)

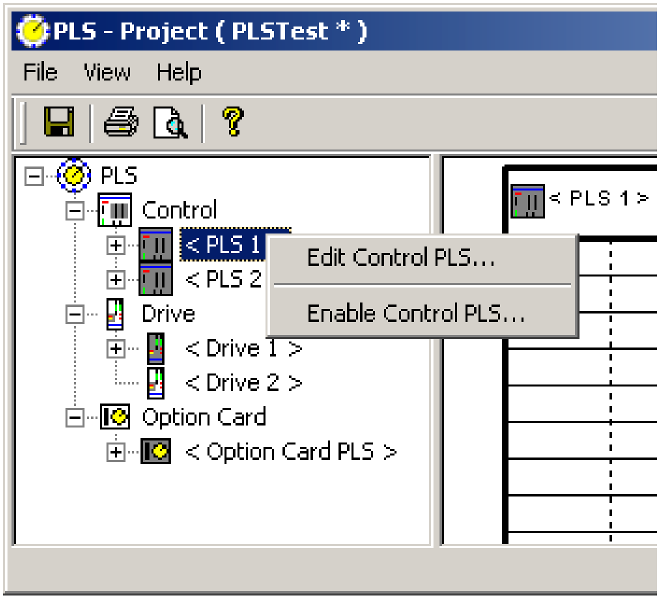
Table 5-20: PLS Master Parameters for an Option Card PLS

Programming Modes

A Control, Drive, or Option Card PLS (if equipped) can be enabled and configured using the VisualMotion PLS tool. PLS configurations can be parameterized or edited in offline, online, or service mode. Online and service modes require a serial or EtherNet connection to the control. The PLS tool is launched by selecting ***Commission ⇒ PLS…***

Enable/Disable a PLS Configuration

The enabled state of a PLS configuration is displayed in the PLS Tool tree structure. An icon with a white background indicates that the PLS configuration is enabled. An icon with a gray background indicates that it is disabled. A PLS configuration is enabled when a PLS master type is assigned. In offline mode, a PLS configuration can be enabled or disabled by right-clicking on the PLS icon and selecting Enable PLS or Disable PLS. This option is only available in offline mode. Disabling a PLS simply replaces the assigned PLS master type with *Not Set*. This feature provides the user with the option to enable or disable a PLS configuration before switching to online mode.



EnablePLS.tif

Fig. 5-37: Enable or Disable a PLS Configuration

Offline Programming of a PLS

Offline programming is used to parameterize PLS configurations when a physical connection to a control is not possible. When opened in offline mode and no PLS configurations exists, the PLS tool displays 2 Control PLSs, 1 Drive PLS and an Option Card PLS in the navigation window.

If a PLS configuration is being parameterized for a new project, it is recommended that you first go online with your project to establish the hardware configuration with the control. This procedure uploads all Diax 04 and EcoDrive 03 drives found on the Sercos ring to the current project.

Note: IndraDrive drives using MPx02 firmwares do not support PLS configurations.

Downloading Hardware Configuration from Control

Perform the following steps to download the current hardware configuration from the control:

1. With a serial or EtherNet connection to the control, establish communication with VisualMotion Toolkit.

2. From VisualMotion's main menu, select ***File*** ⇒ ***Import Project Components***.
3. Select the *Hardware Configuration* checkbox and click the **OK** button.

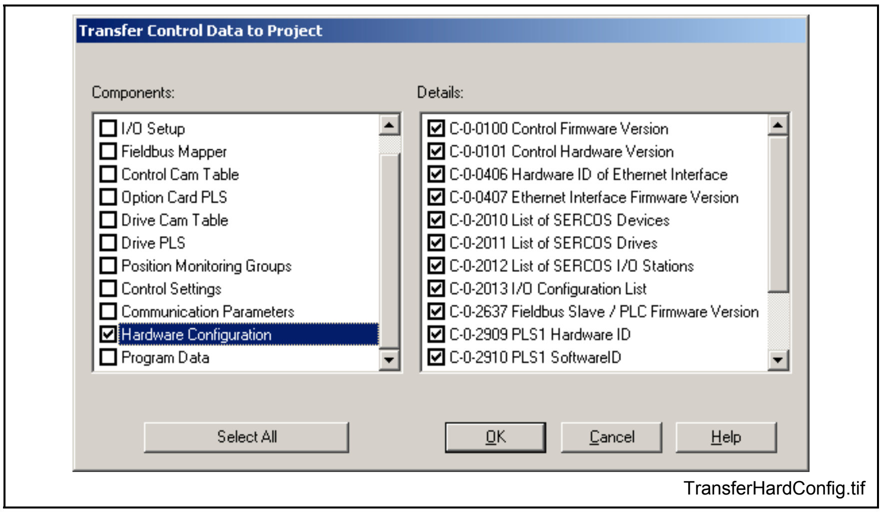


Fig. 5-38: Transfer Control Data to Project

The hardware configuration of all the drives in the Sercos ring has been saved to the offline project data. Now, all PLS configurations, including Drive PLS can now be configured without a connection to the control.

Note: If no control is available, a Drive PLS can still be manually added and parameterized using the PLS Tool. Refer to the steps on page 5-39.

Add a Drive PLS

The PLS Tool displays a Drive icon place holder with no configured Drive PLS when no hardware configuration is downloaded to the control.

Perform the following steps to add a Drive PLS:

1. Right-click on the PLS tree icon and select Add Drive PLS…

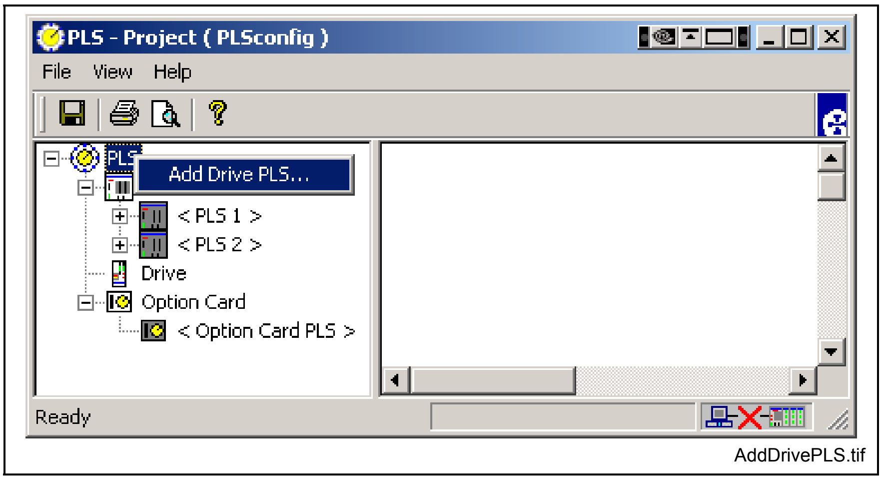


Fig. 5-39: Add Drive PLS

2. Select the drive type (ECODRIVE03 or DIAX04) and specify the Drive # (Sercos Address).

Rexroth Bosch Group

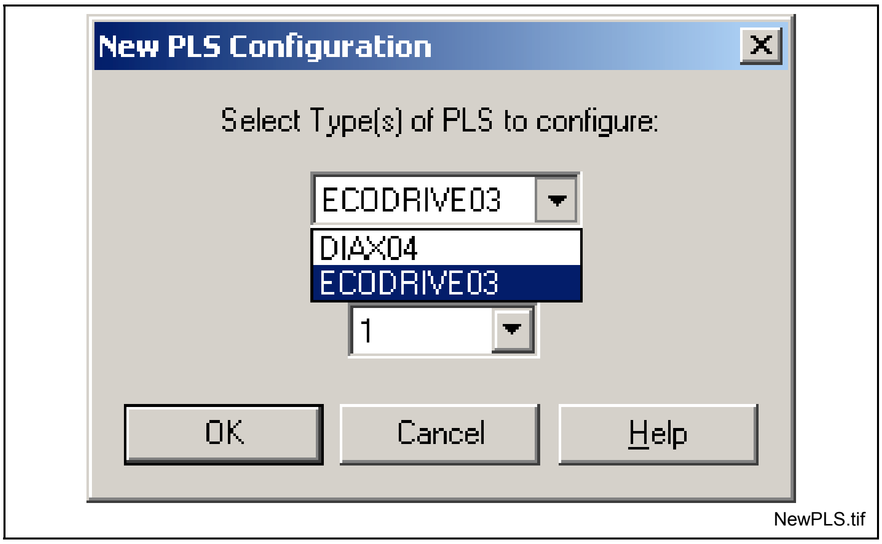


Fig. 5-40: New PLS Configuration

3. Once a drive is selected, a PLS wizard will guide the user through the initial configuration. Refer to Configure a Control or Drive PLS on page 5-41 for details.

Online Programming of a PLS

When the PLS tool is opened in online mode, all control, drive and Option Card PLSs are uploaded from the control's memory and displayed in the PLS tool window. All Diax 04 and EcoDrive 03 drives found on the Sercos ring will be uploaded with their assigned Sercos address.

To parameterize any PLS, simply double-click on one of the Control, Drive or Option Card second level icons and configure the switches, master, registers and outputs. Refer to Configure a Control or Drive PLS on page 5-41 for details.

Service Mode Programming of a PLS

Although service mode can be used to parameterize a PLS configuration and download it to the control, service mode should only be used to make any necessary modifications when the project data files are not available.

PLS configurations are uploaded from the control by selecting the upload icon or by selecting ***File ⇒ Get PLS Configuration from PPC Control***. To edit a PLS in service mode, refer to Service Mode on page 5-57 for details.

Note: PLS configurations edited in service mode are not synchronized with a project unless the PLS data is imported in to the project. Refer to Importing PLS Configurations on page 5-59 for details.

Configure a Control or Drive PLS

Due to the configuration similarities between Control and Drive PLSs, this section will only focus on their parameterization. For steps on how to configure an Option Card PLS refer to page 5-46.

Note: Control PLSs can be configured using the PLS tool or by placing a PLS2 icon in the Initialization task or any runtime task.

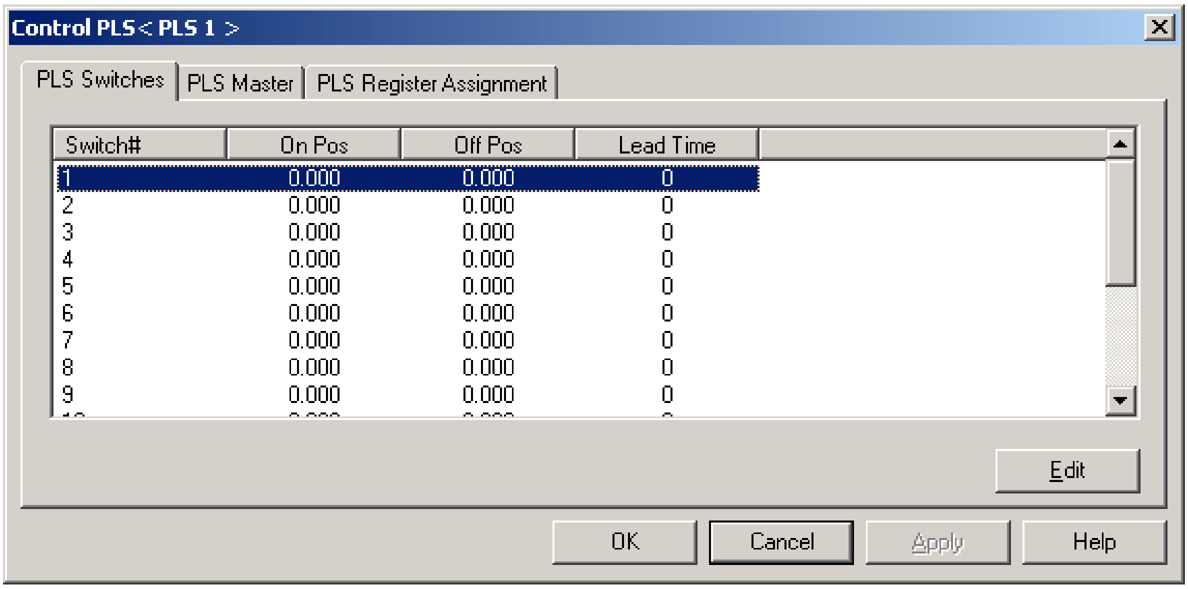
Use the following steps to configure a Control or Drive PLS:

1. Open the PLS Tool by selecting ***Commission*** ⇒ ***PLS…*** from VisualMotion's main menu.

Note: Control and Option Card PLSs are present by default. Drive PLSs are uploaded automatically in online mode. In offline mode, a Drive PLS is added by right-clicking on the main PLS icon and selecting Add a Drive. Refer to Add a Drive PLS on page 5-39 for details.

2. Double-click or right-click on a second level Control or Drive PLS icon to open the *Control PLS or Drive PLS* window.

Note: A Control PLS is displayed with all 16 switches. A Drive PLS can have up to 16 switches, depending on drive firmware.



PLS_Switches.tif

Fig. 5-41: PLS Switches Tab

3. In the **PLS Switches** tab of the *Control PLS* window, assign a switch's On/Off position and lead time by double-clicking on the desired switch number or highlight the switch number and click on the ***Edit*** button.

Switch Configuration for Control or Drive PLS

The following steps configure a switch's On/Off position and lead-time.

1. Select the Switch index number.
2. In the *Switch Configuration* window, enter the On and Off positions and lead-time for the switch. Refer to page 5-51 for an explanation of lead time.

Note: The On and Off positions for each switch is relative to the units of measurement in axis parameter A-0-0005 and task parameter T-0-0005.

For example, if these parameters are set to 1, then the unit of measurement for the switch's On and Off positions is mm.

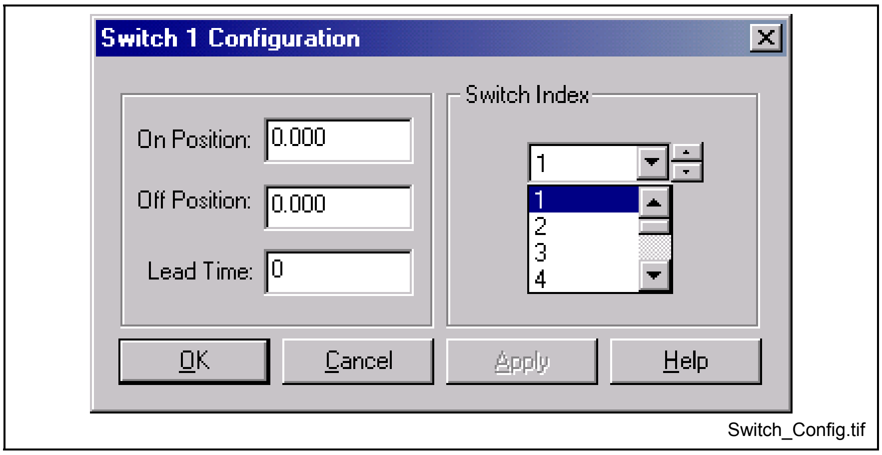


Fig. 5-42: Switch Configuration Window

3. Click the ***Apply*** button to save data.

Once a switch is configured and the ***Apply*** button is pressed, a new Switch Index number must be selected to configure a different switch.

4. When all the switches are configured, press the ***OK*** button to close the *Switch # Configuration* window and return to the *PLS Switches* tab.

Note: VisualMotion issues an error when a different Switch Index number is selected without first applying changes to an already configured switch.

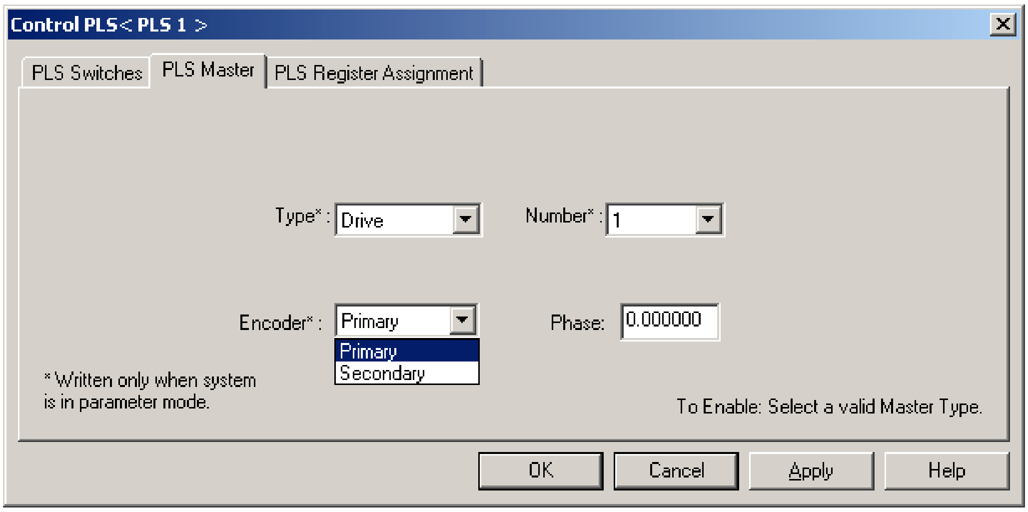
PLS Master Configuration for Control or Drive PLS

The *PLS Master* tab is used to configure 1 drive or master for each Control or Drive PLS.

Control or Drive PLS

Three Master types are available for a Control or Drive PLS:

- ELS Master (System Master) - An ELS Master can be a Virtual Master, the output of an ELS Group, a Real Master or the output of a second PPC-R cross-communicating in a Link Ring.
- ELS Group - In an ELS multiple master configuration, the output position of an ELS Group can be used as an input to the PLS. A *Phase* value added to the output of the ELS Group does not affect any phase offsets that might have been configured in the ELS Group.
- Drive (Diax 04 and EcoDrive 03) – Primary or Secondary encoder must be indicated when Drive type is selected.



Control_PLS_Master.tif

Fig. 5-43: PLS Master Configuration for a Control PLS

The *Number* field contains a drop-down list of numbers corresponding to the input type selected. The following table lists the input types and their range of available numbers.

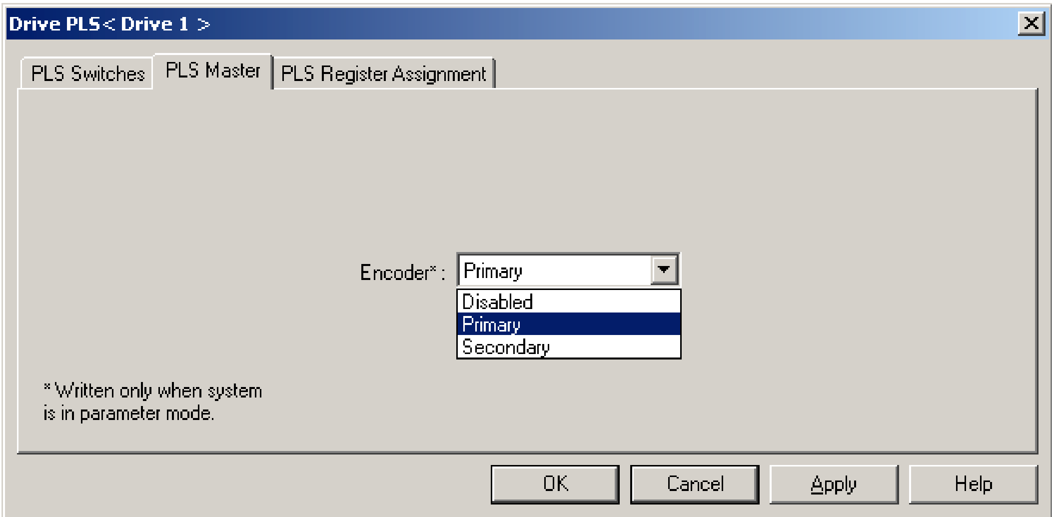
Type	Allowable Number Range	Encoder
ELS Master	1-6	N/A
ELS Group	1-8	N/A
Drive	1-64 (Sercos address)	Primary or Secondary feedback

Table 5-21: Control PLS Input Types

The *Phase* field allows you to enter an offset value that is added to the output of the PLS master.

Drive PLS The available Encoder types to use as a Drive PLS Master are:

- Primary
- Secondary



PLS_Master_Encoder.tif

Fig. 5-44: PLS Master Configuration for a Control PLS

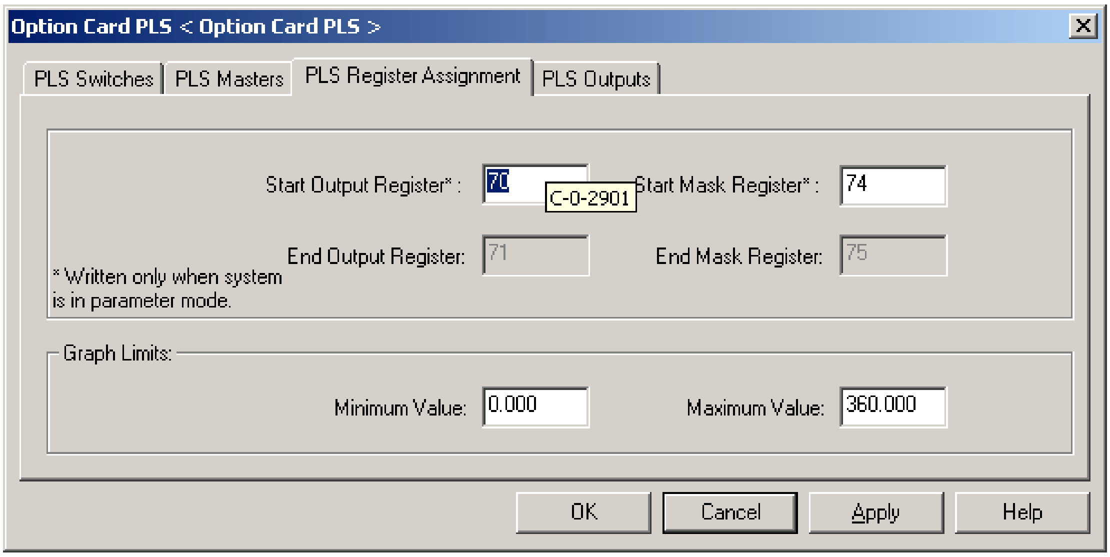
Rexroth Bosch Group

PLS Register Assignment for a Control or Drive PLS

The *Register Assignment* tab for a Control or Drive PLS has fields for configuring the Output register and Mask register. The Output register is assigned to the PLS for monitoring the status of each switch.

Control PLS

The 16 Control PLS switches are assigned to bits 01 to 16. The default state of the Output register is zero. With the Output register in the default state, the corresponding mask register bits must be set each time after power-up.



PLS_Reg_Assign.tif

Fig. 5-45: PLS Register Assignment for a Control PLS

The Mask register is used to force the state of the output register bits. At power-up, the bits in the mask register are set to 0 to prevent any unwanted outputs from becoming enabled. The Control PLS switches assigned to an output register will not function until the state of the register's complementary mask register bits are set to 1.

Note: The default state of the Output and Mask Registers is zero (0). The user must set the corresponding mask register each time after power-up.

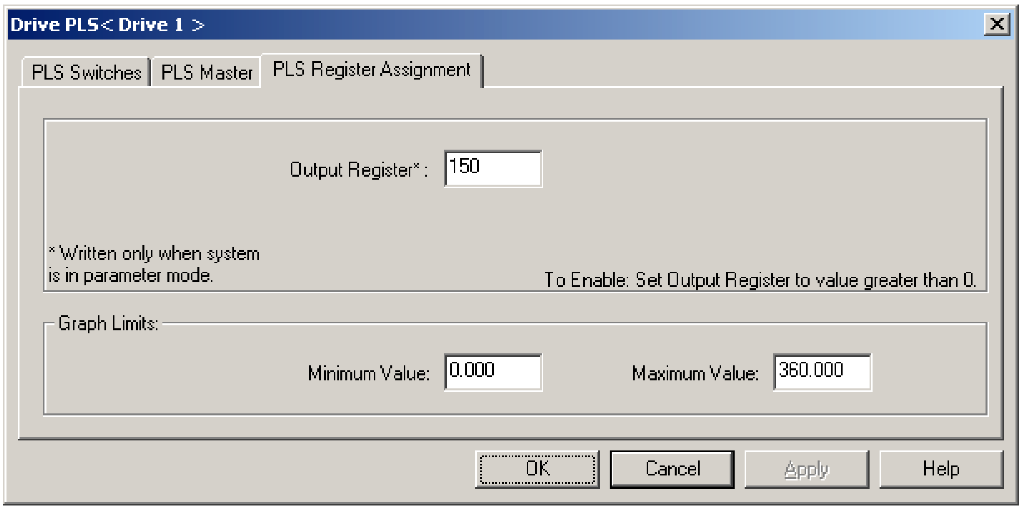
The Mask register bits cannot be set to 1 before the PLS is configured. After the PLS is configured and downloaded to the control, the state of each Mask register bit can be set to 1 by selecting ***Data*** ⇒ ***Registers*** from the VisualMotion main menu. Only an Output Register with a Mask Register set to 1 will output a signal in the Control PLS register.

Note: The **Mask Register** field for Control PLS 2 is inactive because the next consecutive number from the Control PLS 1 Mask Register number is automatically assigned. If the Control PLS 1 mask register is edited, Control PLS 2 mask register will automatically be updated.

Drive PLS

Diax 04 drive switches are assigned to bits 01-08 and EcoDrive 03 drive switches are assigned to bits 01-16.

Note: No Mask register is required for a Drive PLS. The Drive PLS output register is always active.



DrivePLS_Reg_Assign.tif

Fig. 5-46: PLS Register Assignment for a Drive PLS

Configuring Diax 04 Drive-based I/O Cards

VisualMotion can be used to automatically configure a Diax 04 Drive PLS's output signal directly to a DEA I/O card using the Output register for Drive PLS. The Diax 04 digital drive I/O cards that can be used to output Drive PLS signals are:

- DEA04.2M
- DEA05.2M
- DEA06.2M

To have VisualMotion automatically configure I/O cards to output Drive PLS signals:

1. Configure a Diax 04 Drive PLS for axis n and assign a VisualMotion register (for example, register 70) to the PLS *Output Register* using the PLS Tool.
2. Download the configured PLS to the control in parameter mode.

Note: The PLS Output Register assignment is written to axis parameter A-0-0009.

3. Using the I/O Configuration Tool, configure axis n with a DEA04.2M, 5.2M or 6.2M I/O card and assign the outputs to the same register number that was used in the PLS tool.
4. Download the I/O configuration to the control in parameter mode.

During phase initialization (P2 to P3), VisualMotion compares the register numbers assigned to the Diax 04 Drive PLS and the DEA4.2M, 5.2M or 6.2M I/O card. If the register numbers match, then drive parameter P-0-0124 is automatically configured with the IDN number of P-0-0135 and the type of I/O card configured.

During runtime (P4), the Drive PLS outputs (P-0-0135) are sent directly to the physical outputs on the drive's DEA4.2M, 5.2M or 6.2M I/O cards. At the same time, the Drive PLS outputs (P-0-0135) are also written to register 70, across the Sercos AT Cyclic Telegram, for use in the VisualMotion project. The following figure shows the runtime sequence of the configured Drive PLS outputs.

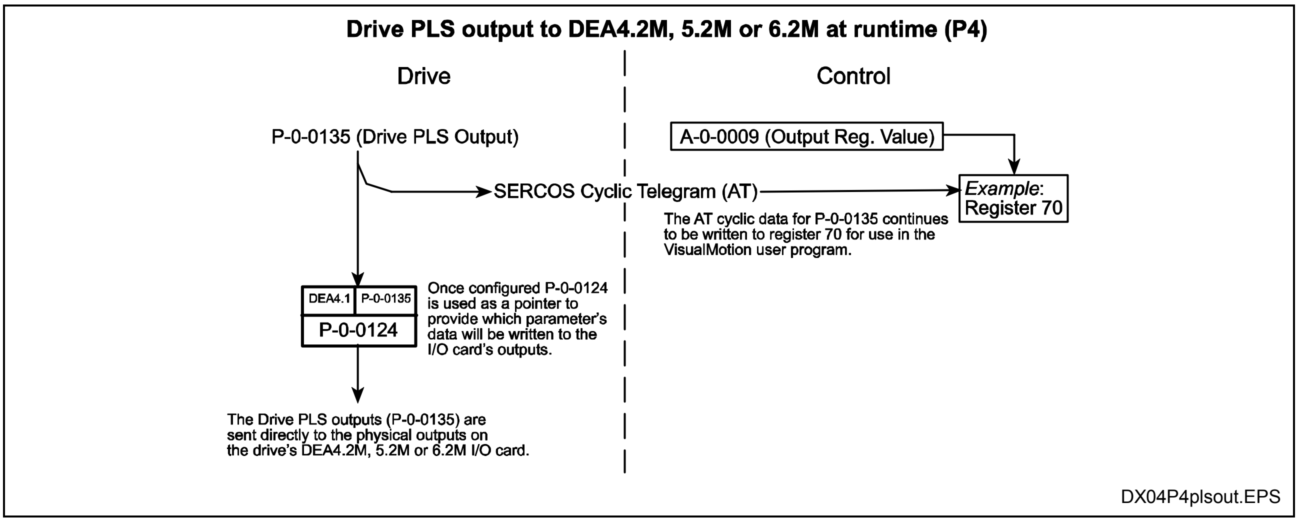


Fig. 5-47: DEA4 I/O Configured for Diax 04 Drive PLS Output

Configure an Option Card PLS

Use the following steps to configure an Option Card PLS:

1. Open the PLS Tool by selecting ***Commission*** ⇒ ***PLS…*** from VisualMotion's main menu.
2. Double-click or right-click on the second level Option Card PLS icon to open the *Option Card PLS* window.

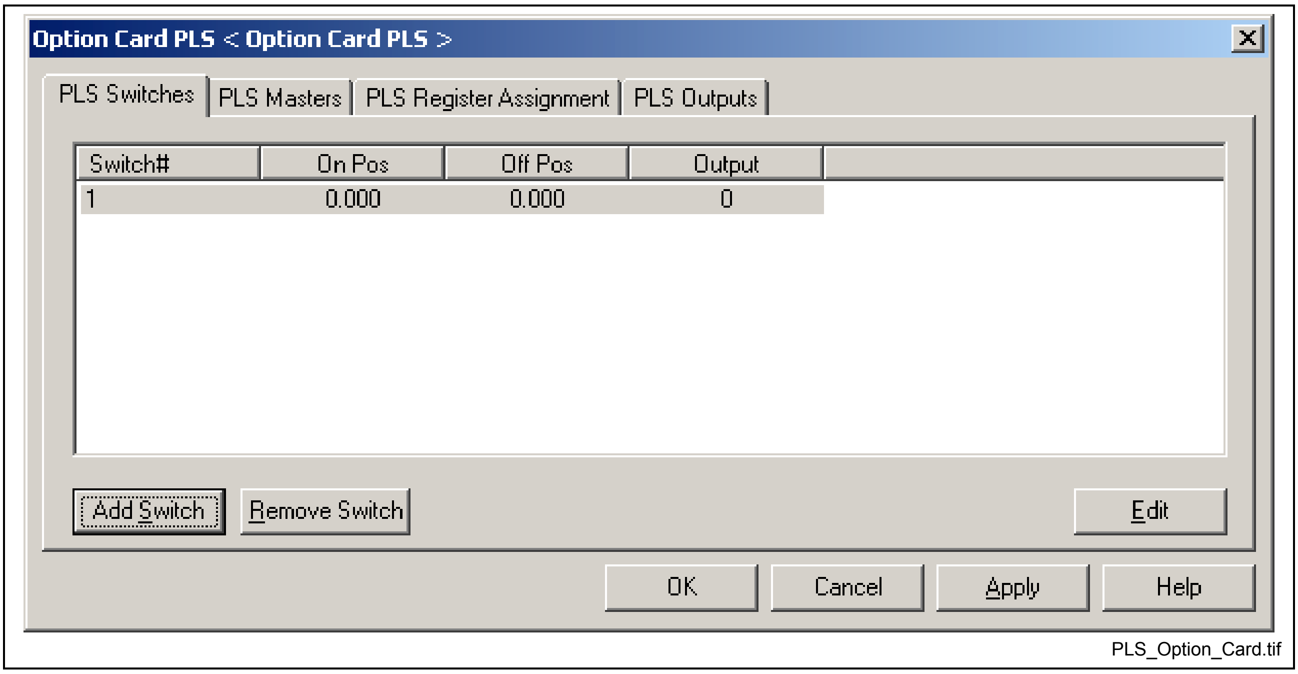


Fig. 5-48: New PLS Configuration

3. Option Card PLS switches are added and removed by clicking on the **Add Switch** or **Remove Switch** button.

Note: A total of 96 switches can be added to a Option Card PLS.

4. From the **PLS Switches** tab, double-click or right-click on a switch number to open the *Switch Configuration* window.

Switch Configuration for an Option Card PLS

The following steps configure a switch's On / Off position and assigns it to an output.

1. Select the Switch Index number.

Note: Once a switch is configured and the ***Apply*** button is pressed, a new Switch Index number must be selected to configure a different switch. Pressing the ***OK*** button accepts the values and closes the window. The current switch number is displayed in the header portion of the window.

Example:

Switch **1** Configuration

---

2. Enter the On and Off Positions.

Note: The On and Off positions for each switch is relative to the units of measurement in axis parameter A-0-0005 and task parameter T-0-0005.

Example:

If these parameters are set to 1 (mm), then the unit of measurement for the switch's On and Off positions is mm.

---

3. Enter an Output number for the switch. The user can assign up to 96 switches to one output or distribute them among the available 16 or 32 outputs.
4. When all the switches are configured, press the ***OK*** button to close the *Switch # Configuration* window and return to the *PLS Switches* tab.

Note: VisualMotion issues an error when a different Switch Index number is selected without first applying changes to an already configured switch.

PLS Master(s) Configuration for an Option Card PLS

The *PLS Master* tab is used to configure up to 8 PLS Masters. The available PLS types are:

Type	**Allowable Number Range**	**Encoder**
ELS Master	1-6	N/A
ELS Group	1-8	N/A
Drive (Diax 04 or EcoDrive 03)	1-64 (Sercos address)	Primary or Secondary feedback

Table 5-22: Option Card PLS Input Types

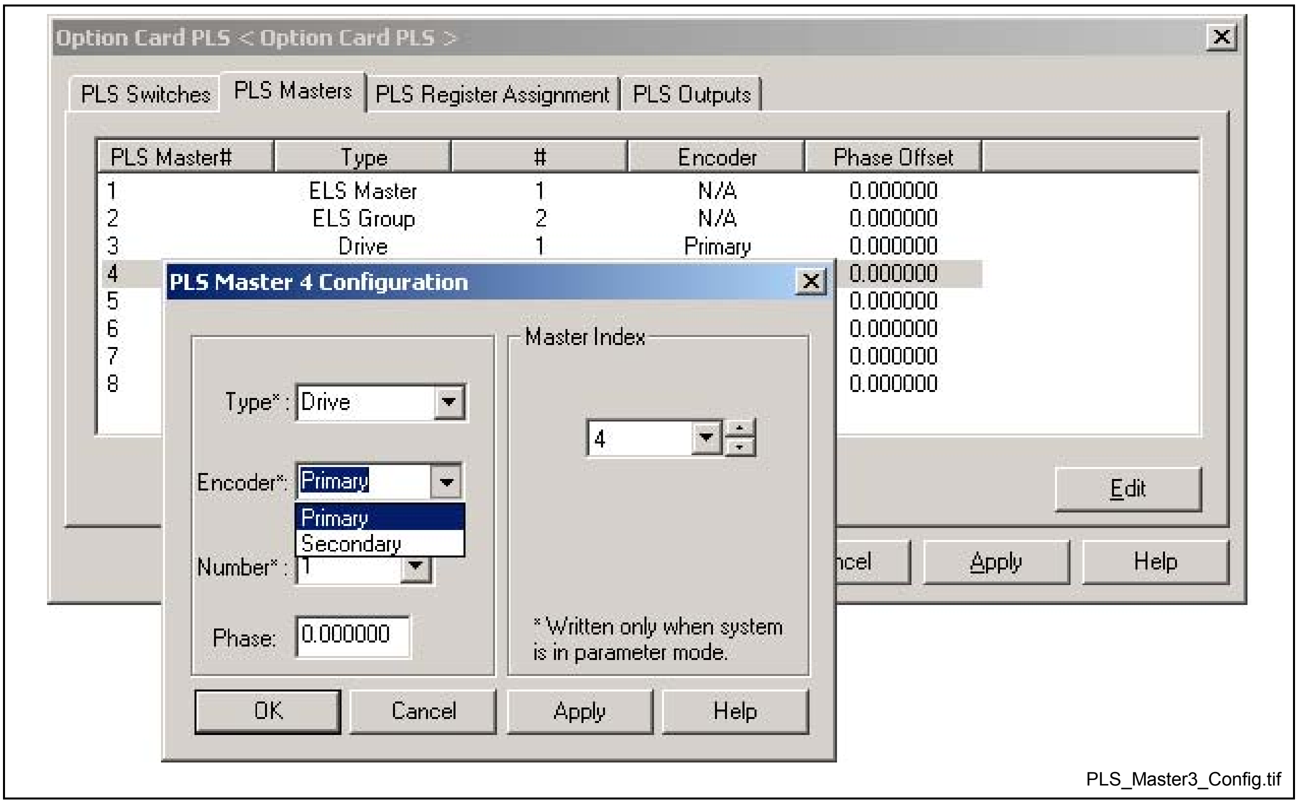


Fig. 5-49: PLS Master Configuration

5. Double-click on the desired *PLS Master #* to open the *PLS Master # Configuration* window.
6. Select a *Master Index* number.

Note: Once a PLS Master is configured and the ***Apply*** button is pressed, a new *Master Index* number must be selected to configure a different PLS Master. Pressing the ***OK*** button accepts the values and closes the window. The current PLS Master number is displayed in the header portion of the window.

Example:

PLS Master **1** Configuration

7. Select a *Type* as either ELS Master, ELS Group, or Drive.

Note: **ELS Master**

An ELS Master can be a Virtual Master, the output of an ELS Group, a Real Master or the output of a second PPC-R cross-communicating in a Link Ring.

ELS Group

If a *Phase* value is added to the PLS Master Configuration window, then the value is added to the output of the ELS Group and does not affect any phase offsets that might have been configured in the ELS Group.

8. Select a Number from the drop down list corresponding to the input type selected.

Note: When a Drive is selected, the allowable number range is 1-64 for up to 64 axes. Then the user must specify the *Encoder* for each axis as either the primary feedback or a secondary feedback device.

9. Enter a Phase offset if applicable. The *Phase* field is an offset that is added to the output of the PLS master.
10. When all PLS Masters are configured, press the ***OK*** button to close the *PLS Master Configuration* window and return to the *PLS Masters* tab.

Note: VisualMotion issues an error when a different Switch Index number is selected without first applying changes to an already configured switch.

PLS Register Assignment for an Option Card PLS

The PLS Register Assignment tab is used to assign a VisualMotion register to both Option Card PLSs output modules (NSW01.1R) installed in the control. The user can also set the range of motion of the graphical representation for the PLS switches.

1. Accept the default register or set a different *Start Output Register*. The *End Output Register* is set automatically to the next consecutive register.
2. Accept the default register or set a different *Start Mask Register*. The *End Mask Register* is set automatically to the next consecutive register.

Note: During the initial setup, the Mask register bits cannot be set to (1). After the PLS is configured and downloaded to the control, the user must manually set the state of each Mask Register bit to (1) by selecting ***Data*** ⇒ ***Registers*** from VisualMotion's main menu. Only the Output Register whose Mask Register is set to (1) will output a signal on the Option Card PLS.

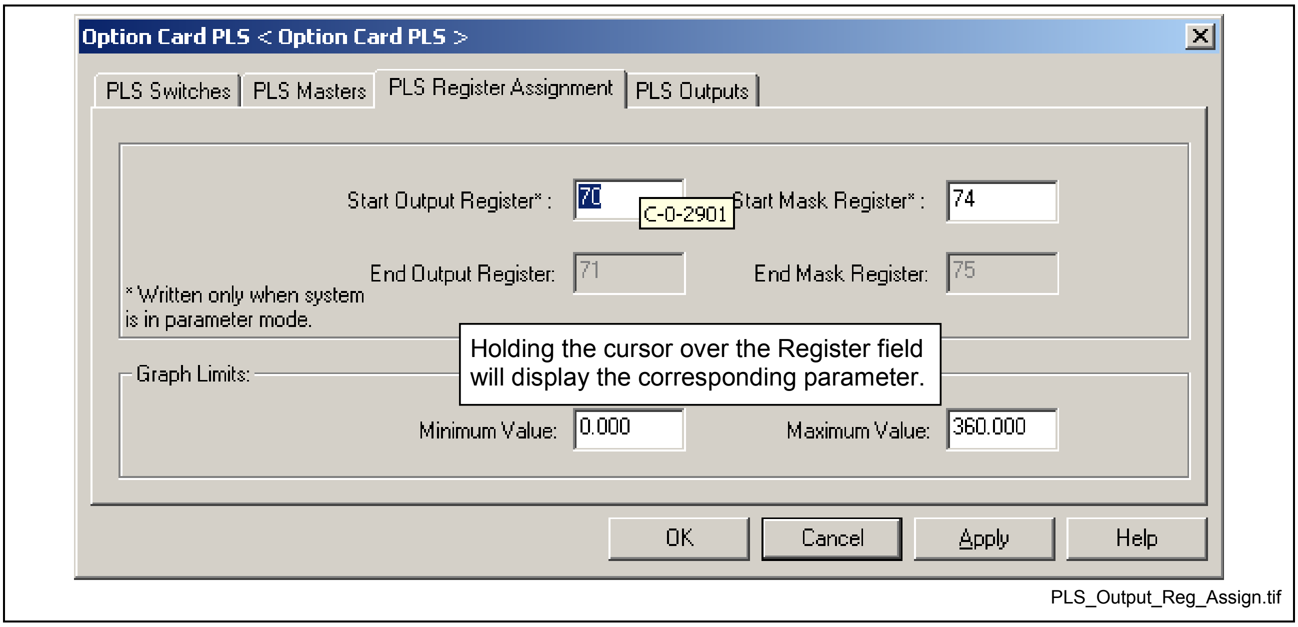


Fig. 5-50: PLS Register Assignment

Output Registers

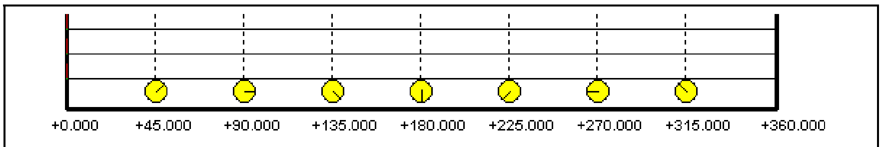
The Start and End registers are assigned to the PLS outputs for monitoring the status of each output. The Start Output Register is assigned to outputs 01-16 on the first NSW01.1R installed in slot U2, where the outputs match the bit numbers respectively. The End Output Register is automatically set to the next consecutive register and assigned to outputs 17-32 of the second NSW01.1R whether or not it is installed in slot U3.

Mask Registers

The Start and End Mask registers are used to force the state of the output registers. On power-up, the bits in the mask registers are set to 0 to prevent any unwanted outputs from enabling. The PLS switches assigned to a start output register will not function unless the state of the register's complementary mask register bits are set to 1.

Note: The default state of the Start and End Registers is zero (0). The user must set the corresponding mask register each time after power-up.

3. Set the *Graphs Limit*'s minimum and maximum value.



PLS_Graph.tif

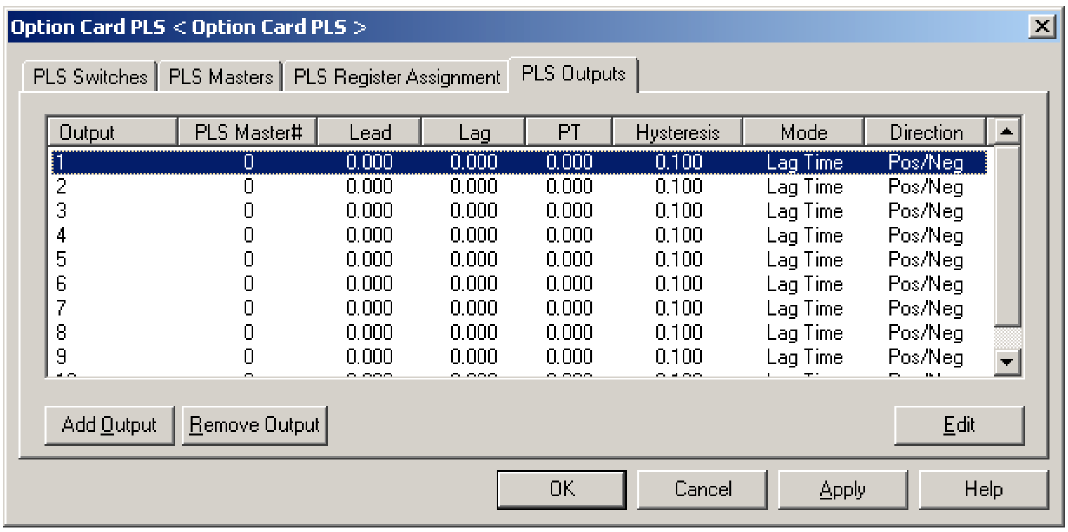
Fig. 5-51: Graph Limits

4. Click on the *PLS Outputs* tab to configure the PLS outputs.

Option Card PLS Outputs

The final process in the configuration of an Option Card PLS is to assign a PLS Master to each output. From the window in Fig. 5-52, the user can scroll through the overall configuration of all 32 outputs.

Note: Although 32 outputs can be displayed and can be configured, only the first 16 outputs will have a physical output for a PPC-R configurations having only one NSW01.1R installed.



PLS_Output_Assign.tif

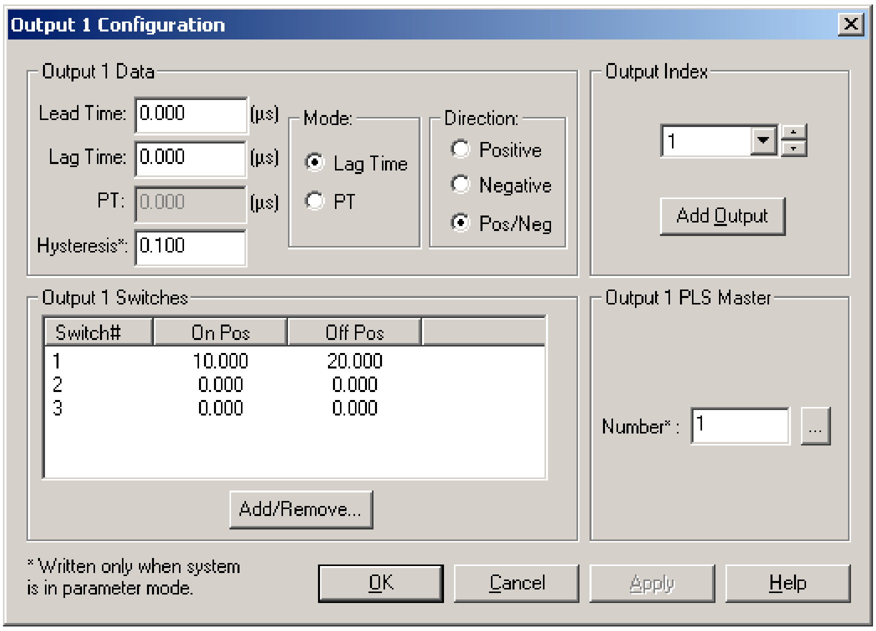
Fig. 5-52: Output Assignment

1. Click on the **Add Output** or **Remove Output** button to add or remove outputs.

2. Double-click on an output or highlight the output and press the ***Edit*** button to open the *Output # Configuration* window.
3. After all the outputs are configured, press the ***Finish*** button.

Output Configuration

The *Output Configuration* window is used to fine tune each PLS output. From the *Output # Data* section, the user can make timing adjustments and set the direction in which the switch is recognized. The *Output Index* section is used to navigate up to 32 outputs. From the *Output # Switches* section, the user can view the current switch(es) configured to an output, add or remove any switches or edit a switch's On/Off position. The *Output # PLS Master* section is used to assign a PLS Master to the output.



PLS_Output_Config.tif

Fig. 5-53: Output Configuration Window

Holding the cursor over the following fields will display the corresponding parameter to which the configured value is stored.

Field Name	Parameter	Condition
Lead Time	C-0-2931	Always active
Lag Time	C-0-2932	Active when Mode = Lag Time
PT (Time Duration)	C-0-2933	Active when Mode = PT Mode
Hysteresis	C-0-2936	Always active
Mode	C-0-2934	Always active
Direction	C-0-2935	Always active
Number (Output # PLS Master)	C-0-2941	Always active

Table 5-23: Output Configuration Parameters

Output # Data

From the Output Configuration window, the user has the following configuration options:

- ***Lead Time (C-0-2931)*** - The amount of time that the output is enabled prior to reaching the switch's On position. This value is used by the control to calculate a position based on the switch's On position and the current speed of the PLS Master.

- ***Lag Time (C-0-2932)*** - The amount of time that the output is disabled prior to reaching the switch's Off position. This value is used by the control to calculate a position based on the switch's Off position and the current speed of the PLS Master. This option is only available when the *Mode* is set to Lag Time.

The time entered for both Lead and Lag Time is in increments of 250µs, up to a maximum of 500,000µs.

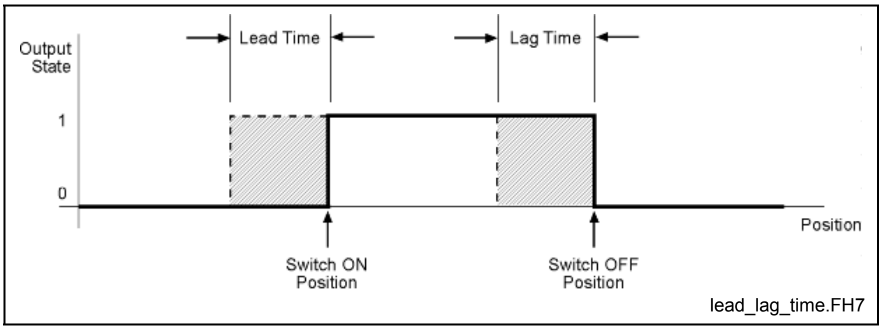


Fig. 5-54: Lead and Lag Time

- ***PT (Time Duration) (C-0-2933)*** - The amount of time that the output is maintained enabled every time the switch's On position (rising edge) is reached. The time is entered in increments of 250µs, up to a maximum of 1,000,000 µs.

  When the Output Mode is set to PT and a value of zero is used, the switch's on position will be ignored. If after a rising edge is encountered and before the falling edge is reached a change of direction occurs and a PT time is set, the timer will run for the set time duration. This time duration can be reset and is only triggered with a rising edge.

Note: If a non-incremental time is entered, VisualMotion will issue an error and set the time to the next closest 250µs increment.

- ***Hysteresis (C-0-2936)*** - This value can be either *positive or negative* and sets the tolerance distance used by the control to determine whether or not a switch is in position. This is used to prevent dithering of the output. The value entered for the hysteresis will affect all switches in the *Output Switches* section. A PLS Master that is either a Real Master or a Drive uses a feedback device to output a positional value. Due to small positional value corrections from the drive trying to hold or maintain a position, the feedback output data will experience small positional changes. The illustration in Fig. 5-55 shows the different reactions for positive and negative hysteresis values.

Note: ***Positive Hysteresis***

In the positive direction, the switch's On and Off positions enables and disables the output.

In the negative direction, the switch's Off and On positions minus the hysteresis value enables and disables the output.
Negative Hysteresis

In the positive direction, the switch's On and Off positions plus the hysteresis value enables and disables the output.

In the negative direction, the switch's Off and On positions enables and disables the output.

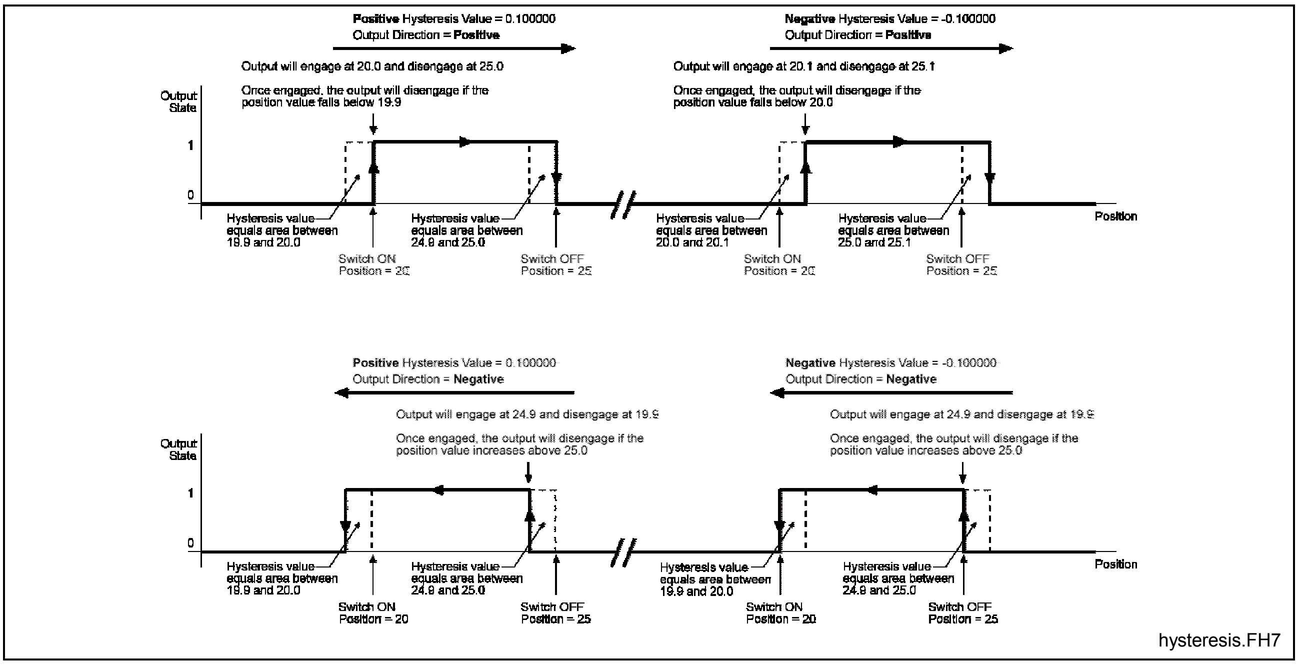


Fig. 5-55: Positive and Negative Hysteresis Value Reaction

Mode (C-0-2934)

The mode radio buttons enable options for Lag Time and PT.

Lag Time When selected, the Lead Time, Lag Time and Hysteresis fields are available for data entry.

PT When selected, the Lead Time, PT and Hysteresis fields are available for data entry.

Direction (C-0-2935)

This option sets the direction of the switches configured for the output in the Output # Switches section.

Positive When set to positive direction, all switches associated with the output will enable the output ***only*** in the positive direction with the On switch position and disable the output with the Off switch position.

Negative When set to negative direction, all switches associated with the output will enable the output ***only*** in the negative direction with the Off switch position and disable the output with the On switch position.

Note: Once the On position is reached for either positive or negative direction, any change in direction (while the output is enable) will immediately disable the output.

Positive/Negative This setting is a combination of the positive and negative direction settings. The outputs will react as described for both the positive and negative direction settings.

Output Index The *Output Index* section is used to navigate between all 32 outputs.

VisualMotion issues the following error within the *Output # Configuration* window when a different Output Index number is selected without first applying any changes to an already configured Output.

Output 1 PLS Master (C-0-2941) The Number field is used to specify the PLS Master that will be used by the system to provide positional data to the output.

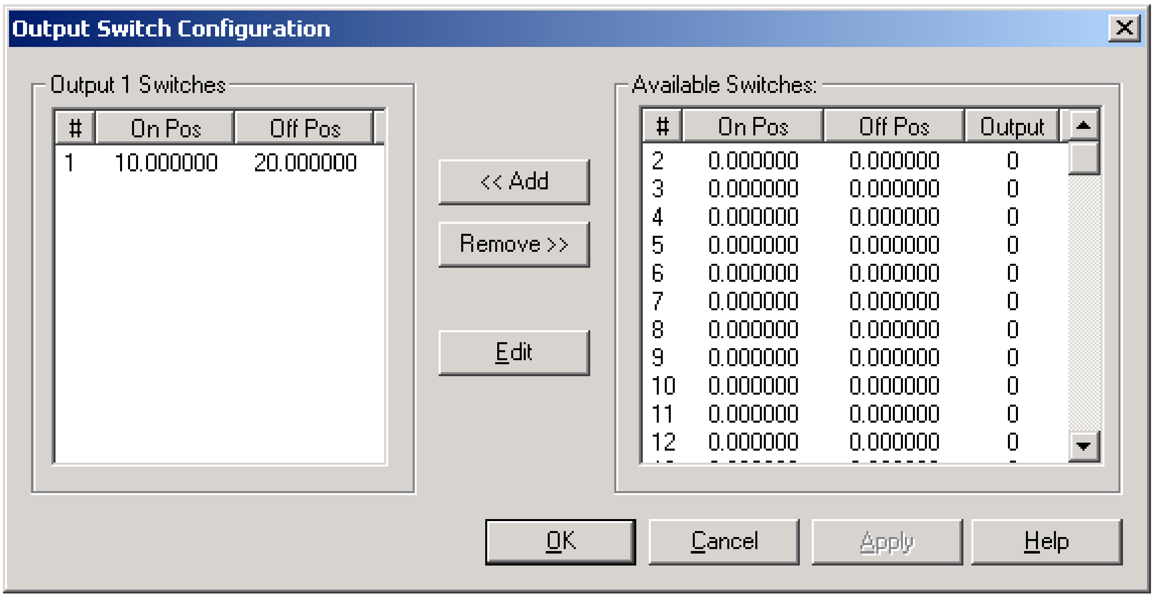
The ellipsoid button to the right of the Number field can be used to select, modify an existing master or configure a new master. Every configured output must have a PLS Master assigned to function.

Note: The PLS Output Master Number field can only be written to while the system is in Sercos phase 2.

Output 1 Switches

The *Output # Switches* section displays all the configured PLS switches that were assigned to the output.

In addition to these switches, the user can double-click on a switch or press the **Add/Remove** button to open the *Output Switch Configuration* window.



Output_Switch_Edit.tif

Fig. 5-56: PLS Output Switch Configuration

Switches can be added or removed using the ***<< Add*** and ***Remove >>*** buttons. between *Available Switches* and *Output # Switches* by first selecting the switch. A switch's On/Off position can be edited in either the *Available Switches* or *Output # Switches* locations.

Note: If a switch is removed from the *Output # Switches*, the On and Off positions will be maintained but the output will be set to 0. The switch will appear in *Available Switches* with the output number set to 0.

Double-clicking a switch opens the *Switch # Configuration* window. Refer to section, "Switch Configuration" for information on configuring a Switch.

Note: The Output field can not be modified from this window. To change the output of a switch, ***<< Add*** it to the desired output in the *Output # Switch* section.

VisualMotion Message for Output Switch Configuration

When adding a switch from the *Available Switches* (assigned to a different output than the output number in the Output # Switches, the following VisualMotion Message will appear.

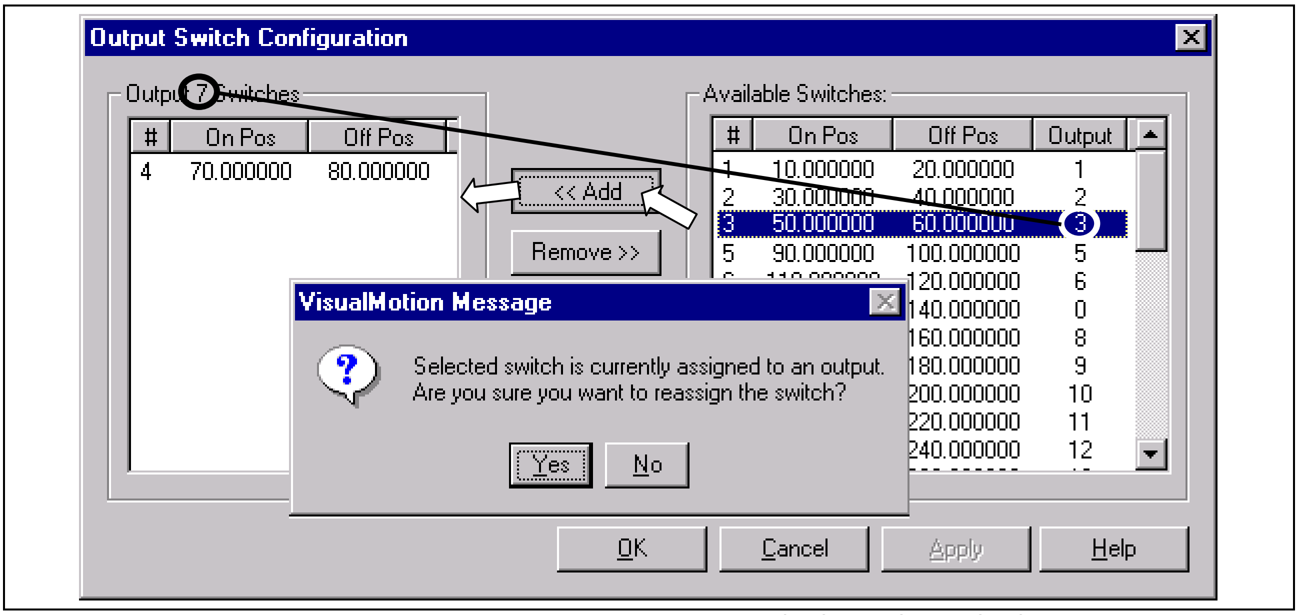


Fig. 5-57: VisualMotion Message for Output Switch Configuration

Note: If a switch is removed from the *Output # Switches*, the On and Off positions will be maintained but the output will be set to 0. The switch will appear in *Available Switches* with the output number set to 0.

Editing PLS Configurations

PLS configurations can be edited in project mode or service mode. Edits to a configuration are synchronized with the control only when they are made in online project mode and the control is in the correct operating phase.

Project Mode

A PLS configuration can be edited in online and offline mode. Data modified in offline mode is read and saved to and from the project's offline data. It is synchronized with control data by selecting the *Online* toolbar button or by selecting ***File*** ⇒ ***Online***. Modifying a PLS configuration that was created as part of a VisualMotion 11 project in offline project mode, ensures the PLS configuration will be saved in the correct format for the projects and the data is synchronized with the project.

Online Editing

PLS data edited in online mode is read from the control's memory. The synchronization of project data edited and saved in online mode is dependent on whether or not the data can be written to in the control's current operating phase (phase 2 or phase 4). The following figure illustrates how edited data is synchronized with the control data.

Rexroth
Bosch Group

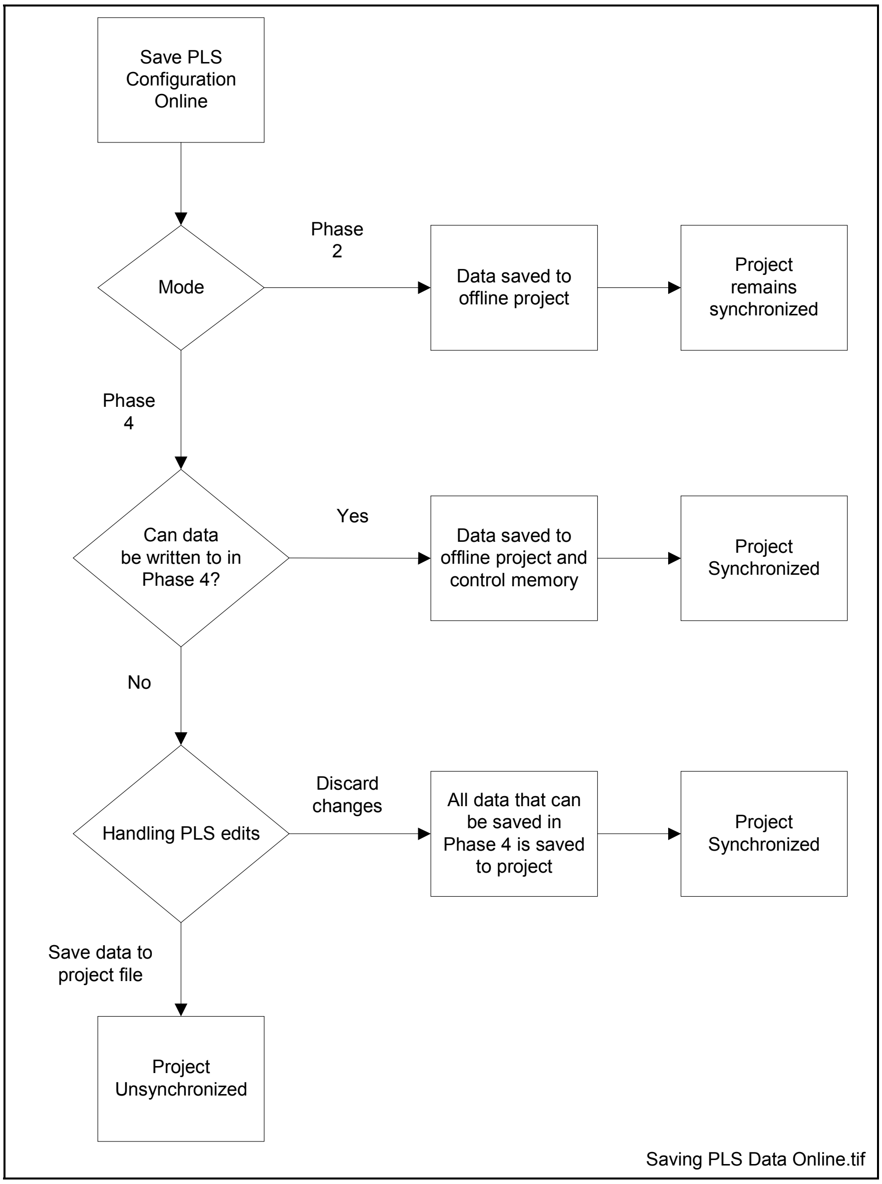


Fig. 5-58: Saving PLS Configuration Data Online

To edit a project's PLS configuration, open the project "*.vmj" file and select ***Commission*** ⇒ ***PLS***. Any Control, Drive or Option Card PLS saved to the offline data or downloaded to the control will be displayed in the PLS tool.

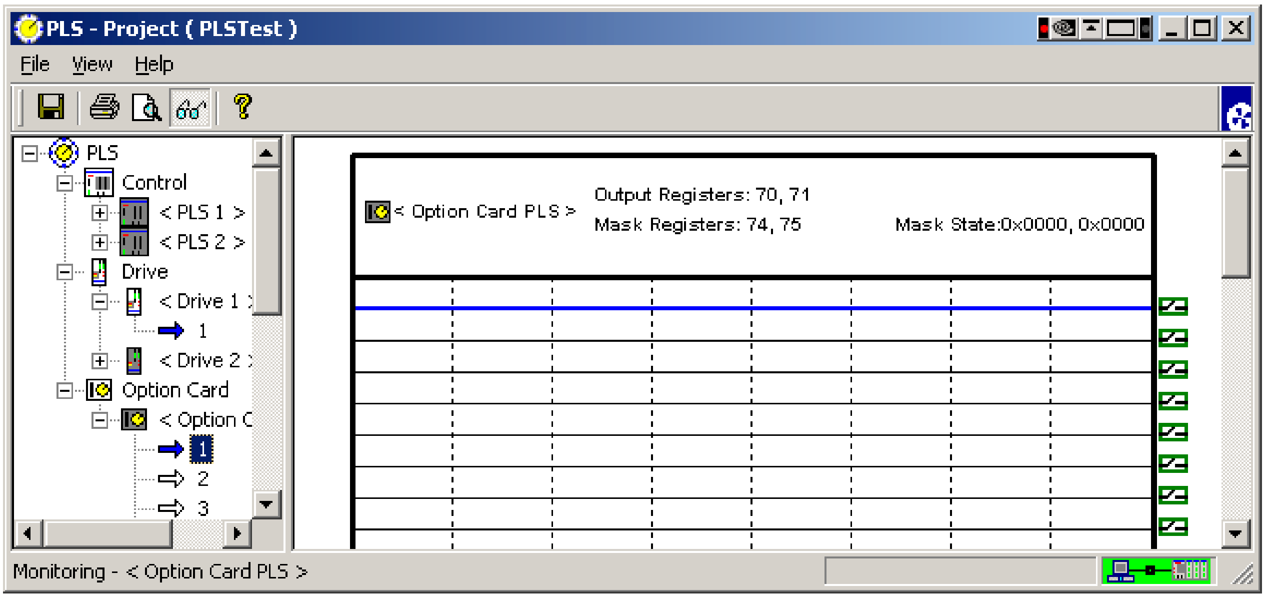

PLS_Upload.tif

Fig. 5-59: Uploaded PLS Configurations

Service Mode

Modifications to a PLS configuration stored in the control's memory can be performed when VisualMotion is in service mode. PLS configurations downloaded to the control or saved to a file in service mode are not synchronized with a project's offline data. Modifications must be imported into the VisualMotion project to be synchronized with offline data.

The following steps outline the procedure for uploading and downloading an existing PLS configuration from and to the control for modifications.

Note: Communication with the control must be established before proceeding.

1. Start VisualMotion and select *View and edit control data in "Service" mode*.
2. Select ***Commission*** ⇒ ***PLS*** to open the PLS Tool.

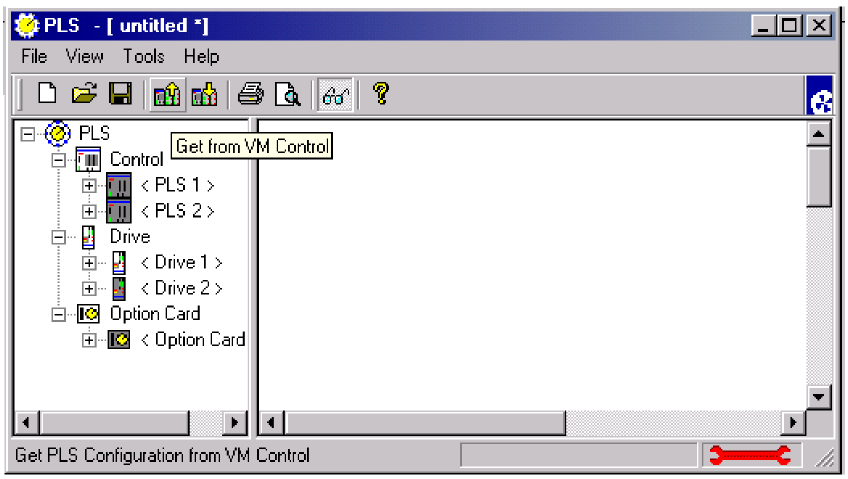

PLS_Tool_Service.tif

Fig. 5-60: PLS Tool in Service Mode

3. Switch the control to parameter mode.

4. Upload the PLS configuration by selecting ***File* ⇒ *Get PLS Configuration from GPP Control*** or click the Get from VM control icon ( ).
5. Make the necessary modifications to the PLS configuration and save the file.
6. Download the modifications to the control by clicking Send to VM control icon ( ).

Saving PLS Configurations

PLS configurations parameterized in offline or online project modes can be saved to offline data or synchronized with the control. The method of saving a PLS configuration is dependent on what mode the project is in. In project mode, all configured PLS types (Control, Drive and Option Card) can be saved to a project folder in offline mode or synchronized with the control if edited and saved while online. The PLS configuration is saved either by selecting the Save icon or by switching VisualMotion Toolkit to online mode.

In service mode, Drive and Option Card PLSs can be saved to a file (**.prm* extension) on the PC's hard drive. However, a Control PLS is saved only with the user program and not as a file on a PC's hard drive. PLS configurations can be saved to the hard drive by selecting ***File* ⇒ *Save As…*** or by pressing the Save icon.

Note: Individual PLS configurations can be saved only when configured using ***File* ⇒ *New*** and saving a single configuration type.

Downloading PLS Configurations

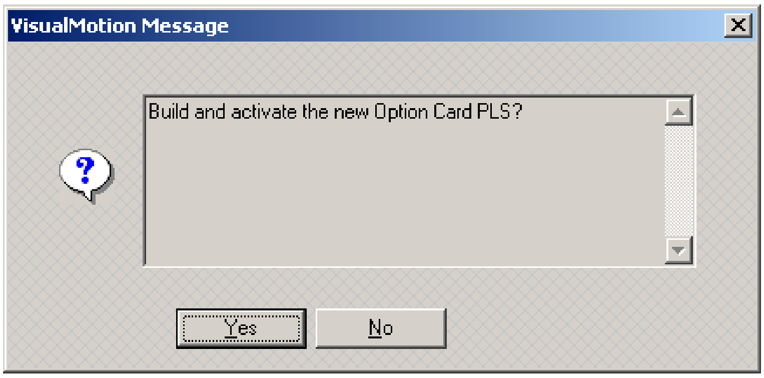
PLS configurations downloaded in project mode are synchronized with the project in which they were created. Service mode is intended for modifying PLS configurations when the project data is not available. After a PLS configuration is downloaded to the control, the assigned PLS mask register must be set before the actual PLS output can fire.

In project mode, a PLS configuration is downloaded when switching to online mode. The system must be in parameter mode to download PLS parameters to the control. Any parameter that can only be downloaded in parameter mode will be indicated by a VisualMotion error. After you are prompted to save any modified data, the *Synchronize Project Data* window is displayed. This window allows you to selectively download PLS types to the control and drives. A configured Option Card PLS downloaded to the control is automatically built and activated before the VisualMotion switches to online mode.

Downloading in Service Mode

In service mode, PLS configurations are downloaded to the control and drives by selecting ***File* ⇒ *Send PLS Configuration to GPP Control*** or by pressing the download icon ( ).

After a successful download, you have the option to build and activate the new Option Card PLS if present in the system. Otherwise, any configured Control or Drive PLS is activated automatically.



PLS_Build.tif

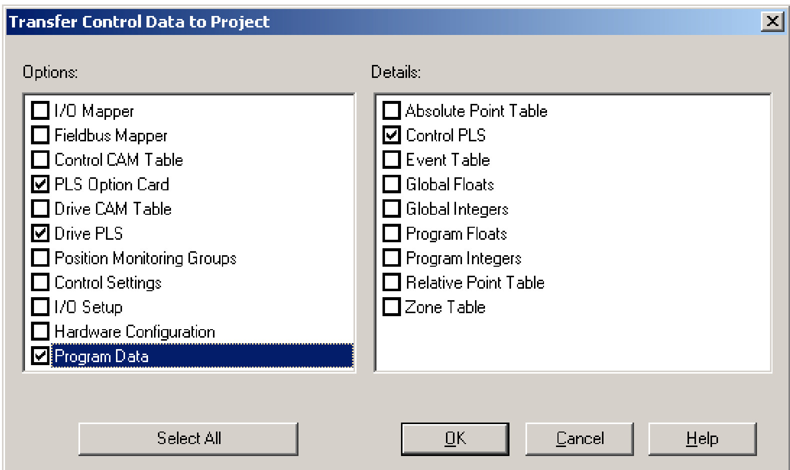
Fig. 5-61: PLS Build New Option Card Message

The system must be in parameter mode to download all the elements of every configured PLS. If the system is not in parameter mode, VisualMotion will issue an error indicating that certain values will not be saved and offer an option to continue. If you select continue, additional errors will be issued indicating which parameter lists will not be written to.

Importing PLS Configurations

A PLS configuration downloaded to the control can be imported into a project while in online mode, or from another project or file when in offline mode. To import a PLS configuration from data stored on the control:

1. Start VisualMotion and open the target project for importing the PLS configuration.
2. Switch VisualMotion to online mode.
3. Select ***File ⇒ Import Project Component*** from VisualMotion Toolkit's main menu.
4. From the *Transfer Control Data to Project* window, select from the following PLS configurations:
   - Option Card PLS
   - Drive PLS
   - Control PLS (located under the Program Data checkbox)



PLSconfigImport.tif

Fig. 5-62: PLS Configuration Transfer from Control

5. Press the ***OK*** button to transfer the selected PLS configurations to the current project. The PLS configuration data is now synchronized with the current project.

Note: When importing data from another project or file in offline mode, the data is not synchronized with the project until VisualMotion is switched to online mode and the data is downloaded to the control.

Uploading PLS Configurations

PLS configurations can be up loaded in either project or service mode.

Project Mode

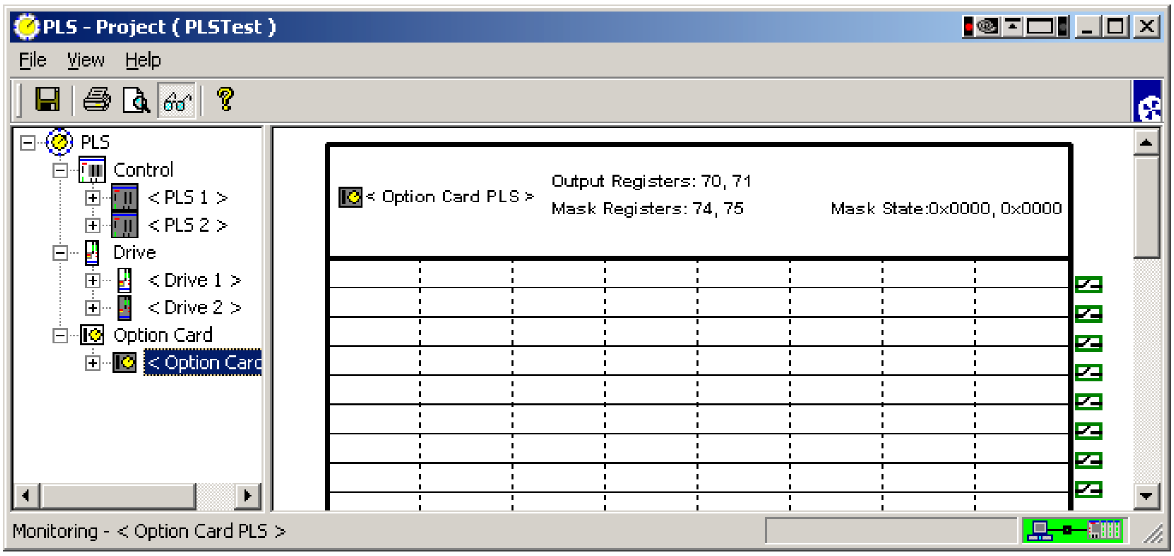
In project mode, a PLS configuration is either uploaded from the control or from the project file, depending on whether the project is online or offline. If a project is offline, the PLS configuration is automatically loaded into the PLS tool from the project file. If a project is online, the PLS configuration is automatically loaded from the control.

Service Mode

Service mode is used to upload PLS configurations from a file or from the control's memory when access to the offline project data is not available. In service mode, a PLS configuration upload is initiated by selecting ***File*** ⇒ ***Get PLS Configuration from GPP Control*** or by selecting the upload icon ( ). PLS configurations can also be uploaded from a file by selecting ***File*** ⇒ ***Open…***. The file extension of a PLS configuration is **.prm* extension. If a PLS configuration is currently open when opening a file, a warning message is displayed. The message indicates that data will be lost and prompts you to verify your action.

Monitoring a PLS Status

The monitoring feature of the PLS tool allows you to monitor the status of the output registers assigned to each PLS type. Monitor mode is activated by selecting the Monitor PLS icon ( ) or by selecting ***View*** ⇒ ***Monitor PLS Status*** in the PLS tool window. PLS configurations can be monitored in project mode while online or service mode.

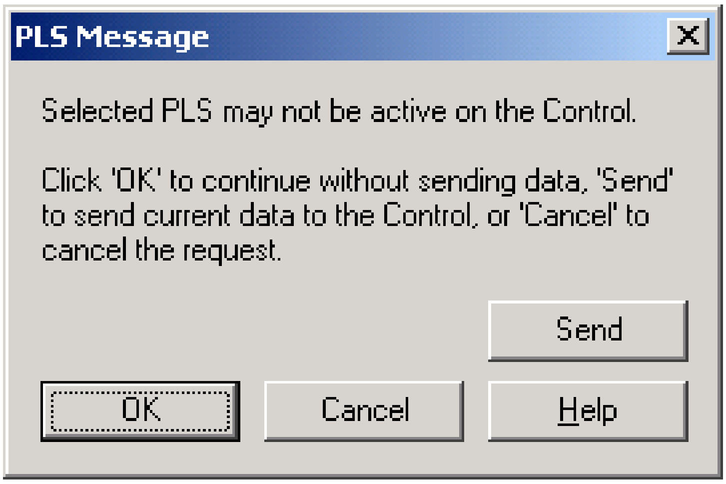


Monitor_PLS.tif

Fig. 5-63: Monitoring a PLS Status

When monitor status is active, a switch icon is displayed to the right of each PLS switch line (see Fig. 5-63). Each switch icon changes from an open to closed switch each time the PLS switch output is active. To monitor a different PLS type, deselect the monitor status icon, select a different PLS type and press the monitor status icon again.

In service mode, the PLS Tool issues the following *PLS Message* window if a PLS configuration is uploaded from the control, modified and downloaded to the control and then the Monitor Status icon () or the F7 key is pressed.



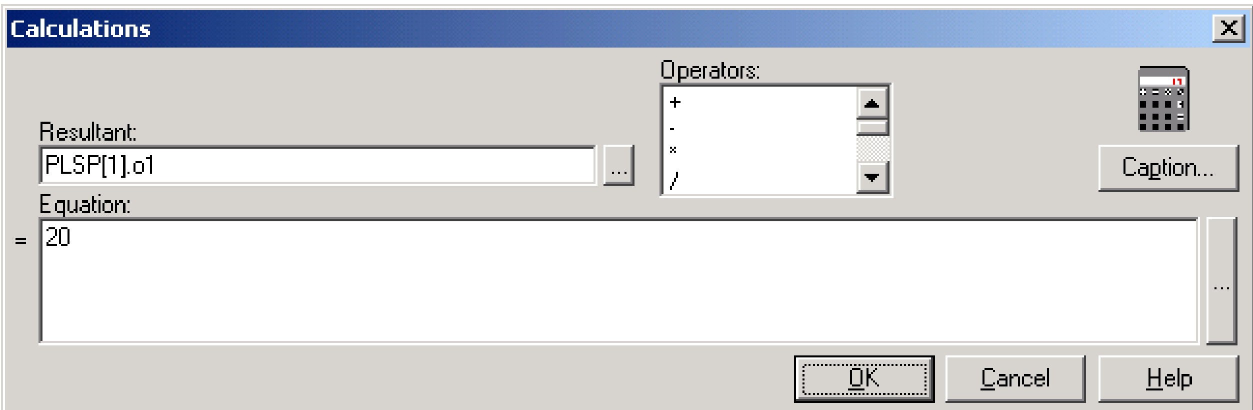
PLS_Message.tif

Fig. 5-64: PLS Message in Service Mode

Access PLS Data via the Calc Icon

After a PLS is configured and downloaded to the control, you can make modifications to the elements of a PLS in the user program using the **Calc** icon. All three PLS types (Control, Drive and Card) can be accessed using the Calc icon.

Note: Any modifications to PLS elements will not be retained after a power down.



Access_PLS.tif

Fig. 5-65: Calc Icon Accessing a PLS

Control PLS

The following table lists the elements of a Control PLS that can be modified using the Calc icon.

Syntax	Description	Variable Range	Example	Comment
PLS[***x***].a	Number	1-64 when t=3 or 4, Drive No. 1-6 when t=5, ELS Master 1-8 when t=6, ELS Group	PLS[1].a=2	set drive number of PLS 1 to 2
PLS[***x***].o	Phase Offset		PLS[1].o=20	add an offset of 20 to PLS 1
PLS[***x***].r	Output Register		PLS[1].r=70	set output register of PLS 1 to 70
PLS[***x***].t	PLS Input Type	3 = Drive's primary feedback 4 = Drive's secondary feedback 5 = ELS Master 6 = ELS Group	PLS[1].t=3	set input type of PLS 1 to 3 (drive's primary feedback)
PLS[***x***].on1-on16	On Position		PLS[1].on1=10	switch 1 On position is 10
PLS[***x***].off1-off16	Off Position		PLS[1].off1=20	switch 1 Off position is 20
PLS[***x***].lt1-lt16	Lead Time		PLS[1].lt1=5	switch 1 lead time is 5
x = PLS number 1-2				

Table 5-24: Elements of a Control PLS

Drive PLS

Note: A drive-based PLS is stored in parameter lists on the drive. The elements of this list cannot be modified individually. Modifying one element of the list requires sending the entire list over the Sercos service channel. This could consume considerable Sercos service channel time.

The following table lists elements in the Drive PLS that can be modified using the Calc Icon:

Syntax	Description	Example	Comment	Parameter
PLSD[***x***].r	Output Register	PLSD[1].r=72	set output register of drive 1 to 72	A-0-0009
PLSD[***x***].t	PLS Input Type	PLSD[1].t=1	set input type of drive 1 to 1 **Diax 04 & EcoDrive 03:** 1 (S-0-0051) 2 (S-0-0053) **IndraDrive using MPx03 and MPx04:** 51 (S-0-0051) 53 (S-0-0053) 32820 (P-0-0052) 33202 (P-0-0434) 33543 (P-0-0775) 33544 (P-0-0776) 33546 (P-0-0778)	P-0-0131
PLSD[***x***].on1-on16	On Position	PLSD[1].on1=10	switch 1 On position for drive 1 is 10	P-0-0132
PLSD[***x***].off1-off16	Off Position	PLSD[1].off1=20	switch 1 Off position for drive 1 is 20	P-0-0133
PLSD[***x***].lt1-lt16	Lead Time	PLSD[1].lt1=5	switch 1 lead time for drive 1 is 5	P-0-0134
x = drive number range 1- 64				

Table 5-25: Elements of a Drive PLS

Option Card PLS

A project can have only one active Option Card PLS. The following tables list the elements of an Option Card PLS.

Master Elements

The following Master elements can be modified through the Calc icon.

Note: Only 1 PLS master is available, syntax is always indicated as PLSP[1].

Syntax	Description	Example	Comment	Parameter
PLSP[1].o***y***	Phase Offset	PLSP[1].o1=20	add an offset of 20 to PLS Master 1	C-0-2943
y= Master range 1-8				

Table 5-26: Master Elements of an Option Card PLS

Switch Elements

The following switch elements can be modified through the Calc icon.

Syntax	Description	Example	Comment	Parameter
PLSP[1].on***y***	On Position	PLSP[1].on3=40	switch 3 On position is 40	C-0-2920
PLSP[1].off***y***	Off Position	PLSP[1].on3=60	switch 3 Off position is 60	C-0-2921
PLSP[1].out***y***	Assigned Output	PLSP[1].out3=2	switch 3 is assigned to output 2	C-0-2922
y = switch range 1- 96				

Table 5-27: Switch Elements of an Option Card PLS

Output Elements

The following Output elements can be modified through the Calc icon.

Syntax	Description	Example	Comment	Parameter
PLSP[1].lt***y***	Lead Time	PLSP[1].lt2=500	add a lead time of 500µs to output 2	C-0-2931
PLSP[1].lg***y***	Lag Time	PLSP[1].lg2=500	add a lag time of 500µs to output 2	C-0-2932
PLSP[1].ss***y***	PT (Time Duration)	PLSP[1].ss2=1000	maintain output 2 On for 1000µs	C-0-2933
PLSP[1].h***y***	Hysteresis	PLSP[1].h2=0.1	add a hysteresis of 0.1 to the output 2	C-0-2936
PLSP[1].md***y***	Mode	PLSP[1].md2=0	set mode of output 2 to Lag Time (0=Lag Time, 1=PT)	C-0-2934
PLSP[1].dr***y***	Direction	PLSP[1].dr2=0	set direction of output 2 to positive (0=pos, 1=neg, 2=pos/neg)	C-0-2935
y = output range 1-32				

Table 5-28: Output Elements of an Option Card PLS

6 Operating Modes

6.1 Electronic Line Shafting (ELS)

Electronic Line Shafting (ELS) is a robust synchronization system that precisely controls motion using electronic gearing and cams. The ELS system has the following components that are connected through adaptable interfaces:

- Command Sources
- Signal Router
- Axis Groups
- Individual Axes

The interfaces can switch between the defined signal path to meet the dynamics of the application, see Fig. 6-1.

VisualMotion GPP software and firmware allow more than one active master at a time to support ELS functionality. Electronically synchronized axes can be combined to form ELS Groups. Active masters can control a maximum of eight ELS Groups. Every ELS Group will follow its selected master.

Note: ELS motion functionality can be associated with task A or with the PLC (IndraLogic). The association is configured under *Setup ⇒ Axes*. Any motion associated with an ELS Group within a project will stop if task A or the PLC stop running.

ELS Overview

GPP supports six system masters in any combination up to a maximum of the following types:

- 2 Virtual Masters
- 6 Real Masters (including Master Encoder Card)
- 6 ELS Group Masters
- 6 Link Ring Master IN
- 1 Link Ring Master OUT

Virtual Masters, Real Masters (primary, secondary feedback device, or Master Encoder Card) and ELS Group Masters can be combined and assigned to one of 6 ELS System Masters. The master signal from each active system master is conditioned (e.g., geared or filtered) and made available for controlling groups of axes.

Position commands originate from either a Virtual Master in the control or a Real Master generated by the feedback device on the motor. The Virtual Master or Real Master signal is assigned a number in the ELS System connection interface and can be modified before it is sent to an ELS Group. An output signal from an ELS Group can be sent to another ELS Group after being assigned a new ELS System Master number. An ELS Group can switch to a Local Virtual Master signal, which functions the same as a system Virtual Master when it is used for the ELS Group Master. The example configuration in Fig. 6-1 shows the signal flow through a system with 2 Virtual Masters, 2 Real Masters, 1 ELS Group Master and 4 ELS Groups. Five ELS System Master numbers are assigned in the connection interface.

Rexroth Bosch Group

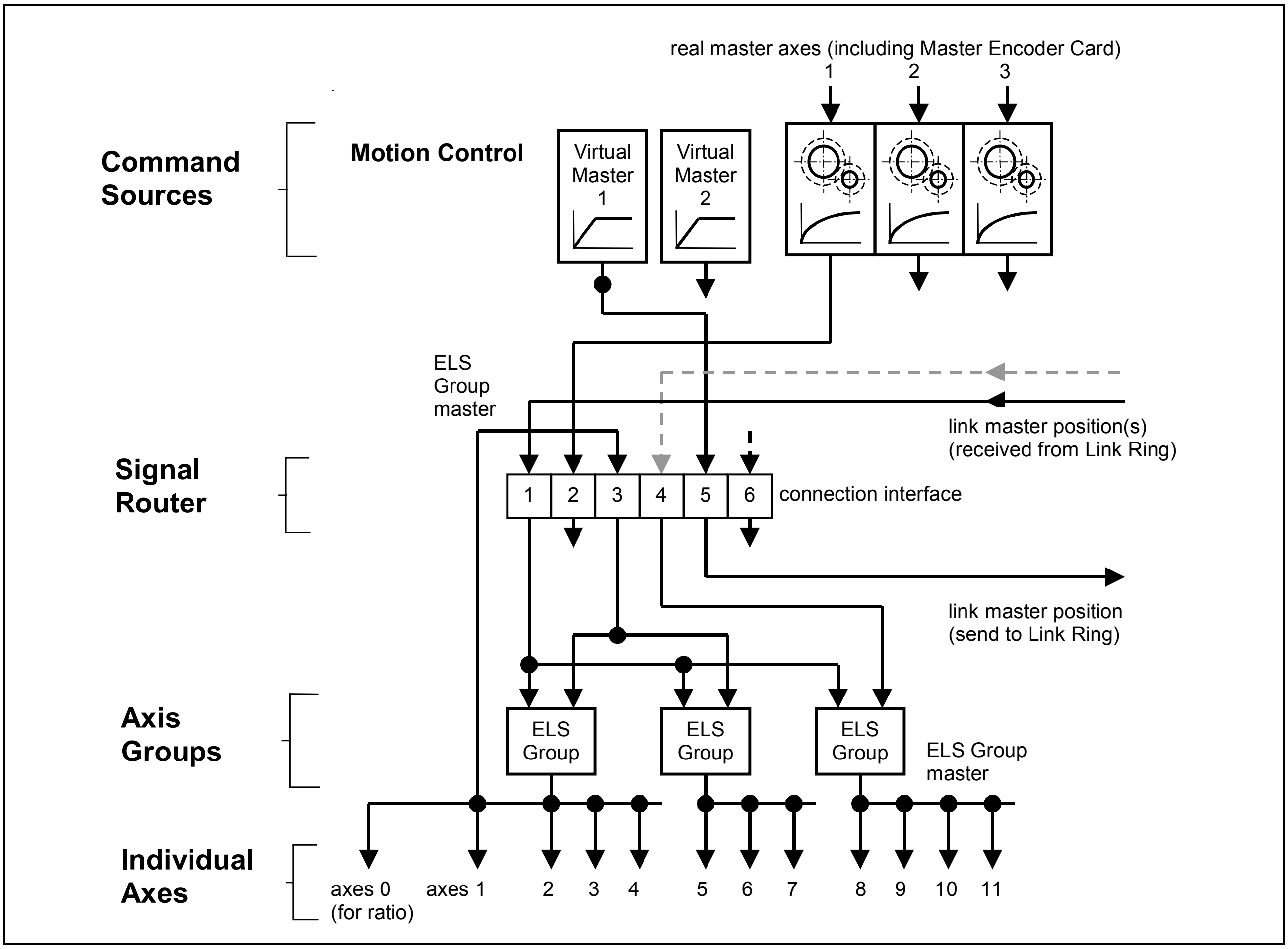


Fig. 6-1: Multiple Master Configuration Example

Multiple master functionality in VisualMotion includes 5 types of masters and ELS Groups that control individual axes:

Virtual Master

A Virtual master is a component that generates a command stream based on its input settings. VisualMotion provides two independent Virtual masters that are used to drive a group of axes or programmable limit switches. A Virtual Master has two primary modes of operations:

- Velocity Mode – continuous running
- Position Mode – indexing, jogging

Group Local Virtual Master

The Group Local Virtual Master can be used to define a set of position values within an ELS Group that the group can be switched to on the fly. This master position is assigned in the ELS Group icon and is accessible only by the ELS Group where it is resident or to another ELS Groups if the position signal is assigned to that group as an ELS Group Master signal. The Group Local Virtual Master signal can be handled in the same way as the system Virtual Master signal.

Real Master

A Real Master is either a primary motor (positive feedback), secondary encoder (Aux) signal from a drive or a Master Encoder Card. Each drive in the system can potentially provide two Real Masters. The raw position value of the Real Master can be filtered and geared by an M/N ratio.

ELS Group Master

An ELS Group Master is the output position of an ELS Group used as an input master signal, geared by an M/N ratio, to a different ELS Group.

Link Ring Master

A link Ring Master is an external PPC control configured as a master, which can have up to 31 PPC control configured as Link Ring Slave controls interfacing with it in a fiber optic ring, called a Link Ring.

Master Encoder Card

The Master Encoder Card (LAG) provides up to two real master EnDat or 1 Vpp Sinusoidal encoder input signals to the ELS System.

ELS Group

An ELS Group is defined as a set of slave axes (64 axes maximum) that follow the position command signal from one of the 6 System Masters. By using ELS Groups, slave axes are combined, according to their function, in groups that control each machine section as an independent process. During operation, an ELS Group can be switched between master signals. Any changes to ELS Group parameters are immediately available to all axes assigned to that group, so that precise synchronization of the machine section is maintained.

Assigning Register and Variable Labels

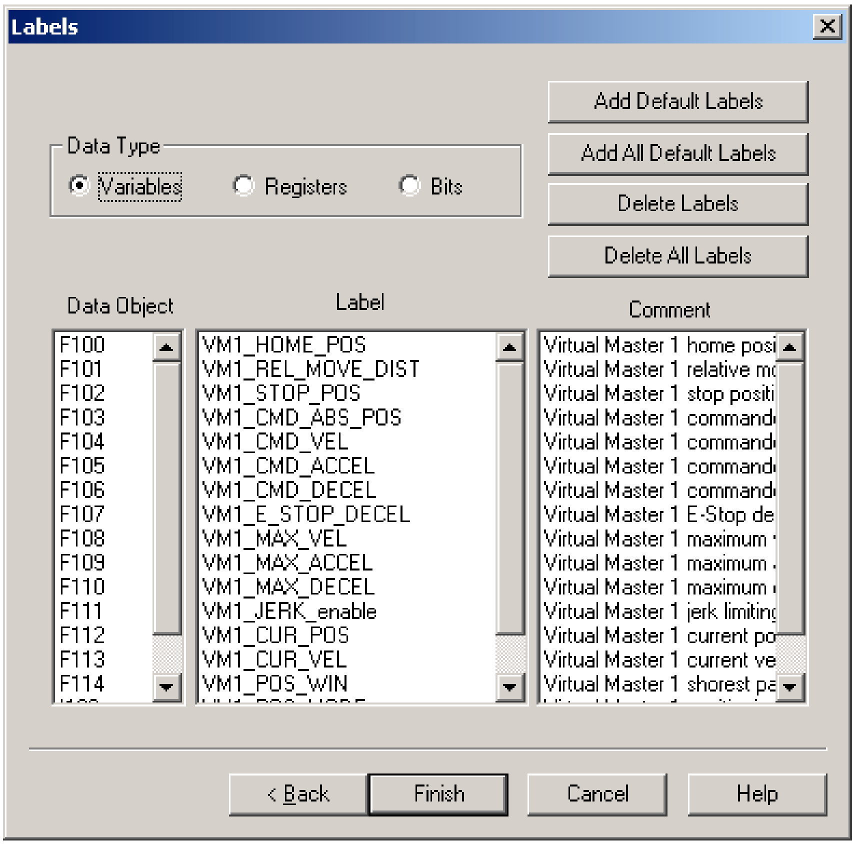
VisualMotion provides 1024 registers for controlling and monitoring a project. During the **Setup** configuration of the Virtual Masters, ELS Groups, and System Masters, their associated control and status registers are assigned. The system has default address ranges defined for the various components, which are automatically indexed as each is created.

Note: Default values are automatically assigned for registers and variable blocks. It is strongly recommended that the programmer use the default values for registers and variables. This makes documentation and modifications to a user program an easier task over the scope of the project.

Default labels and comments are added during the configuration of ELS components under ***Setup ⇒ ELS ⇒ Processes***.

The window has two buttons that allow you to add the default labels for the variables, registers, and bits:

- ***Add Default Labels*** - This button adds all default labels for the data type selected.
- ***Add All Default Labels*** – This button adds all default labels for all three data types.



variable_label.tif

Fig. 6-2: Adding Default Labels during Setup

Note: The *VM Data Table* can be used to add additional labels or modify existing default labels. Refer to section 4.12, *VM Data Table,* for details.

ELS System Default Control and Status Registers

The following tables list the default control and status registers that are assigned by VisualMotion when System Master, Virtual Master and ELS Groups are configured.

ELS System Master	Control Register	Status Register
1-6	140	141

Table 6-1: ELS System Master Default Register

Virtual Master	Control Register	Status Register
1	150	241
2	151	242

Table 6-2: Virtual Master Default Registers

ELS Group	Control Register	Status Register
1	152	243
2	153	244
3	154	245
4	155	246
5	156	247
6	157	248
7	158	249
8	159	250

Table 6-3: ELS Group Default Registers

ELS System Default Program Variables

The following table list the default program variables (floats and integers) used by the ELS System:

Function	Number of Floats	Number of Integers	Float ID Block	Integer ID Block
Virtual Master 1	15	2	F100-F114	I100-I101
Virtual Master 2	15	2	F120-F134	I105-I106
ELS System Masters	48	30	F140-F187	I110-I139
ELS Group 1	40	10	F190-F229	I140-I149
ELS Group 2	40	10	F230-F269	I150-I159
ELS Group 3	40	10	F270-F309	I160-I169
ELS Group 4	40	10	F310-F349	I170-I179
ELS Group 5	40	10	F350-F389	I180-I189
ELS Group 6	40	10	F390-F429	I190-I199
ELS Group 7	40	10	F430-F469	I200-I209
ELS Group 8	40	10	F470-F509	I210-I219

Table 6-4: Program Variable Default Start ID Blocks

Virtual Master 1 & 2 Default Register Labels

The default labels for the Virtual Master registers are listed in Table 6-5. The corresponding default bit labels are listed in Table 6-6.

Data Object Type	Label (20 character limit)	Comment (80 character limit)
Assigned control register number	VM#_CONTROL_REG	Virtual Master # control register
Assigned status register number	VM#_STATUS_REG	Virtual Master # status register

Table 6-5: Virtual Master Default Registers

Default Label Virtual Master 1 & 2 Control Register	Data Object Virtual Master 1 Control Register-Bit	Data Object Virtual Master 2 Control Register-Bit	Comment (80 character limit) Virtual Master 1 & 2 Control Register
VM#_CT_FSTOP	150-1	151-1	VM # control, 0 → 1 triggers fast stop
VM#_CT_HOME	150-2	151-2	VM # control, 0 → 1 loads home position
VM#_CT_GO	150-3	151-3	VM # control, 0=stop, 1=go
VM#_CT_VMODE	150-4	151-4	VM # control, 0=position, 1=velocity mode
VM#_CT_RELMODE	150-5	151-5	VM # control, 0=absolute, 1=relative mode
VM#_CT_RELTRIG	150-6	151-6	VM # control, 0 → 1 triggers relative mode move
Default Label Virtual Master 1 & 2 Status Register	**Data Object Virtual Master 1 Control Register-Bit**	**Data Object Virtual Master 2 Status Register-Bit**	**Comment (80 character limit) Virtual Master 1 & 2 Status Register**
VM#_ST_FSTOP	241-1	242-1	VM # status, 1=fast stop active
VM#_ST_HOME	241-2	242-2	VM # status, 1=home complete
VM#_RESERVE3	241-3	242-3	Reserved
VM#_ST_VMODE	241-4	242-4	VM # status, 1=velocity mode
VM#_ST_RELMODE	241-5	242-5	VM # status, 1=relative mode
VM#_RESERVE6	241-6	242-6	Reserved
VM#_ST_ZEROVEL	241-7	242-7	VM # status, 1=standstill, 0=velocity
VM#_ST_INPOS	241-8	242-8	VM # status, 1=in position
Each # symbol represents an entry for the number of the Virtual Master			

Table 6-6: Virtual Master 1 & 2 Default Register Bits

Virtual Master 1 & 2 Default Program Variable Labels

The default labels for the Virtual Master program variables are listed in the following table:

Default Label Virtual Master 1 & 2 Program Variable	Data Object Virtual Master 1	& 2	Comment (80 character limit) Virtual Master 1 & 2 Program Variable	Default Value	Units	Update Mode
VM#_HOME_POS	F100	F120	Virtual Master # home position	0	Degrees	Phase 4
VM#_REL_MOVE_DIST	F101	F121	Virtual Master # relative move distance	1	Degrees	Phase 4
VM#_STOP_POS	F102	F122	Virtual Master # stop position	0	Degrees	Phase 4
VM#_CMD_ABS_POS	F103	F123	Virtual Master # commanded absolute position	0	Degrees	Phase 4
VM#_CMD_VEL	F104	F124	Virtual Master # commanded velocity	20	RPM	Phase 4
VM#_CMD_ACCEL	F105	F125	Virtual Master # commanded acceleration	100	Rad/sec^2	Phase 4
VM#_CMD_DECEL	F106	F126	Virtual Master # commanded deceleration	100	Rad/sec^2	Phase 4
VM#_E_STOP_DECEL	F107	F127	Virtual Master # E-Stop deceleration	500	Rad/sec^2	Phase 4
VM#_MAX_VEL	F108	F128	Virtual Master # maximum velocity	3200	RPM	Phase 2
VM#_MAX_ACCEL	F109	F129	Virtual Master # maximum acceleration	1000	Rad/sec^2	Phase 2
VM#_MAX_DECEL	F110	F130	Virtual Master # maximum deceleration	1000	Rad/sec^2	Phase 2
VM#_JERK_ENABLE	F111	F131	Virtual Master # jerk limiting enable	1		Phase 2
VM#_CUR_POS	F112	F132	Virtual Master # current position	0	Degrees	Phase 4
VM#_CUR_VEL	F113	F133	Virtual Master # current velocity	0	RPM	Phase 4
VM#_POS_WIN	F114	F134	Virtual Master # shortest path window	1	Degrees	Phase 2
VM#_POS_MODE	I100	I105	Virtual Master # positioning mode [1.)]	0		Phase 2
VM#_RESERVE_I1	I101	I106	Reserved			
Each # symbol represents an entry for the number of the Virtual Master Note 1.) Absolute Position Mode, 0=Positive, 1= Negative, 2= Shortest Path						

Table 6-7: Virtual Master 1 & 2 Default Program Variables

ELS System Master Default Registers

The default labels for the ELS System Master registers are listed in Table 6-5. The corresponding default bit labels are listed in Table 6-9.

Data Object Type	Label (20 character limit)	Comment (80 character limit)
Control Register 140 (Default)	ELS_MSTR_CONTROL	Control Register for ELS Masters
Status Register 141 (Default)	ELS_MSTR_STATUS	Status Register for ELS Masters

Table 6-8: Default ELS Master Registers

Default Label ELS Master 1 to 6	Data Object ELS Master Register-Bit	Comment (80 character limit) ELS Master Status Register
ELS_M_CT_RESERVE1-6	140-1 to 140-6	
ELS_M_CT_SET_REF1 - 6	140-7 to 140-12	0 -> 1 Sets ELS Master 1 reference position for real master (Phase 4 only)
ELS_M_CT_RESERVE13- 14	140-13 to 140-14	
ELS_MCT_SLIP_CAPT1	140-15	ELS Master Capture Slip Monitoring 0 = No Capture 1 = Capture Run – Time Data
ELS_M_CT_SLIP_EN	140-16	ELS Master Enable Slip Monitoring 0 = Off 1 = On
ELS_M_ST_STOPPED1 - 6	141-1 to 141-6	ELS Master at Standstill
ELS_M_ST_REF1 - 6	141-7 to 141-12	ELS Master Referenced 1 = Referenced 0 = Not referenced(Real master only)
ELS_M_ST_RESERVE13	141-13	Reserved
ELS_M_ST_SLIP_ERR	141-14	ELS Master Slip Monitoring Error 0 = No error 1 = Error
ELS_M_ST_SLIP_ENC	141-15	ELS Master Slip Monitoring Lead Encoder 0 = Primary 1 = Secondary
ELS_M_ST_SLIP_ENA	141-16	ELS Master Slip Monitoring Enabled 0 = Off 1 = On

Table 6-9: ELS System Masters Default Register Bits

ELS System Master Default Program Variable Labels

The default labels for the ELS System Masters program variables are listed in the following table:

ELS System Master Program Variable Default Label	ELS System Master Program Variable 1	 2	 3	 4	 5	 6	ELS System Master Program Variable Default Comment (80 character limit)	Update Mode
ELS_MSTR_FREQ#	F140	F141	F142	F143	F144	F145	ELS Master # filter cutoff frequency	Phase 2
ELS_MSTR_M#	F146	F147	F148	F149	F150	F151	ELS Master # M factor	Phase 2
ELS_MSTR_N#	F152	F153	F154	F155	F156	F157	ELS Master # N factor	Phase 2
ELS_MSTR_SLIP_WINDOW	F158						ELS Master max allowed slip deviation window	Captured on rising edge of capture bit in P4
ELS_MSTR_SLIP_OFFSET	F159						ELS Master position offset for slip monitoring	Phase 2/4
ELS_MSTR_SLIP_VELTHD	F160						ELS Master slip monitoring primary velocity threshold	Phase 4
ELS_MSTR_SLIP_PEAK	F161						ELS Master peak slip deviation	Phase 4 (read-only)
ELS_MSTR_SLIP_ACTUAL	F162						ELS Master current slip deviation (actual)	Phase 4 (read-only)
ELS_MSTR_STANDSTILL	F163						ELS Master Standstill Velocity Threshold	Phase 4
ELS_MSTR_POS#	F164	F165	F166	F167	F168	F169	ELS Master # output position	Phase 4 (read-only)
ELS_MSTR_VEL#	F170	F171	F172	F173	F174	F175	ELS Master # output velocity	Phase 4 (read-only)
ELS_MSTR_OFFSET#	F176	F177	F178	F179	F180	F181	ELS Master # real master offset	Phase 4
ELS_MSTR_REF_POS#	F182	F183	F184	F185	F186	F187	ELS Master # real master reference position	Phase 4
ELS_MSTR_A#	I110	I111	I112	I113	I114	I115	ELS Master # ID number	Phase 2
ELS_MSTR_EC#	I116	I117	I118	I119	I120	I121	ELS Master # encoder, Real Master only	Phase 2
ELS_MSTR_FLTR#	I122	I123	I124	I125	I126	I127	ELS Master # filter	Phase 2
ELS_MSTR_TYPE#	I128	I129	I130	I131	I132	I133	ELS Master # type	Phase 2
ELS_MSTR_SLIP_PRI	I134						ELS Master slip primary address	Phase 2
ELS_MSTR_SLIP_SEC	I135						ELS Master slip secondary address	Phase 2
ELS_MSTR_CONFIG	I136						ELS Master slip monitoring settings	Phase 2/4
ELS_MSTR_RSVD1	I137						reserved for ELS Master	Phase 2
ELS_MSTR_RSVD2	I138						reserved for ELS Master	Phase 2
ELS_MSTR_RSVD3	I139						reserved for ELS Master	Phase 2
Each # symbol represents the number of the ELS Master Shaded program variables are read-only (You can overwrite the current value. However, the ELS system will also overwrite the current value if it changes in the system)								

Table 6-10: ELS Master Assignment Default Program Variables

Rexroth
Bosch Group

ELS System Master Configuration Word

For every ELS Master, an ELS Master Configuration Word (ELS_MSTR_CONFIG) is used to configure all settings for velocity rounding, slip offset, and slip error. These settings are initially configured within VisualMotion Toolkit's ELS System Master icon and become active when the project is compiled and downloaded to the control. These can also be modified by accessing the appropriate integer number and entering an equivalent hexadecimal value.

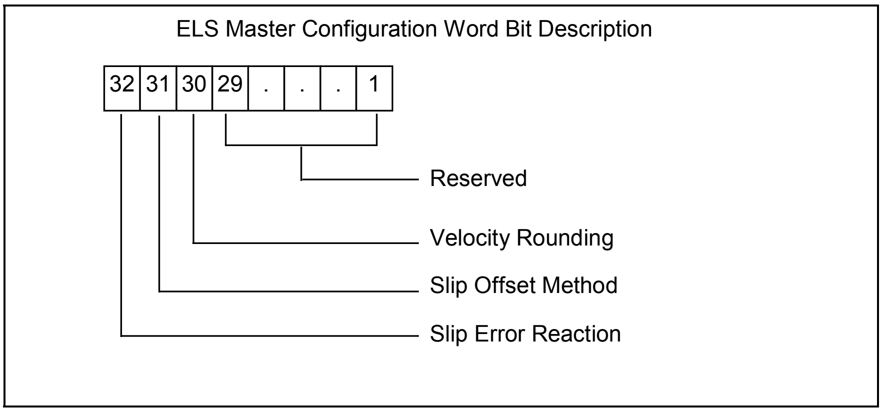


Fig. 6-3: ELS Master Configuration Word

Bit 30
Velocity Rounding

This bit rounds the Virtual Master (VM#_CMD_VEL) and Group Jogging (G#_JOG_VEL) velocity down to the nearest ELS increment in order to eliminate small cycle-to-cycle variations in drive velocity that would otherwise occur.

0 = Disabled

1 = Virtual Master and Group Jogging velocities are rounded-down to the nearest ELS increment

Bits 31
Slip Offset Method

In Absolute Position Monitoring mode this parameter is used as an absolute offset between the two master encoders, this allows the Feedback/Master signals to have an offset between the two without needing to mechanically zero the machine.

0= Fixed Absolute Offset – the value is static and you set an absolute offset between the two master encoders.

1= Dynamic Offset – the system automatically records the position difference between the two master signals when the control switches from phase 2 to phase 4 and on the rising edge of the ELS_MCT_SLIP_CAPT1 bit.

Bit 32
Slip Error Reaction

This bit sets the system reaction to the master encoder signal exceeding the maximum allowed deviation window limits.

0= Fatal Error (system stops)

1= Warning (user defined)

ELS System Master Variable Definition

ELS_MSTR_FREQ#

Only Real Masters use the filter constant. When a filter (*ELS_MSTR_FLTR#*) is selected for an axis' position feedback, a cutoff frequency for the filter must be entered. The cutoff frequency is the frequency where the signal is reduced by 3dB.

ELS_MSTR_M#
and ELS_MSTR_N#

Only Real Masters use the ratio constants (M/N). The output of the master is governed by the equation $y=(M/N)*x$, where $\boldsymbol{x}$ is the feedback value from the real master and $\boldsymbol{y}$ is the master signal used for ELS Groups. All ELS Masters and ELS Groups outputs are modulo 360 degrees.

ELS_MSTR_A#

This variable identifies a valid ID number for a defined master type. For example, when *ELS_MSTR_TYPE#* is set to 3 (Virtual Master) this number must be a 1 or 2. Valid ID numbers are…

- Virtual Master: 1 - 2
- ELS Group Master: 1 - 8
- Real Master: 1 - 3
- Master Encoder Card: 1 - 2

ELS_MSTR_EC#

When using an encoder device as a Real Master, this variable identifies the source.

- 0 = motor encoder
- 1 = external encoder

ELS_MSTR_FLTR#

This variable identifies the type of filtering to use for the axis position feedback. Valid types are…

No Dead Time Compensation	Velocity Feed Forward w/ Dead Time Compensation Enabled
20 = no filter	0 = no filter
21 = 1st order low pass	1 = 1st order low pass
22 = 2nd order low pass	2 = 2nd order low pass
23 = 3rd order low pass	3 = 3rd order low pass
24 = 2nd order Butterworth	4 = 2nd order Butterworth
25 = 3rd order Butterworth	5 = 3rd order Butterworth
26 = 2nd order low pass with velocity feed forward	6 = 2nd order low pass with velocity feed forward
27 = 3rd order low pass with velocity and acceleration feed forward	7 = 3rd order low pass with velocity and acceleration feed forward

Table 6-11: ELS_MSTR_FLTR# Values

ELS_MSTR_TYPE#

Available master types are…

- 0 = Real Master
- 1 = ELS Group master
- 2 = Link Ring Master
- 3 = Virtual Master
- 4 = None
- 5 = Master Encoder Card

ELS_MSTR_SLIP_WINDOW

In absolute position monitoring mode, this sets the maximum allowable position difference between the two selected master signals.

- Max Master velocity @ 2ms Sercos = 7500 RPM
- Max Master velocity @ 4 ms Sercos = 3750 RPM
- Max Master velocity @ 8 ms Sercos = 1875 RPM (Link Ring)

ELS_MSTR_SLIP_OFFSET

In absolute position monitoring mode, this variable is used to offset two master encoders. This eliminates the need to mechanically zero the machine.

Range limits: $-180° < X \leq 180°$

ELS_MSTR_SLIP_VELTHD This is the velocity threshold that the primary master encoder must exceed before the system is able to detect the direction of the primary encoder.

ELS_MSTR_SLIP_PEAK In absolute positioning mode, this is used to store the peak slip position deviation between the primary and secondary master signals. Every Sercos cycle, the system compares the value stored in this variable to the current slip deviation. If this value is higher than any of the previous values, it will be set as the new peak deviation position. If you make a change to your program that lowers your peak value, you can write a zero to the variable to force the program to overwrite it with the new peak value.

ELS_MSTR_SLIP_ACTUAL In absolute position mode, this variable displays the current position difference between the two selected master signals.

ELS_MSTR_STANDSTILL This variable is the velocity threshold that each master output must exceed before the system sets the master status standstill bit to zero for each given master (ELS_MSTR_STATUS, bits 1 to 6). This bit will only go high if the associated ELS System Master has been at or below the threshold velocity for the last two Sercos cycles.

ELS_MSTR_POS# This variable displays the current output position in degrees for each of the six master signals. This variable is stored in the control's memory and when the control is turned on and off, the last recorded value is maintained. You can enter a value in the variable, but it will be overwritten by the current value in the control.

ELS_MSTR_VEL# This variable displays the current output velocity for each of the six master signals.

ELS_MSTR_SLIP_PRI This variable sets the address of the primary master signal to be monitored.

Limits: 0 to 6 (0 not configured)

ELS_MSTR_SLIP_SEC This variable sets the address of the secondary master signal to be monitored.

ELS_MSTR_CONFIG This variable is used to set the different modes of operation for the master slip monitoring.

Bit	Description
1 to 30	Reserved
31	slip offset method – 0 = Fixed Absolute Offset, 1 = Dynamic Offset
32	slip offset reaction – 0 = Fatal Error, 1 = Warning

Table 6-12: ELS_MSTR_CONFIG Bits

ELS_MSTR_CONTROL

Bit	Description
1 to 6	Reserved
7	Master 1, Set Reference Position
8	Master 2, Set Reference Position
9	Master 3, Set Reference Position
10	Master 4, Set Reference Position
11	Master 5, Set Reference Position
12	Master 6, Set Reference Position
13 to 14	Reserved
15	Capture Slip Monitoring (0 = Off, 1 = On)
16	Enable Slip Monitoring (0 = Disabled, 1 = Enabled)

Table 6-13: ELS_MSTR_CONTROL Bits

ELS_MSTR_STATUS

Bit	Description
1	Master 1 at Standstill
2	Master 2 at Standstill
3	Master 3 at Standstill
4	Master 4 at Standstill
5	Master 5 at Standstill
6	Master 6 at Standstill
7	Master 1 Referenced
8	Master 2 Referenced
9	Master 3 Referenced
10	Master 4 Referenced
11	Master 5 Referenced
12	Master 6 Referenced
13	Reserved
14	Monitoring ERROR Active (0 = No Error, 1 = Error)
15	Lead Encoder (0 = Primary, 1 = Secondary)
16	Slip Monitoring Enabled (0 = Not Enabled, 1 = Enabled)

Table 6-14: ELS_MSTR_STATUS Bits

ELS Group 1- 8 Default Register Labels

The default labels for the ELS Group registers are listed in Table 6-15. The corresponding default bit labels are listed in Table 6-17.

Data Object Type	Label (20 character limit)	Comment (80 character limit)
Assigned control register number	G#_CONTROL_REG	Group # control register
Assigned status register number	G#_STATUS_REG	Group # status register

Table 6-15: ELS Group 1- 8 Default Registers

Default Label ELS Group 1-8	Data Object ELS Group Control Register-Bit								Comment (80 character limit) ELS Group 1-8
Control Register	1	2	3	4	5	6	7	8	Control Register
G#_CT_LOCK_OFF	152-1	153-1	154-1	155-1	156-1	157-1	158-1	159-1	Group # control, 0 → 1 start lock cycle, 1 → 0 start unlock
G#_CT_M_REL_PH	152-2	153-2	154-2	155-2	156-2	157-2	158-2	159-2	Group # control, 0 → 1 triggers master relative phase adjust
G#_CT_S_REL_PH	152-3	153-3	154-3	155-3	156-3	157-3	158-3	159-3	Group # control, 0 → 1 triggers slave relative phase adjust
G#_CT_MSTR_SEL	152-4	153-4	154-4	155-4	156-4	157-4	158-4	159-4	Group # control, 0=master 1, 1=master 2
G#_CT_VAR_CLK	152-5	153-5	154-5	155-5	156-5	157-5	158-5	159-5	Group # control, 0 → 1 forcing
G#_CT_LOCAL	152-6	153-6	154-6	155-6	156-6	157-6	158-6	159-6	Group # control, 0 → 1 local mode, 1 → 0 selected master
G#_CT_LM_FSTOP	152-7	153-7	154-7	155-7	156-7	157-7	158-7	159-7	Group LM # control, 0 → 1 triggers fast stop
G#_CT_LM_HOME	152-8	153-8	154-8	155-8	156-8	157-8	158-8	159-8	Group LM # control, 0 → 1 loads home position
G#_CT_LM_GO	152-9	153-9	154-9	155-9	156-9	157-9	158-9	159-9	Group LM # control, 0=stop, 1=go
G#_CT_LM_VMODE	152-10	153-10	154-10	155-10	156-10	157-10	158-10	159-10	Group LM # control, 0=position, 1=velocity mode
G#_CT_M_ABS_PH	152-11	153-11	154-11	155-11	156-11	157-11	158-11	159-11	Group # control, triggers master absolute phase adjust
G#_CT_S_ABS_PH	152-12	153-12	154-12	155-12	156-12	157-12	158-12	159-12	Group # control, triggers slave absolute phase adjust
G#_CT_LM_RELMODE	152-13	153-13	154-13	155-13	156-13	157-13	158-13	159-13	Group LM # control, 0=absolute, 1=relative mode
G#_CT_LM_RELTRIG	152-14	153-14	154-14	155-14	156-14	157-14	158-14	159-14	Group LM # control, 0 → 1 triggers relative mode move
G#_CT_MSTR_FOL_PH	152-15	153-15	154-15	155-15	156-15	157-15	158-15	159-15	Group # control, 0=standard phase adjust 1=group master following phase adjust enabled
G#_CT_MSTR_ABORT_PH	152-16	153-16	154-16	155-16	156-16	157-16	158-16	159-16	Group # control, 0=n/a 1=abort active group master following phase adjust
Each # symbol represents an entry for the number of the ELS Group									

Table 6-16: ELS Group 1- 8 Default Register Bits

Default Label ELS Group 1-8	**Data Object ELS Group Status Register-Bit**								**Comment (80 character limit) ELS Group 1-8**
Status Register	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Status Register**
G#_ST_LOCK_ON	243-1	244-1	245-1	246-1	247-1	248-1	249-1	250-1	Group # status, 0=unlocked, 1=locked to master
G#_ST_M_REL_PH	243-2	244-2	245-2	246-2	247-2	248-2	249-2	250-2	Group # status, 1= master relative phase adjust in progress
G#_ST_S_REL_PH	243-3	244-3	245-3	246-3	247-3	248-3	249-3	250-3	Group # status, 1= slave relative phase adjust in progress
G#_ST_MSTR_SEL	243-4	244-4	245-4	246-4	247-4	248-4	249-4	250-4	Group # status, 0=master 1, 1=master 2
G#_ST_VAR_ACK	243-5	244-5	245-5	246-5	247-5	248-5	249-5	250-5	Group # status, 1=variables updated
G#_ST_LOCAL	243-6	244-6	245-6	246-6	247-6	248-6	249-6	250-6	Group # status, 1=local mode active
G#_ST_LM_FSTOP	243-7	244-7	245-7	246-7	247-7	248-7	249-7	250-7	Group LM # status, 0 → 1 triggers fast stop
G#_ST_LM_HOME	243-8	244-8	245-8	246-8	247-8	248-8	249-8	250-8	Group LM # status, 0 → 1 loads home position
G#_ST_MOTION	243-9	244-9	245-9	246-9	247-9	248-9	249-9	250-9	Group # status, 0=no motion, 1=group is in motion
G#_ST_LM_VMODE	243-10	244-10	245-10	246-10	247-10	248-10	249-10	250-10	Group LM # status, 0=position, 1=velocity mode
G#_ST_M_ABS_PH	243-11	244-11	245-11	246-11	247-11	248-11	249-11	250-11	Group # status, 1= master absolute phase adjust in progress
G#_ST_S_ABS_PH	243-12	244-12	245-12	246-12	247-12	248-12	249-12	250-12	Group # status, 1= slave absolute phase adjust in progress
G#_ST_LM_RELMODE	243-13	244-13	245-13	246-13	247-13	248-13	249-13	250-13	Group LM # status, 0=absolute, 1=relative mode
G#_ST_LM_ZEROVEL	243-14	244-14	245-14	246-14	247-14	248-14	249-14	250-14	Group LM # status, 0=velocity, 1=standstill
G#_ST_LM_INPOS	243-15	244-15	245-15	246-15	247-15	248-15	249-15	250-15	Group LM # status, 1=in position
Each # symbol represents an entry for the number of the ELS Group									

Table 6-17: ELS Group 1- 8 Default Register Bits

ELS Group 1- 8 Default Program Variable Labels

The default labels for ELS Group program variables are listed in the following table:

Default Label ELS Group 1-8	**Data Object ELS Group 1-8 Program Variable**								**Comment (80 character limit) ELS Group 1-8**	**Update Mode**
Program Variable	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Program Variable**	
G#_SYNC_ACCEL	F190	F230	F270	F310	F350	F390	F430	F470	Group #, dynamic sync acceleration	Phase 4
G#_SYNC_VEL	F191	F231	F271	F311	F351	F391	F431	F471	Group #, dynamic sync velocity	Phase 4
G#_M1	F192	F232	F272	F312	F352	F392	F432	F472	Group #, M factor	Phase 4 & Forcing*
G#_N1	F193	F233	F273	F313	F353	F393	F433	F473	Group #, N factor	Phase 4 & Forcing*
G#_PROG_M_PH	F194	F234	F274	F314	F354	F394	F434	F474	Group #, master phase adjust value	Phase 4
G#_PROG_S_PH	F195	F235	F275	F315	F355	F395	F435	F475	Group #, slave phase adjust value	Phase 4
G#_ABS_M_PH	F196	F236	F276	F316	F356	F396	F436	F476	Group #, absolute master phase adjust	Phase 4 (read-only) & Forcing*
G#_ABS_S_PH	F197	F237	F277	F317	F357	F397	F437	F477	Group #, absolute slave phase adjust	Phase 4 (read-only) & Forcing*
G#_H_LOCKON	F198	F238	F278	F318	F358	F398	F438	F478	Group #, H factor lock on cam profile	Phase 4 & Forcing*
G#_H_RUN	F199	F239	F279	F319	F359	F399	F439	F479	Group #, H factor 1:1 cam profile	Phase 4 & Forcing*
G#_H_LOCKOFF	F200	F230	F310	F320	F360	F400	F440	F480	Group #, H factor lock off cam profile	Phase 4 & Forcing*
G#_H_USER	F201	F231	F311	F321	F361	F401	F441	F481	Group #, H factor user cam profile	Phase 4
G#_LOCK_WIN	F202	F232	F312	F322	F362	F402	F442	F482	Group #, shortest path window for dynamic sync. phase correction	Phase 4
G#_LM_HOME_POS	F203	F233	F313	F323	F363	F403	F443	F483	Group Local Master # home position	Phase 4
G#_LM_REL_MOVE_DIST	F204	F234	F314	F324	F364	F404	F444	F484	Group Local Master # relative move distance	Phase 4
G#_LM_STOP_POS	F205	F235	F315	F325	F365	F405	F445	F485	Group Local Master # stop position	Phase 4
G#_LM_CMD_ABS_POS	F206	F236	F316	F326	F366	F406	F446	F486	Group Local Master # commanded absolute position	Phase 4
G#_LM_CMD_VEL	F207	F237	F317	F327	F367	F407	F447	F487	Group Local Master # commanded velocity	Phase 4
G#_LM_CMD_ACCEL	F208	F238	F318	F328	F368	F408	F448	F488	Group Local Master # commanded acceleration	Phase 4
G#_LOCKON_OFFSET	F209	F239	F319	F329	F369	F409	F449	F489	Group #, offset added to the output when lock on cam profile is being forced	Phase 4
G#_IN_POS	F210	F240	F290	F330	F370	F410	F450	F490	Group #, input position	Phase 4 & Forcing*
G#_IN_VEL	F211	F241	F291	F331	F371	F411	F451	F491	Group #, input velocity (read only)	Phase 4
G#_OUT_POS	F212	F242	F292	F332	F372	F412	F452	F492	Group #, output position (read only)	Phase 4 & Forcing*
G#_OUT_VEL	F213	F243	F293	F333	F373	F413	F453	F493	Group #, output velocity (read only)	Phase 4

Table 6-18: ELS Group 1- 8 Default Program Variables (part 1 of 2)

Default Label ELS Group 1-8 Program Variable	Data Object ELS Group 1-8 Program Variable 1	2	3	4	5	6	7	8	Comment (80 character limit) ELS Group 1-8 Program Variable	Update Mode
G#_OUT_ACC	F214	F254	F294	F334	F374	F414	F454	F494	Group #, output acceleration (read only)	Phase 4
G#_CAM_INPUT	F215	F255	F295	F335	F375	F415	F455	F495	Group #, group cam profile ID input position	Phase 4 & Forcing*
G#_MST1_TRIGPOS	F216	F256	F296	F336	F376	F416	F456	F496	Group #, master 1 switching trigger position	Phase 4
G#_MST2_TRIGPOS	F217	F257	F297	F337	F377	F417	F457	F497	Group #, master 2 switching trigger position	Phase 4
G#_STANDSTILL_WIN	F218	F258	F298	F338	F378	F418	F458	F498	Group # standstill velocity threshold	Phase 4
G#_LM_CMD_DECEL	F219	F259	F299	F339	F379	F419	F459	F499	Group Local Master #, commanded deceleration	Phase 4
G#_LM_E_STOP_DECEL	F220	F260	F300	F340	F380	F420	F460	F500	Group Local Master #, E-Stop deceleration	Phase 2
G#_LM_MAX_VEL	F221	F261	F301	F341	F381	F421	F461	F501	Group Local Master #, maximum velocity	Phase 2
G#_LM_MAX_ACCEL	F222	F262	F302	F342	F382	F422	F462	F502	Group Local Master #, maximum acceleration	Phase 2
G#_LM_MAX_DECEL	F223	F263	F303	F343	F383	F423	F463	F503	Group Local Master #, maximum deceleration	Phase 2
G#_LM_JERK_ENABLE	F224	F264	F304	F344	F384	F424	F464	F504	Group Local Master #, jerk limiting enabled	Phase 2
G#_LM_CUR_POS	F225	F265	F305	F345	F385	F425	F465	F505	Group Local Master #, current position	Phase 4
G#_LM_CUR_VEL	F226	F266	F306	F346	F386	F426	F466	F506	Group Local Master #, current velocity	Phase 4
G#_LM_POS_WIN	F227	F267	F307	F347	F387	F427	F467	F507	Group Local Master #, shortest path window	Phase 2
G#_MST_DIST_M_PH	F228	F268	F308	F348	F388	F428	F468	F508	Group #, master distance for master phase adjust	Phase 4
G#_MST_DIST_S_PH	F229	F269	F309	F349	F389	F429	F469	F509	Group #, master distance for slave phase adjust	Phase 4
G#_CONFIG	I140	I150	I160	I170	I180	I190	I200	I210	Group #, configuration word	Refer to Fig. 5-2
G#_MSTR1_AXIS	I141	I151	I161	I171	I181	I191	I201	I211	Group #, ELS master ID, number 1	Phase 4
G#_MSTR2_AXIS	I142	I152	I162	I172	I182	I192	I202	I212	Group #, ELS master ID, number 2	Phase 4
G#_ACTIVE_STATE	I143	I153	I163	I173	I183	I193	I203	I213	Group #, active state of state machine for lockon/lockoff	Phase 4 & Forcing*
G#_ACTIVE_CAM	I144	I154	I164	I174	I184	I194	I204	I214	Group #, active cam profile table number	Phase 4
G#_LOCKON_CAM	I145	I155	I165	I175	I185	I195	I205	I215	Group #, lock on cam profile table number	Phase 4 & Forcing*
G#_RUN_CAM_ID	I146	I156	I166	I176	I186	I196	I206	I216	Group #, 1:1 cam profile table number	Phase 4 & Forcing*
G#_LOCKOFF_CAM	I147	I157	I167	I177	I187	I197	I207	I217	Group #, lock off cam profile table number	Phase 4 & Forcing*
G#_USER_CAM	I148	I158	I168	I178	I188	I198	I208	I218	Group #, user cam profile table number (state machine disabled)	Phase 4
G#_LM_POS_MODE	I149	I159	I169	I179	I189	I199	I209	I219	Group Local Master #, positioning mode see note 1)	Phase 2
Forcing is reinitializing an ELS Group in Phase 4 when local mode is active (G#_ST_LOCAL) and the ELS Group Master is at standstill (G#_ST_MOTION is 0). Note 1.) Absolute Position Mode, 0=Positive, 1=Negative, 2=Shortest Path										

Table 6-19: ELS Group 1- 8 Default Program Variables (part 2 of 2)

ELS Group Configuration Word

For every ELS Group, an ELS Group configuration word (**G#_CONFIG**) is used to configure all settings for Switching Synchronization, Phase Control and Initialization. These settings are initially configured within VisualMotion Toolkit's ELS Group icon and become active when the project is compiled and downloaded to the control. These settings can also be modified by accessing the appropriate integer number and entering an equivalent hexadecimal value.

Bit 2:
Sync. to ELS Group Master

When the control is switched to manual mode, all ELS Groups are switched to local mode. In local mode, each ELS Group can be moved independently. When switching back to automatic mode, the user can configure bit 2 using the following two options:

0 = Automatically switch back to the ELS Group master and perform a dynamic synchronization if necessary, see bits 5, 6 and 7 (***default***)

1 = Groups will stay in local mode and must be switched manually

(*Updated in Phase 2*)

Bit 4:
Enable CAM Profiling

This bit enables the lock on / lock off CAM profile state. For user CAM profiles to function, disable this feature.

0 = state machine enabled (***default***)

1 = state machine disabled

(*Updated in Phase 2*)

Modifications to the variable G#_H_USER can only be performed when the state machine is disabled. While disabled, the user can select a CAM profile for the ELS Group and modify the G#_H_USER factor. When enabled, the state machine uses as an H factor the values of G#_H_LOCKON, G#_H_LOCKOFF and G#_H_RUN. The G#_H_USER variable displays the current H factor being used for the lock on and lock off CAM profiles.

Bit 5:
Synchronization Type

This bit is used to specify the type of synchronization that will be used when switching between ELS Group input masters 1 and 2 or only Group master 1 in Advanced Mode (see bit 22 for master 2).

0 = Dynamic synchronization (***default***)

1 = Immediate (*On the Fly when switching to an unused Virtual Master*)

(*Updated in Phase 4*)

When bit 5 is set to 1 and an ELS Group's input master is switched to an unused Virtual Master, this Virtual Master will adapt "On the Fly" to the current ELS Group master's position and velocity.

Bits 6-7:
Phase Correction Type

These bits determine method of phase correction during Dynamic Synchronization between group input masters 1 and 2 or for group input master 1 only in Advanced Mode (see bits 23, 24 for group 2).

Bit 6	Bit 7	Description
0	0	Shortest path (***default***)
1	0	Positive direction if phase difference is greater than "G#_LOCK_WIN". Otherwise, use shortest path.
0	1	Negative direction if phase difference is greater than "G#_LOCK_WIN". Otherwise, use shortest path.
1	1	No phase correction (*only velocity synchronization is performed*)

(*Updated in Phase 4*)

Bit 8:
ELS Group Master Position Initialization at Phase 2

This bit is used to reinitialize an ELS Groups output master position when the system is switched to Phase 2 (parameter mode) or powered down.

When an ELS Group's M/N or H factor has a value other than 1; for example 0.9, and the ELS Group has been moved, then the group's output master position cannot be calculated using the CAM equation.

The reason for this is as follows:

The control monitors and internally stores the ELS Group's current output position. For example, if after two revolutions of the input master (as illustrated in Fig. 6-4), the system is switched to Phase 2 or loses power; the ELS Group's output master position is stored. The user has the option to restart the ELS Group, to an initial position, by setting bit 8 to 0. This will recalculate the ELS Group's output master position using the CAM equation. Setting bit 8 to 1 allows the ELS Group's output master position to start from the stored position (old values) and continue; using the CAM equation, for consecutive revolutions of the ELS Group's input master.

[(input master * M/N) + master offset]H + slave offset = Group output

Group output position with a 0.9 M/N and no offsets

[(0° * 0.9) + 0°] * 1 + 0° ⇒ 0° ;initial position at start

[(0° * 0.9) + 0°] * 1 + 0° ⇒ 324° ;after one revolution

[(0° * 0.9) + 0°] * 1 + 0° ⇒ 288° ;after second revolution

Fig. 6-4: CAM Equation Example

0 = Initialization with calculated value using the CAM equation (***default***)

1 = Use old values

(*Updated in Phase 2 & Forcing*)

Bit 9:
Group Master Position Evaluation with Forcing

This bit is used to initialize an ELS Group's output position when switched to local mode (G#_CT_LOCAL).

0 = Group master positions will be calculated using CAM equation (***default***)

1 = Use old values

(*Updated in Phase 2 & Forcing*)

When forcing states 0 or 1 with this bit reset, G#_ LOCKON_OFFSET is added to the group master output position. If this bit is set, G#_ LOCKON_OFFSET is not used.

Bit 11:
Master Phase Adjust Type

This bit sets the motion profile type for the active master.

0 = Trapezoidal profile using a velocity profile with dynamic synchronization acceleration/deceleration and additive velocity (***default***)

1 = Immediate – step function

(*Updated in Phase 4*)

Bit 12:
Slave Phase Adjust Type

This bit sets the motion profile type for the all slave axes associated with the ELS Group.

0 = Trapezoidal profile using a velocity profile with dynamic synchronization acceleration/deceleration and additive velocity (***default***)

1 = Immediate – step function

(*Updated in Phase 4*)

Rexroth
Bosch Group

Bit 17: Advanced Master Switching Options

0 = Disabled – Enables standard ELS Group Master Switching functionality in VisualMotion 9

1 = Enabled – Enables Enhanced ELS Group Master Switching

Bits 18-19: These bits specify a condition that must be met before the process of switching to Group Master 1 is triggered.

Bit 18	Bit 19	Switching Trigger	Description
0	0	Instantaneous (default)	Master switching is triggered as soon as the G#_CT_MSTR_SEL group control bit is changed.
1	0	Master 1 Position	Master switching is triggered when Group Master 1 passes the position defined in float variable G#_MSTR1_TRIGPOS
0	1	Master 2 Position	Master switching is triggered when Group Master 2 passes the position defined in float variable G#_MSTR2_TRIGPOS
1	1	Optimal	***Dynamic Switching***: switching is triggered when the group's constant acceleration ramp (using G#_SYNC_ACC) will result in the group synchronizing to the target master's position and velocity nearly simultaneously ***Immediate switching***: switching is triggered when the groups master's positions coincide

Bits 20-21 These bits specify a condition that must be met before the process of switching to Group Master 2 is triggered. The bit format is the same as for bits 18-19.

Bit 22 This bit indicates the type of ELS group master synchronization used when switching to Group Master 2. This requires the Master Switching Option bit to be set.

0 = Dynamic synchronization

1 = Immediate synchronization

Bits 23-24 This bit indicates the options for phase correction during dynamic synchronization when switching to Group Master 2. This requires the Master Switching Option bit to be set. The bit format is the same as bits 6 and 7.

Bit 32 Reserved for internal use.

ELS System Master

The ELS System Masteris a position signal router and conditioner. Signals used in the ELS system are brought into the control through these masters. VisualMotion supports 6 ELS System Master per control.

Initializing Registers and Program Variables The initialization of control and status registers and project variable labels for all 6 ELS System Masters is defined under ***Setup ⇒ ELS ⇒ System Masters*** in the Project Navigator window.

Note: From the Setup configuration, only one ELS System Master must be configured. This is the only time control and status registers as well as variable labels for all 6 ELS System Masters are assigned. Afterwards, additional System Masters can be added under Setup or by using the ELSMstr icon in the Initialization task. The ELSMstr icon can also be used to modify System Masters configured under Setup as part of the logical program flow. Refer to *System Masters Setup* in section 4.5, *Setup ELS Processes*, for details.

Modifying or Declaring System Masters at Runtime

To declare ELS System Master runtime data in a project, open the Initialization task, select the ELSMstr icon, and place it in the project workspace. VisualMotion provides several methods for declaring initial values to System Masters. Refer to *Initialization of ELS Program Variables* in section 4.5, *Setup ELS Processes*, for details.

Support signal types include Virtual Masters, Real Masters (incremental, single-turn and multi-turn absolute encoders, resolver, linear scale, primary motor feedback), Master Encoder Card, Link Ring Masters (optional peer-to-peer control link), and Group outputs. The position output is always based on units of degrees and has a range between 0 and 360 degrees with the module (rollover) fixed at 360 degrees. Although you have the ability to select many different signal sources, some may not be compatible with the master source. For example, if a drive’s primary position signal is set to linear mode with a current position of 1000mm input into the system master using a 1:1 coupling ratio, it would output 280 degrees (1000 mod 360), which could be invalid.

The following are rules for signal selection and configuration:

- Virtual Masters, Real Masters, Link Ring Masters, Group outputs, and Master Encoder Card have different properties and require specific parameters. The following table contains the parameters relevant for each master type:

Master Type	Coupling Ratio (M/N)	Filter	Dead Time Comp.	Master Offset	Reference Position	Master Pos/Vel
Virtual Master	1:1 (fixed)	No	No	No	No	Yes
Real Master using Secondary Feedback	Yes (as integers)	Yes	Yes	Yes	Yes	Yes
Real Master using Primary Feedback	Yes (as floats)	Yes	Yes	No	No	Yes
Link Ring	1:1 (fixed)	No	No	No	No	Yes
Group output	1:1 (fixed)	No	No	No	No	Yes
Master Encoder Card	Yes (as integers)	Yes	Yes	Yes	Yes	Yes

Table 6-20: Master Parameters

Note: To use the primary motor feedback as a master source signal, the drive must be set to modulo operation (not absolute) and the modulo value (S-0-0103) must be set to 360 degrees.

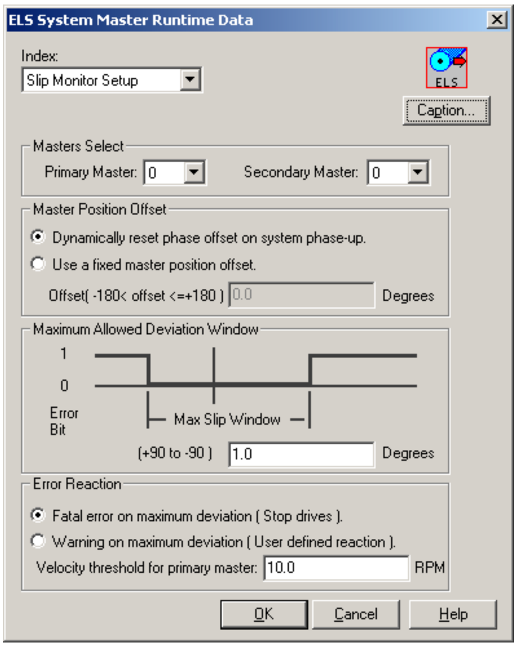
Slip Monitoring for ELS System Masters

Slip monitoring is a feature in VisualMotion that can detect if an encoder is operating properly by comparing its signal with a second encoder signal. In comparing the two signals, slip monitoring can determine the extent of deviation and if the deviation exceeds the maximum allowable range set in VisualMotion. The system response to a deviation outside of the range is also determined by settings in VisualMotion.

The slip monitoring feature can be used with any type of master signal, an internal signal or a signal from an external encoder. A difference in gear ratio between the two signals will cause the primary signal to continually deviate beyond the range limits over time. If a gear reduction exists between the two encoders, the ELS_MSTR_Mx and ELS_MSTR_Nx variables must be set to compensate for the mechanical gear ratio difference.

The slip monitoring feature is initially setup through the *System Master Assignment* window, which is displayed when placing an ELS Master Assignment icon in the VisualMotion programming workspace. After initial setup of the feature, slip monitoring can be modified through its designated program variables.

Slip Monitoring is initialized from the *ELSMstr* icon by selecting **Slip Monitor Setup** for the Index drop-down list.



Slip_Monitor_Setup.tif

Fig. 6-5: Slip Monitor Setup Window

Masters Select

The Primary and Secondary master numbers correspond to the **master** number assignments in the *System Master Setup*.

The numbers selected here correspond to the addresses that are set in the default variables ELS_MSTR_SLIP_PRI, for the primary number, and ELS_MSTR_SLIP_SEC, for the secondary number.

Note: The default values of 0 for both the primary and secondary masters indicate that the Slip Monitoring functionality is disabled.

Master Position Offset

An offset can be applied after the initial slip monitoring calculation of the difference between signals. The offset value is stored in the

ELS_MSTR_SLIP_OFFSET default label. There are two methods for applying to offset:

Dynamically Reset phase offset on system phase up – The system records the position difference between the two masters on entering phase 4 and on the rising edge of the ELS_M_ CT_SLIP_CAPT1 bit.

Fixed Offset – you select a fixed value between –180 and 180 degrees for the offset, which is stored in the ELS_MSTR_SLIP_OFFSET variable. Changes to the fixed offset value are updated on entering phase 4 and on the rising edge of the ELS_M_ CT_SLIP_CAPT1 bit.

Maximum Allowed Deviation Window

The value entered in this field represents the range between the minimum and maximum limits for the position difference of signals and it is assigned the ELS_MSTR_SLIP_OFFSET default label. This value is updated in Parameter mode.

Error Reaction

If the maximum allowed deviation value is exceeded, one of the following responses can be set:

- **Fatal error** – This response will result in a system shutdown. Shutdown Message #552 Excessive Master Position Slip Deviation is issued.
- **Warning** – This response results in Warning Message #221 Excessive Master Position Slip Deviation being issued. This option allows you to configure a response, such as switching to the lead encoder signal.

In the **Velocity threshold for primary master** field, the value entered is the velocity that the primary master encoder must exceed before the system can detect the direction of the primary encoder. This value is referenced by the default variable label ELS_MSTR_SLIP_VELTHD.

This value is also used to detect the lead when a deviation error occurs. The lead encoder bit (ELS_M_ST_SLIP_ENC of the ELS Master Configuration Word (see *ELS Master Configuration Word*, 6-10) is set by the control based on which master encoder is assumed to be leading at the time of the error, according to the following table.

Primary Master Velocity	Secondary Master Velocity	Monitoring Cycle	Deviation Error*** (ELS_MSTR_SLIP_ACTUAL)	Lead Encoder Status Bit (ELS_MSTR_STATUS Reg, Bit 15)
Positive	N/A	Any	Positive	0 = Primary Master is Lead
Positive	N/A	Any	Negative	1 = Secondary Master is Lead
Negative	N/A	Any	Negative	0 = Primary Master is Lead
Negative	N/A	Any	Positive	1 = Secondary Master is Lead
None (0)*	N/A	First**	Positive or Negative	0 = Primary Master is Lead
None (0)*	None (0)*	Any	Positive or Negative	0 = Primary Master is Lead
None (0)*	Positive or Negative	Not First	Positive or Negative	1 = Secondary Master is Lead

* A velocity of “None (0)” means that the specified slip master’s velocity is not exceeding the Slip Masters Standstill Velocity Threshold (ELS_MSTR_SLIP_ VELTHD) value.

** “First Monitoring Cycle” refers to the first active Slip Monitoring cycle after:

- Entering Sercos phase 4 with the Slip Monitoring Enable Control bit already set high.
- Bringing the Slip Monitoring Enable Control bit high (rising edge) while in Sercos phase 4.
- Bringing the Slip Monitoring Capture Control bit high (rising edge), with the Enable bit already set high, while in Sercos phase 4.

*** “Deviation Error” refers to a Current Slip Deviation (ELS_MSTR_SLIP_ACTUAL) value that *exceeds* the Maximum Allowed Deviation Window (ELS_MSTR_SLIP_WINDOW) value.

Table 6-21: System Logic for Determining Lead Encoder

Warning: If a non-zero master position offset is used in a project, with automatic master switching based on the lead encoder status bit when the maximum allowed deviation is exceeded, the system position may shift by an amount close to that of the master position offset. The result could be a significant shift in position when the lead encoder status bit changes.

The slip monitoring feature captures the primary and secondary master signals every Sercos cycle update and compares the two values to determine the difference in the signals, refer to the figure below. An offset can be applied to the difference of the two signals if necessary. The resulting deviation value is measured to determine if it is within the limits of the maximum allowed deviation window.

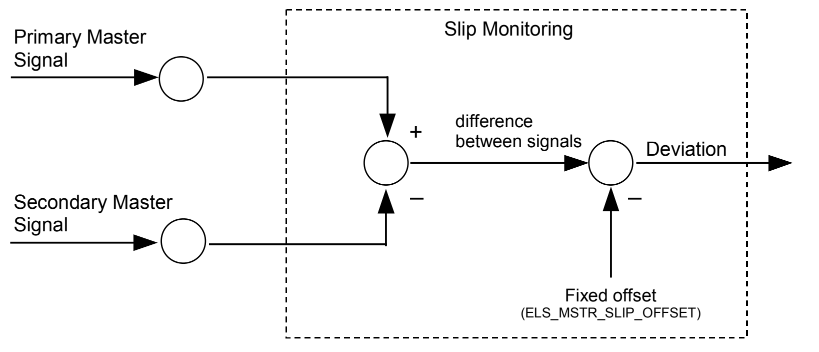


Fig. 6-6: Illustration of Slip Monitoring Feature Applied to Master Signals

Slip Monitoring Variables

Default variables assigned for the slip monitoring features are listed in the table below. The Data Object column indicates the float or integer number of the label. Update mode indicates which phase of the project update cycle the label value is updated in the control. Some values, such as the dynamic reset phase offset value and maximum allowed deviation window value, are not updated until the project transitions from phase 2 to phase 4.

Default Label	Data Object	Comment	Update Mode
ELS_MSTR_SLIP_WINDOW	F158	Maximum Allowed Deviation Window	Captured upon entry in to phase 4 and on the rising edge of the capture bit.
ELS_MSTR_SLIP_OFFSET	F159	Maximum Position Offset	Fixed offset - Phase 2 Dynamic reset offset - Captured upon entry in to phase 4 and on the rising edge of the capture bit.
ELS_MSTR_SLIPVELTHD	F160	Primary Master Velocity Threshold	Phase 2
ELS_MSTR_SLIP_PEAK	F161	Peak Slip Deviation	Phase 4
ELS_MSTR_SLIP_ACTUAL	F162	Current Slip Deviation (actual)	Phase 4
ELS_MSTR_SLIP_PRI	I134	Primary Master Signal Address	Phase 2
ELS_MSTR_SLIP_SEC	I135	Secondary Master Signal Address	Phase 2

Table 6-22: ELS Slip Monitoring Feature Variables

Using Register Bits to Adjust Slip Monitoring Feature

The Bits used to enable, monitor and activate the slip monitoring feature are listed with their associated labels and registers in the table below.

Default Label ELS Master 1 to 6	Data Object ELS Master Register-Bit	Comment (80 character limit)
ELS_M_CT_SLIP_CAPT1	140-15	ELS Master Capture Slip Monitoring 0 = No capture 1 = Capture Run-Time Data

Default Label ELS Master 1 to 6	Data Object ELS Master Register-Bit	Comment (80 character limit)
ELS_M_CT_SLIP_EN	140-16	ELS Master Enable Slip Monitoring 0 = Off 1 = On
ELS_M_ST_SLIP_ERR	141-14	ELS Master Slip Monitoring Error 0 = No error 1 = Error
ELS_M_ST_SLIP_ENC	141-15	ELS Master Slip Monitoring Lead Encoder 0 = Primary 1 = Secondary
ELS_M_ST_SLIP_ENA	141-16	ELS Master Slip Monitoring Enabled 0 = Off 1 = On

Table 6-23: ELS Slip Monitor Feature Default Register Bits

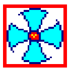
The slip monitoring feature is enabled with the ELS_M_CT_SLIP_EN bit. Slip monitoring is monitored by the ELS_M_ST_SLIP_ENA, ELS_M_ST_SLIP_ERR, and ELS_M_ST_SLIP_ENC bits. The ELS_M_ST_SLIP_ENA bit indicates that slip monitoring has been enabled. The ELS_M_ST_SLIP_ERR bit indicates an excessive master position slip deviation error is active. To clear the error:

1. Bring the masters back in to alignment by physically moving the masters if a fixed master position offset is being used or by toggling the ELS_M_CT_SLIP_CAPT1 bit if dynamic reset phase offset is being used.
2. Clear the error by toggling the ELS_M_CT_SLIP_EN bit or by toggling bit 5, CLEAR_ALL_ERRORS, of the control register.

The ELS_M_ST_SLIP_ENC monitors the encoder signals and determines the lead signal, which it indicates by displaying a 0 for primary or 1 for secondary. The ELS_M_CT_SLIP_CAPT1 bit, with dynamic reset of phase offset selected, captures the slip offset of the master positions.

Rexroth
Bosch Group

Virtual Master



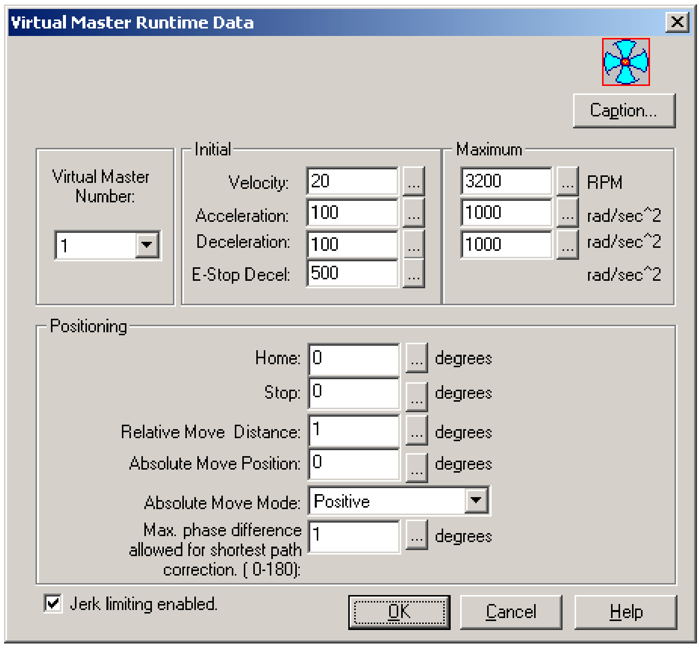
A Virtual Master is an internal motion profiler that drives geared or CAM axes in a machine. A set of control and status parameters and registers command the axis to run at a set speed, stop at a position, or make a profiled move. The command stream is based on units of degrees and has a range between 0 and 360 degrees; with the module (rollover period) fixed at 360 degrees. This output format (0 and 360) forms the basis for the entire ELS system, where a machine/product cycle is defined as a single revolution. For example, a system that is commanded by a Virtual Master running at 300 RPM produces 300 products per minute.

Initializing a Virtual Master

A Virtual Master is controlled by a VisualMotion project and/or a PLC using I/O registers and project variables. GPP supports a maximum of 2 Virtual Masters. The initialization of control and status registers and project variable labels is defined under ***Setup ⇒ ELS ⇒ Virtual Masters*** in the Project Navigator window. Refer to *Virtual Master Setup* in section 4.5, *Setup ELS Processes*, for details.

Declaring Virtual Master Runtime Data

To declare Virtual Master runtime data in a project, open the Initialization task, select the VM1 icon, and place it in the project workspace. VisualMotion provides several methods for declaring initial values to Virtual Masters. Refer to *Initialization of ELS Program Variables* in section 4.5, *Setup ELS Processes*, for details.



vm_init_values.tif

Fig. 6-7: Virtual Master Runtime Data

Note: Virtual Masters can be associated to task A or to the PLC (IndraLogic). The association is configured under *Setup ⇒ Axes*. Any motion associated with a Virtual Master will stop if task A or the PLC stop running.

A Virtual Master can be controlled from anywhere in the runtime program and/or outside the icon program from a PLC using the defined control registers and parameter variables.

Virtual Master 1 or 2 signals can not be used directly, they must first be defined as one of the six system masters, then used to drive a group master input, PLS or Rotary event.

Initial and Maximum Values

The initial and maximum values set for each Virtual Master in the *Virtual Master Runtime Data* window are embedded in the icon. During program compiling, an initial value of 0 is assigned to all the Virtual Master program variables. At runtime, the default values or user-defined values declared in the VM1 icon are assigned when the program flow encounters the VM1 icon.

Note: VisualMotion supports various methods for assigning value to program variables. Refer to *Initialization of ELS Program Variables* in section 4.5, *Setup ELS Process*, for details.

Velocity

The *Initial* Velocity value defines a constant velocity that the Virtual Master will accelerate towards when set in motion.

The *Maximum* Velocity value defines the maximum velocity that can be achieved by the Virtual Master during runtime.

Note: The Virtual Master moves in a clockwise (positive) direction when a non-negative velocity value is used. A negative velocity value causes the Virtual Master to move in a counter clockwise (negative) direction.

The velocity the drive (axes) can obtain following a Virtual Master is limited by the drive's Bipolar Velocity Limit. The following conditions also affect the drive velocity:

- If the Virtual Master's maximum velocity is less than the drive's Bipolar Velocity Limit Value (S-0-0091), the Virtual Master limits the drive.
- If the Virtual Master's maximum velocity is greater than the drive's Bipolar Velocity Limit Value (S-0-0091), the drive will fault when S-0-0091 is exceeded.

Acceleration

The *Initial* Acceleration value defines a constant acceleration that the Virtual Master will use to achieve a desired velocity.

The *Maximum* Acceleration value defines the maximum acceleration that can be achieved by the Virtual Master during runtime.

Deceleration

The *Initial* Deceleration value defines a constant deceleration that the Virtual Master will use to decelerate the velocity.

The *Maximum* Deceleration value defines the maximum deceleration that can be achieved by the Virtual Master during runtime.

E-Stop Deceleration

This value specifies the emergency stop deceleration for each Virtual Master.

Positioning

Positioning values are embedded in the icons and are stored as project variables when the project is compiled.

Home

The Virtual Master starts an immediate move to the home position when a low-to-high transition is seen in the Virtual Master control register bit:

Control Register Bit	State
Bit 2 (VM#_CT_HOME)	0 → 1

This value is written to project variable VM#_CUR_POS within one Sercos cycle of the home bit's rising edge. This process will take longer than one Sercos cycle if jerk limiting is enabled.

The bit transition will be ignored if the VM#_CT_GO bit is high or if the Virtual Master is actively being used by an ELS Group. If these conditions exist, the VM#_CT_HOME bit will stay low to indicate that the homing request has been ignored. Otherwise, the virtual master's home status bit will be raised after the virtual master has been set to the home position to indicate the operation is complete.

Stop

This is the programmed stop position the Virtual Master moves to when the control is switched from velocity mode to absolute positioning mode. This field contains the initial value that is written to project variable VM#_STOP_POS when the project is compiled. The state of the following bits determine the mode of operation:

Control Register Bit	Velocity Mode	Absolute Position Mode
Bit 3 (VM#_CT_GO)	1	1
Bit 4 (VM#_CT_VMODE)	0 → 1 (velocity mode)	1 → 0 (moves to stop position)

Once Velocity Mode is turned off (Bit 4 = 0), the value in VM#_STOP_POS is written to the project variable VM#_CMD_ABS_POS and the Virtual Master moves to the ABS_POS. The control is now operating in absolute positioning mode.

Relative Move Distance

The Virtual Master moves in increments of this value when the Virtual Master's control register bits are set as follows:

Control Register Bit	State
Bit 3 (VM#_CT_GO)	1
Bit 5 (VM#_CT_RELMODE)	1
Bit 6 (VM#_CT_RELTRIG)	0 → 1 (will move with every transition)

This value is written to project variable VM#_REL_MOVE_DIST when the project is compiled.

Absolute Move Position

Positioning values are embedded in the icons and are stored as project variables when the project is compiled. This value is written to project variable VM#_CMD_ABS_POS. The state of the following bits set the mode of operation to absolute position.

Control Register Bit	State
Bit 3 (VM#_CT_GO)	0 → 1
Bit 4 (VM#_CT_VMODE)	0

When bit 3 is set to 1, the Virtual Master moves to the value in project variable VM#_CMD_ABS_POS. Any change to this value, while in absolute position mode, will cause the Virtual Master to move to the new position.

If the Virtual Master's mode of operation is switched from velocity to absolute position, the value in VM#_CMD_ABS_POS is replaced with the value in project variable VM#_STOP_POS. Only positive values can be used for an absolute position move

Absolute Move Mode

This selection determines the direction that the Virtual Master will use when moving to the *Absolute Move Position* variable. This value is written to project variable VM#_POS_MODE when the project is compiled. The following choices are as follows:

- Positive (0 in *VM#_POS_MODE*)
- Negative (1 in *VM#_POS_MODE*)
- Shortest Path (2 in *VM#_POS_MODE*)

Max. phase difference allowed for shortest path correction

This value (0-180°) is used to create a "shortest path" positioning window for the Virtual Master's positive and negative move mode. When the *Absolute Move Mode* is set to positive or negative, the Virtual Master will move in the specified direction unless the new target position is inside the positioning window. If so, then shortest path will be used. Once the Virtual Master has moved to a new absolute position, a new positioning window is created around the new position. This feature is not available when the *Absolute Move Mode* is set to Shortest Path. Fig. 6-8 illustrates the function of this value.

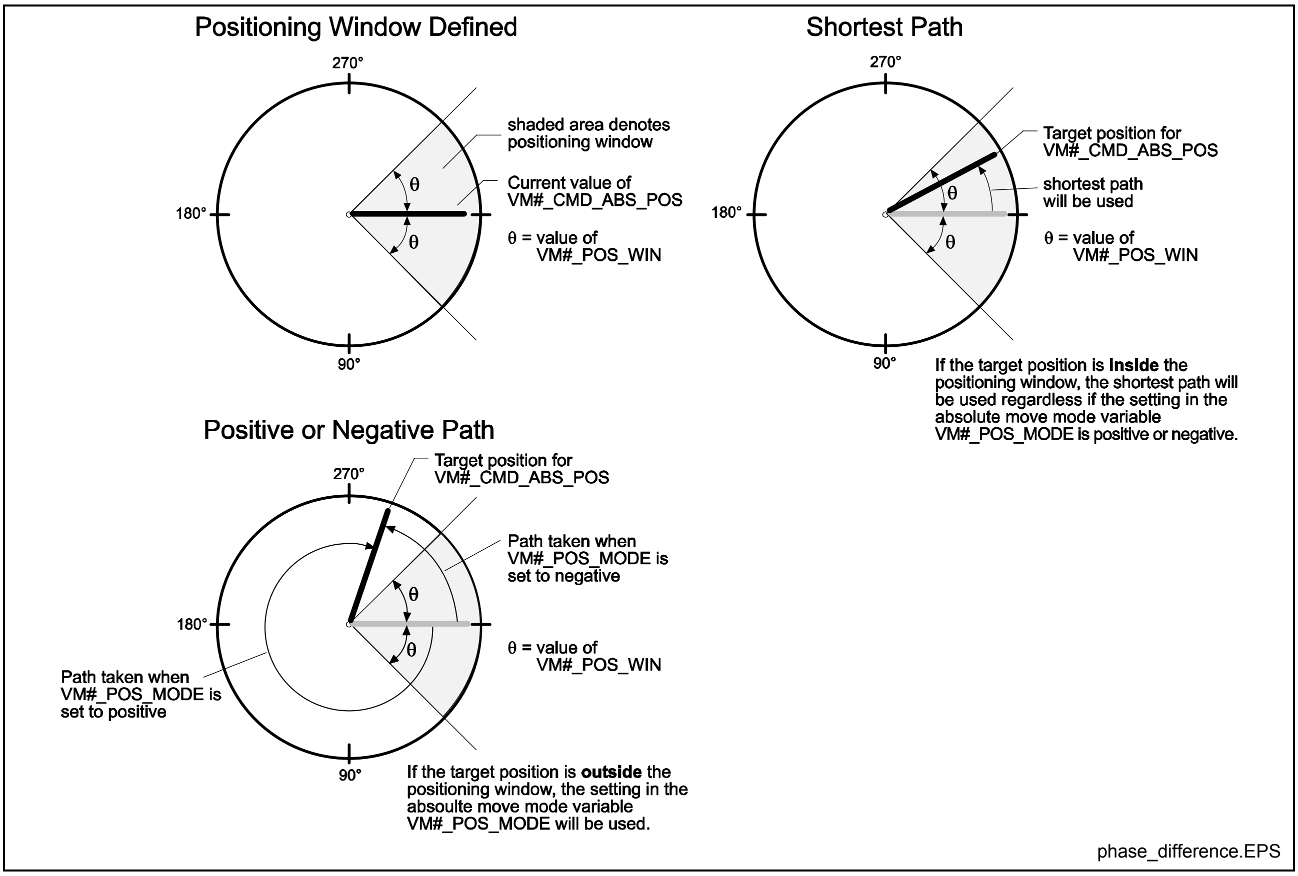


Fig. 6-8: Maximum Phase Difference

Jerk Limiting

This allows you to limit the jerk of a motion, such as vibration caused by acceleration or deceleration.

Virtual Master Modes of Operation

The Virtual Master can operate in two modes, velocity or position mode. The mode is determined by the settings selected in *Virtual Master Runtime Data window (VM1 icon)*. By default, the Virtual Master is set in velocity mode with values in the Initial and Maximum fields for velocity. Position mode requires values in the **Positioning** field of the window. In addition, by selecting the VM1_CT_RELMODE bit, the Virtual Master will switch to relative mode.

Velocity Mode

In Velocity mode, the Virtual Master moves at its commanded velocity. The rate of change in the commanded velocity (VM#_CMD_VEL) is performed using the defined acceleration/deceleration (VM#_CMD_ACCEL, …DECEL) rate. In this mode, the Virtual Master can be either stopped with immediate deceleration or stopped at a designated position between 0 and 360 degrees. Stopping the master at a designated position may take several revolutions (stop ramp) depending on the current velocity and programmed deceleration.

When a master is in velocity mode, an integrator is engaged, providing positional output so that all masters have a uniform signal type (position value with modulo of 360 degrees.)

Position Mode

In Position mode, the Virtual Master moves to a programmed relative or absolute position. With relative positioning, travel distances can be greater than the modulo value for relative positioning moves of the Virtual Master. For absolute positioning, the maximum travel distance is +/- 180 degrees (shortest path) or 359.99 degrees (positive or negative direction) with absolute positioning.

Real Master

A Real Master is an external position signal that is transmitted into the control through the System Master icon. GPP supports a maximum of 6 real masters. The Master Encoder Card (LAG) can be used to accept up to 2 external feedback signals.

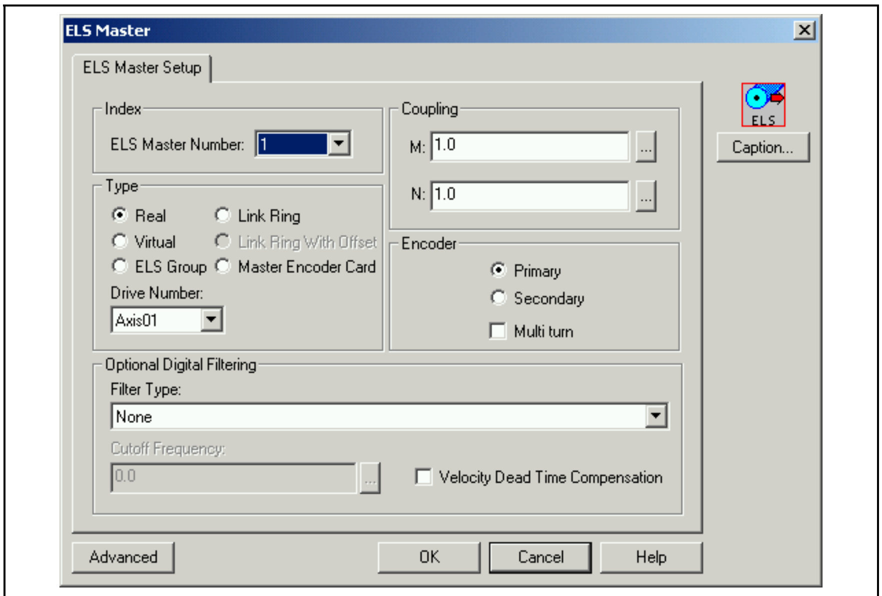
After configuring the Master Encoder Card parameters in the control and setting up the System Master icon, there will be a signal ready to be used. The command stream is based on units of degrees and has a range between 0 and 360 degrees with the module (rollover) fixed at 360 degrees. This output format (0 to 360 degrees) is the basis for the ELS system position where a machine/product cycle is defined as a single revolution. For example, a system following a real master running at 300 RPM produces 300 products per minute.

Initializing a Real Master

The initialization of control and status registers and project variable labels for Real Masters is defined under ***Setup ⇒ Processes ⇒ ELS ⇒ System Masters*** in the Project Navigator window. Refer to ELS System Master on page 6-20 for details.

Declaring Real Master Runtime Data

To declare runtime data for a Real Master in a project, place an ELSMstr icon in the Initialization task and select Real as the master. Refer to the ***ELSMstr icon*** in volume 2, chapter 14, of the *VisualMotion 11 Functional Description* for details.



Setup_ELS_Mstr.tif

Fig. 6-9: Setup ELS System Master: Real Master

Coupling (float) This field contains the edit boxes for the gear ratio. A gear ratio is only applicable to real master signals.

Encoder In this field, you can select primary or secondary encoder and single or Multi-turn encoder type:

- Primary encoder – This is the motor feedback encoder (X4 EcoDrive 03)
- Secondary encoder – this is the extra encoder input at the drive (X8 EcoDrive 03)
- Multi-turn checkbox – this option can be selected if the feedback is a multi-turn absolute device and you want the system to track its absolute position. To reset (home) the current reference value, set the System Master, control register, ELS_M_CT_SET_REF# input bit to 1. This causes the respective system master to instantly move to the position stored in the ELS_MSTR_REF_POS# parameter. To assure the system comprehends the absolute position, monitor the status bit ELS_M_ST_REF#. This bit is cleared every time the parameters controlling the system master are changed or, in the case of an incremental encoder, when the system goes from phase 2 to phase 4.

Filter Type For a real master signal, you can apply several types of filters to dampen noise in the signal. These filters include:

- Low-pass (1st, 2nd, or 3rd)
- Butterworth (2nd or 3rd)
- LP(2nd) with velocity feed forward (Dead Time Compensation)
- LP(3rd) with velocity and acceleration feed forward (Dead Time Compensation)

Cutoff Frequency (float) When a filter type is chosen, a cutoff frequency for the filter must be entered. The cutoff frequency is the frequency where the signal is reduced by 3db. When set to 0 the filter is disabled.

To ensure a stable system, use the following calculation when entering a value for the Digital Filter Cutoff Frequency:

$$\text{Cutoff Frequency} \leq \frac{1}{2 * \text{Sampling Rate(sec.)}}$$

Velocity Dead Time Compensation

Dead time compensation can be activated or disabled by selecting this checkbox. This feature provides the option to apply 4 Sercos cycles of velocity feed forward phase advance to the ELS Master, when activated, to compensate for delays in control processing.

Dead Time Compensation only compensates for the phase lag created by the 4 cycles of ELS processing/Sercos delays, not for additional dead time caused by the various Real Master filters. To address this, some filter types include velocity and/or acceleration feed forward part to compensate for dead time cause by the filter.

The Dead Time Compensation can be disabled for individual Real Masters for applications in which undershoot/overshoot during velocity changes could cause problems. The functionality of the feature is such that compensation is enabled by default. Users will need to disable it if they do not want to use it with their application.

Positioning a Secondary Encoder Signal

A secondary encoder signal can be used as a master axis position. Within the drive, the offset position feedback value 3 (drive parameter P-0-0087) is applied to the secondary encoder value (refer to Fig. 6-10). A gear ratio of only integer values, stored as floats ELS_MSTR_M1…6 and ELS_MSTR_N1…6, is applied to the position feedback if required, for example, using a gear ratio to offset the mechanical gear setting.

A reference position for the position feedback value 3 is set in the float variable ELS_MSTR_REF_POSx (where x is the ELS Master number from 1 to 6). To send the reference position to feedback value 3:

1. Place the ELS Group axis (following the secondary encoder feedback signal) in local mode.
2. Toggle the ELS_M_CT_SET_REFx (where x is a number from 1 to 6) bit of the ELS_MSTR_CONTROL register.
3. Verify the homing sequence with the ELS_M_ST_REFx (where x is a number from 1 to 6) bit of the ELS_MSTR_STATUS register.

An offset is calculated from the comparison of the current feedback position and homing position and is stored in the ELS_MSTR_OFFSETx variable (where x is a number from 1 to 6). The offset is referenced later when the master axis position is set. A filter is added to the feedback position (filter type selected in the Setup ELS System Master window, Fig. 6-9) to smooth the signal. The ELS Master position is stored in the ELS_MSTR_REF_POSx variable (where x is a number between 1 to 6).

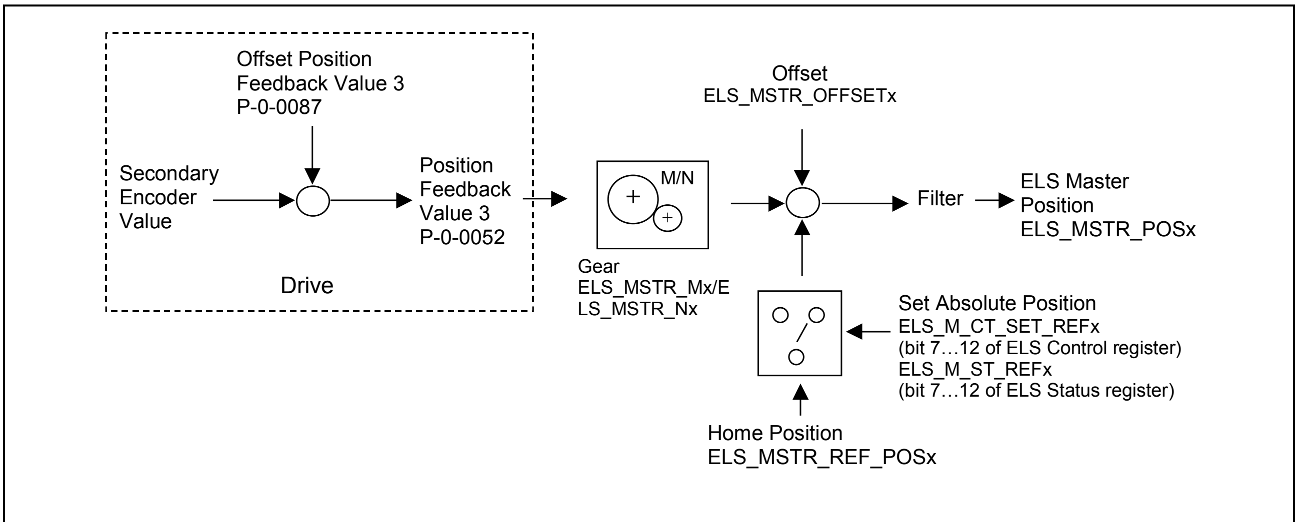


Fig. 6-10: Illustration of Positioning the Real Master Axis

ELS Group Master

An ELS Group signal is the output signal from an ELS Group, which is used by another ELS Group. An ELS Group can output a signal to an ELS Group slave axis, to another ELS Group, or to both. The signal that is output to another ELS Group is assigned a number in the ELS System Master Connection Interface as an ELS Group Master. When assigning the ELS Group Master signals, the concept referred to as cascading is applied to group assignment.

Cascading ELS Groups

A maximum of 6 ELS Group Masters can be cascaded to other ELS Groups. When cascading ELS Groups, the lower numbered group's output should become the higher numbered group's input.

Note: An ELS Group Master's output cannot be fed back into the same ELS Group's input.

If a group number is inputted into a lower group number, then there will be a position delay of one Sercos cycle. This delay exists because the group outputs are processed in numerical sequence and data from the last cycle is used before it can be updated. Depending on the application, this lag may be acceptable.

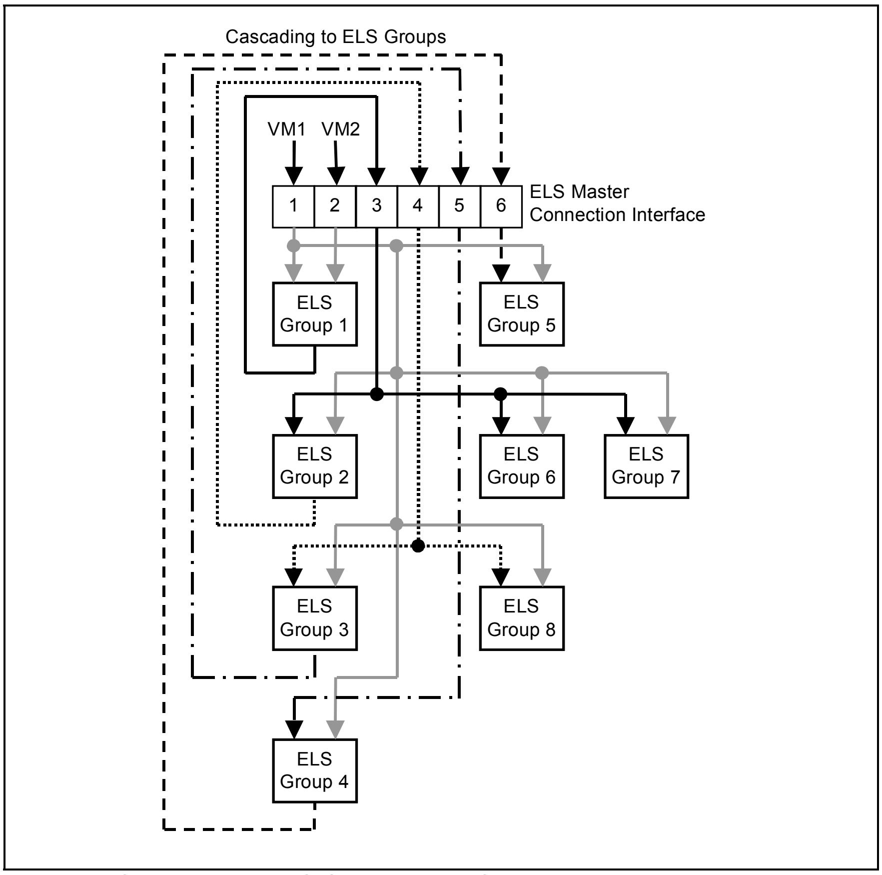


Fig. 6-11: Cascading an ELS Group Master Output

GPP supports a maximum of 6 ELS Group Masters, which are selected in the ELSMstr icon. To configure, assign the ELS master type and the ELS Group Master number (the number indicating which ELS Group Master it is, not its number designation in the ELS Master Connection Interface).

Rexroth
Bosch Group

Link Ring Master

A Link Ring Master is an external position signal sent over a fiber optic ring and received by the control using the DAQ option card. The fiber optic ring can interface with up to 32 controls. GPP will support a maximum of 5 Link Ring Master signals received from other controls in the system and export a maximum of one master signal to the Link Ring for use by other controls. The position signals are based on units of degrees and have a range between 0 and 360 degrees with the module (rollover) fixed at 360 degrees.

The following items are factors when using a Link Ring Master:

- The minimum Link Ring cycle time is scalable based on the maximum number of PPC controls configured in the Link Ring system. Refer to *Configuring a Link Ring for ELS System* on page 6-63 for details.
- The I/O system still runs at the 2/4 ms setting, but Sercos based I/O are delayed accordingly.
- If all controls are to be synchronized to the same master signal, the control that generates the signal must also receive that same signal and use it for commanding its axes. If the axes directly follow the source signal, they will be two cycles (16 ms) ahead of the other controls.
- If you are using a Real Master encoder physically connected to the machine and using Link Ring, there will be two cycles (16 ms) of delay between the actual position and the Link Ring Master position. Depending on the application, this lag may not be acceptable.

Link Ring Masters are defined in a project through the *ELS Master Setup* window.

Master Encoder Card

The Master Encoder Card (LAG) communicates over the control's PC-104 ISA bus and provides real master input signals to the ELS System. It supports EnDat or 1 Vpp Sinusoidal encoder signals. Real master gear ratio (Coupling), filtering, and velocity dead time compensation can be set from the *ELS Master Setup* window by selecting ***Setup ⇒ Processes ⇒ ELS ⇒ System Master*** from the project navigator.

The following items are factors to consider when using a Master Encoder Card:

- Only one Master Encoder Card can be installed in the control.
- It is not intended to provide secondary feedback signals for a drive that supports only one primary feedback.
- Only rotary scaling in degrees (one revolution = 2^{20}) with a modulo of 360.

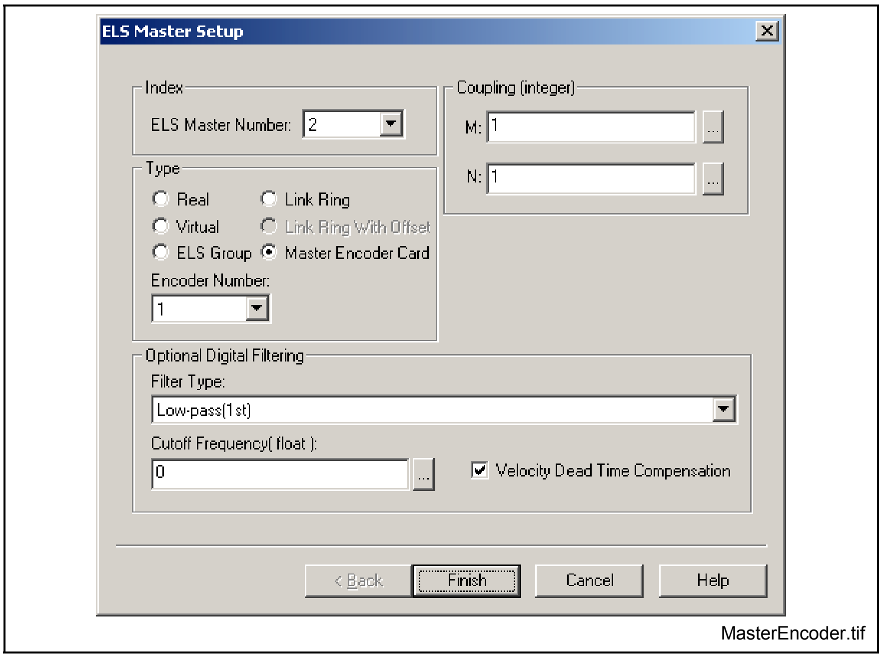
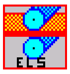


Fig. 6-12: Master Encoder Card System Master

ELS Group



An ELS Group is a container that allows you to sectionalize a machine or process by grouping axes working in unison to follow a master command signal. Up to 8 ELS Groups can be initialized under ***Setup ⇒ Processes ⇒ ELS ⇒ ELS Groups***.

Declaring ELS Group Runtime Data

ELS Group runtime data is configured using the *ELSGrp3* icon. To reference groups in the system, they are given an index from 1 to 8. Each is assigned a set of control and status parameters and registers for configuring and controlling its functionality. Basic functions include; selecting a master signal to follow or internal profile, dynamically synchronizing to a master signal, utilizing a CAM profile, applying phase offsets, and activating lock on/off function and jogging. The output units are degrees with a range between 0 and 360 degrees and the module (rollover period) fixed at 360 degrees.

When using an ELS Group to control an axis, the following consideration should be noted:

- An axis can only be assigned to a single group. The system detects multiple assignments during the compile process and flags the errors.
- By default, an axis can be positioned independently of the group for setup or other requirements using Single axis positioning or Velocity mode.
- Every time the system phases up (phase 2 to phase 4), each axis in the group defaults to a secondary mode of operation (single axis positioning or velocity mode). You must use the ELS Mode Change icon in the program to command the axis to follow the group master position.

An ELS Group's output provides a master position to its assigned ELS Slave axes. ELS Slave axes can only be assigned to an ELS Group at compile time. The ELS Group Master's output position is derived from the currently active group master input. The ELS Group's output signal can be modified using the following options:

- M/N gear ratio
- GMP (Group Master phase offset)

- CAM Profile with/without a Lock On/Lock Off feature using a 3 CAM profile
- GSP (Group Master phase offset)

An ELS Group can only have one active master at any given time determined by the group's control register input bit (G#_CT_MSTR_SEL.) To stop or move a group's master, independent from the two input masters, every ELS Group has it's own Group Local Virtual Master, which has the same functionality as the system Virtual Masters. To synchronize to the Group Local Virtual Master, the group has to be switched to local mode (G#_CT_LOCAL). When the project's task A is in manual mode, the groups are stopped before switching into local mode and synchronizing with the Group Local Virtual Master.

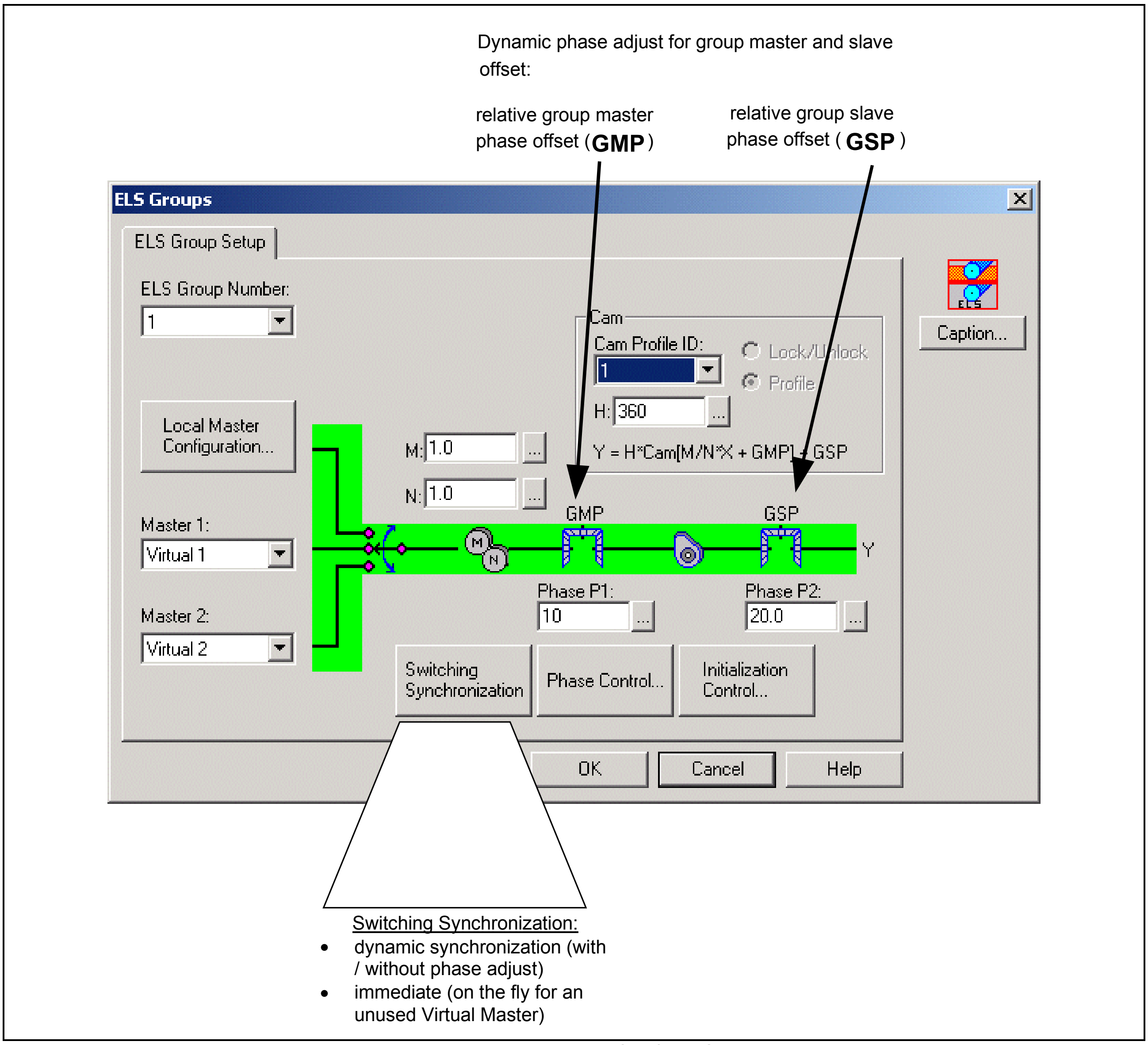


Fig. 6-13: Electronic Line Shafting Group

The ELS Group's active master input signal "X" is a condition of the equation in Fig. 6-14:

$$Y = H * CAM[M / N * X + GMP] + GSP$$

Fig. 6-14: ELS Group Output Equation

Where CAM[ ] is a control CAM profile table or index CAM profile, M and N is the current master input/output ratio, H is a CAM profile scale factor and GMP and GSP are group master and slave relative phase adjusts. The signal Y drives the group's slave axes (group master position).

Note: When CAM Lock/Unlock is selected, modifications to the M/N ratio or phases P1 and P2, while the program is in phase 4, will not take affect until a transition from phase 2 to phase 4 is made. When CAM Profile is selected, modifications to the M/N ratio or phases P1 or P2, while the program is in phase 4, will take effect immediately.

ELS Group Axis Setup

ELS Group axes are configured under ***Setup*** ⇒ ***Axes***. Refer to *ELS Group Axis Setup* in section 4.3, *Setup Axes*, for details.

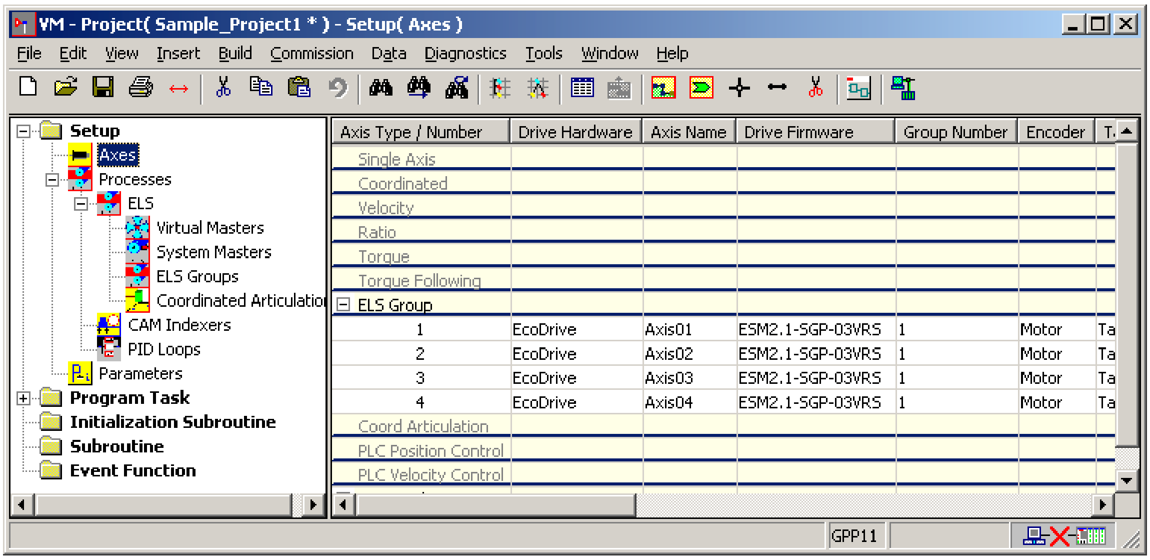


Fig. 6-15: ELS Axis Configuration

Rexroth Bosch Group

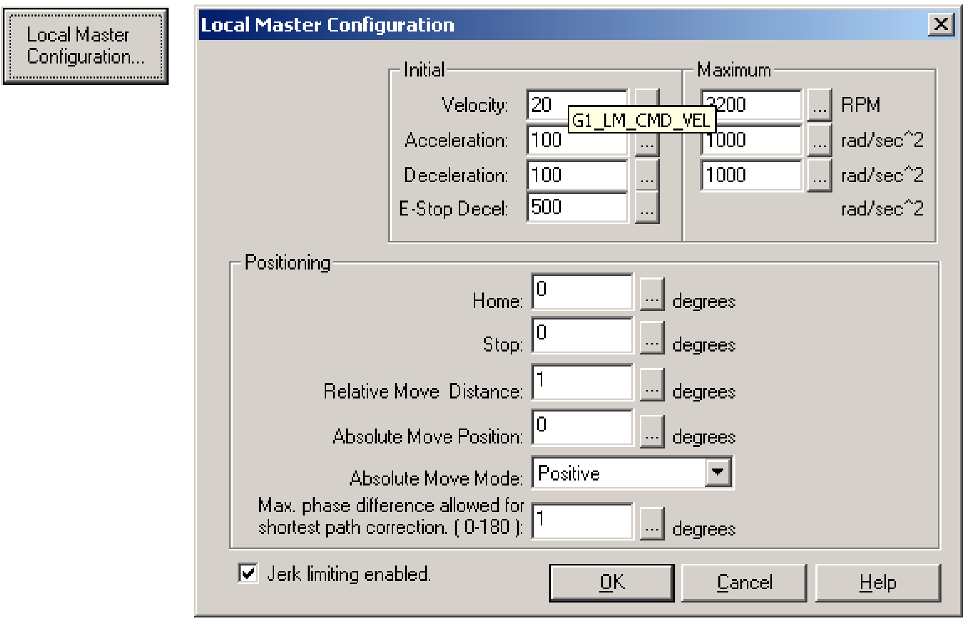
Local Master Configuration

ELS Groups support a Local Master functionality that allows the Group to be switched from its current system input master to its own Virtual Master. It can be used as a Virtual Master or used to independently move the ELS Group in velocity mode or position mode (relative or absolute).

The switch to the Local Master is performed as immediate "On the Fly". Dynamic synchronization, Immediate or Advanced Trigger options are not supported when switching to the Local Master. When exiting the Local Master to an active system master, the ELS Group's current Synchronization switching type is used.

When an ELS Group is first configured, default initial and maximum values are set for the Group's Local Master. These values can be written to by the program or modified within the ELS Group icon by clicking on the **Local Master Configuration** button.

Note: Placing the cursor over a field will display the associated program variable label in a tool tip. Refer to Table 6-18 and Table 6-19 for a listing of the default float and integer numbers assigned to each ELS Group program variable label.



local_master_config.tif

Fig. 6-16: Local Master Configuration

The Local Master is controlled by the same control register, status register and program variables assigned to the ELS Group. Refer to the following tables for a complete listing of ELS Group register bits and program variables:

- Control Register Bits Table 6-16, page 6-14
- Status Register Bits Table 6-17, page 6-15
- Program variables Table 6-18, page 6-16 and Table 6-19, page 6-17

Switching to the Local Master

When switching an ELS Group from it's current input master to the Local Master, the switch is performed immediate "On the Fly" for either velocity or position modes. This means that the Local Master is initialized with the current velocity and position of the input master at the instant the switch is performed. Refer to Fig. 6-21: Immediate “On the Fly“ Switching of ELS Group 1 on page 6-44 for details.

When exiting the Local Master, the ELS Group's configured Synchronization switching type (Dynamic synchronization, Immediate or Advanced Trigger options) is used.

Local Master Velocity Mode

An ELS Group is switched to the Local Master running in velocity mode when the following control register bits are set:

Control Reg-Bit	Label	Description
Bit 6	G#_CT_LOCAL	transition from 0 ⇒ 1
Bit 7	G#_CT_LM_FSTOP	set to 0
Bit 9	G#_CT_LM_GO	set to 1
Bit 10	G#_CT_LM_VMODE	set to 1 (Velocity mode)

Table 6-24: Local Master Velocity Mode

When bit 6 (G#_LM_LOCAL) is transitioned from 0 to 1, the Group's current input master velocity and position are copied to the Local Master's current velocity (G#_LM_CUR_VEL) and position (G#_LM_CUR_POS).

Note: The value in program variable G#_LM_CMD_VEL is always overwritten by the Group's current input master velocity when the switch is performed.

Setting bit 9 (G#_CT_LM_GO) to 0 stops the Local Master. To increase or decrease the velocity of the Local Master, change the value in program variable G#_LM_CMD_VEL.

Local Master Position Mode

An ELS Group is switched to the Local Master running in position mode when the following control register bits are set:

Control Reg-Bit	Label	Description
Bit 6	G#_CT_LOCAL	transition from 0 ⇒ 1
Bit 7	G#_CT_LM_FSTOP	set to 0
Bit 9	G#_CT_LM_GO	set to 1
Bit 10	G#_CT_LM_VMODE	set to 0 (Position mode)
Bit 13	G#_CT_LM_RELMODE	0 = absolute mode, 1 = relative mode
Bit 14	G#_CT_LM_RELTRIG	0 ⇒ 1 triggers a relative move when G#_CT_LM_RELMODE = 1

Table 6-25: Local Master Position Mode

When bit 6 (G#_LM_LOCAL) is transitioned from 0 to 1, the Group's current input master velocity and position are copied to the Local Master's current velocity (G#_LM_CUR_VEL) and position (G#_LM_CUR_POS) and then the following behavior is observed:

- If bit 13 (G#_CT_LM_RELMODE) is set to 0 (Absolute Mode), The Local Master moves to the absolute position value in program variable G#_LM_CMD_ABS_POS and stops. Modifying the value in G#_LM_CMD_ABS_POS will cause the Local Master to move to the new position.
- If bit 13 (G#_CT_LM_RELMODE) is set to 1 (Relative Mode), the Local Master decelerates to a stop using the value in program variable G#_LM_CMD_DECEL. Toggling bit 14 (G#_CT_LM_RELTRIG) causes the Local Master to make a relative move equal to the value in G#_LM_REL_MOVE_DIST.

Switching Synchronization between Group Input Masters

Switching to either ELS Group input master 1 or 2 could be performed using one of the following methods:

- **Immediate Switching** - The group's output position and velocity immediately switches, within one Sercos cycle, between the two masters causing a step or bump in transition. When switching to an inactive Virtual Master, the position and velocity are adjusted “on the fly.”
- **Dynamic Synchronization with or without Phase Adjust** - The velocity and position difference will be compensated for by an internal ramp function synchronizing the transition between masters.

Switching to the Group Local Virtual Master occurs when the ELS Group is switched to local mode, regardless of which input master the group is currently following (refer to Table 6-26). This transition to the Group Local Virtual Master can only be done with immediate “On the Fly” switching. Deactivation of the local mode will cause dynamic synchronization/immediate switching to the active group input master.

Master Selected	G#_CT_MSTR_SEL	G#_CT_LOCAL
Group Master 1	0	0
Group Master 2	1	0
Group Local Master	X	1

Table 6-26: Bit Settings for ELS Group Master Selection

In the following example, ELS Group 1 with master input 1 is designed to synchronize to master input 2 when 2 is selected as the master input of ELS Group 1. In effect, master input 2 acquires the slave axes of ELS Group 1, thus replacing 1.

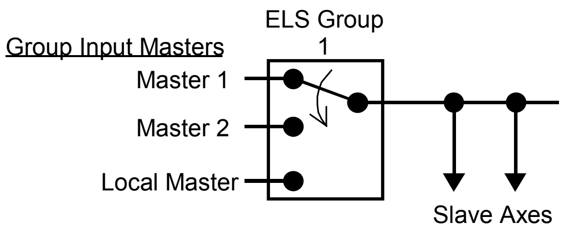
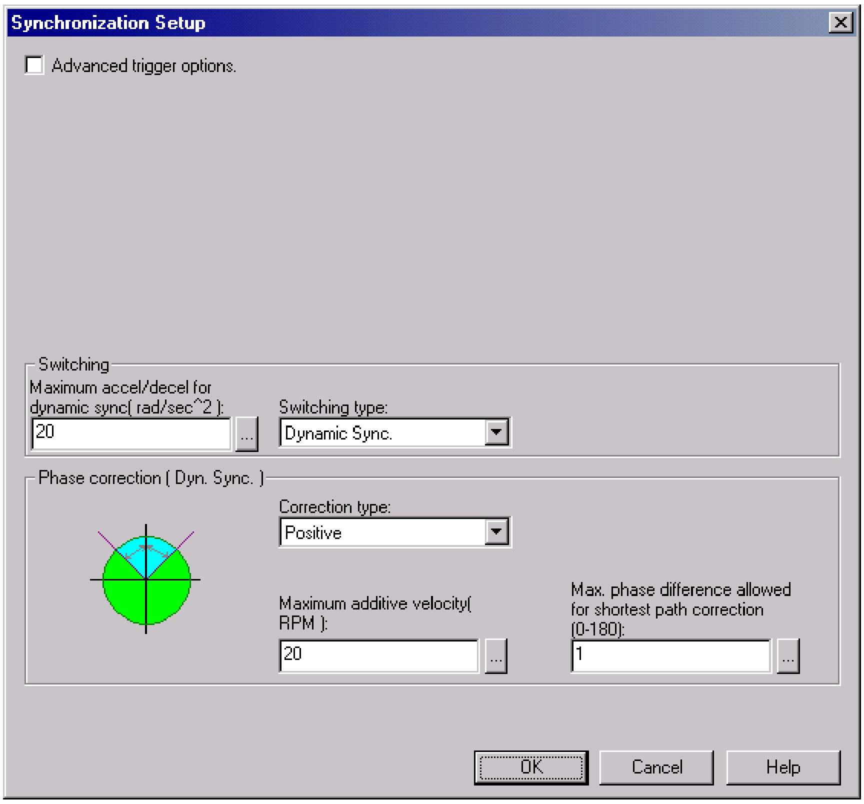


Fig. 6-17: Group Input Master Switching

When the *Switching Synchronization* button is selected, the *Synchronization Setup* window in Fig. 6-18 displays the default values generated by the ELS Group's project variables.



Synch_Setup.tif

Fig. 6-18: Synchronization Setup Window

Switching

Maximum accel/decel for dynamic sync.

This value is the maximum acceleration or deceleration that the ELS Group will use to ramp to the new master's velocity and perform any phase corrections with a trapezoidal velocity profile.

Note: The maximum acceleration and deceleration value is only used for dynamic synchronization.

Switching type:

This selection determines the method for switching between ELS Group input masters, immediate switching or dynamic synchronization.

Note: Phase correction is only available for dynamic synchronization.

Rexroth Bosch Group

Phase Correction

Correction type

Specifies the direction in which the phase correction will be made. The user can select **shortest path**, **positive**, **negative** or **no phase correction**.

Maximum additive velocity

Specifies the maximum increase or decrease in velocity allowed for matching the phase (position) of a target master.

Maximum phase difference allowed for shortest path correction

When selecting a positive or negative direction, the value (± 0-180 degrees) entered creates a range (monitoring window) around the position of the target master. If any phase errors are within this window, the shortest path will be used for the correction. This allows large phase corrections to be eliminated depending on the size of the window.

Enable CAM profiling; disable lock on/lock off function

This must be selected to activate the CAM profile ID settings. If you change this selection during operation, you must go in and out of Parameter mode to activate the change in selection. Clocking will not activate this selection change.

Advanced…

The ***Advanced…*** button opens the *Lock / Unlock CAM Advanced Setup* window.

Note: The ***Advanced…*** button is only available in the ELS Group Runtime Data window if the ELS Group was initially setup to use the Lock / Unlock CAM option in the ELS Group Setup window. Refer to *ELS Group Initialization* in section 4.5, *Setup ELS Process*, for details.

This window displays the default CAM numbers used for the ELS Lock On / Lock OFF function. The user can modify the default settings with CAM numbers and H factors that have been designed for their specific application. Refer to *Synchronized "Lock On / Lock Off" of ELS Group Master* on page 6-53 for a description of this feature.

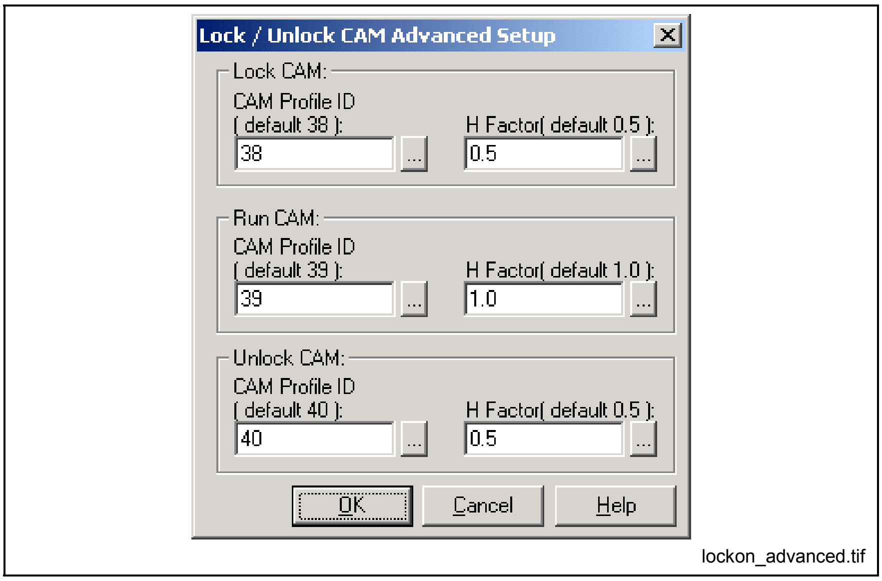


Fig. 6-19: Lock / Unlock CAM Advanced Setup

Immediate Switching

This method allows for an immediate transition to a new input master. Switching takes place within one Sercos cycle without regard to bumpless transitions. When switching between input masters, the group's velocity and position immediately change to match the target master. The bump is caused by the sudden change in velocity and correction of position difference between input masters. The following graph shows a typical immediate switch with a transitional bump.

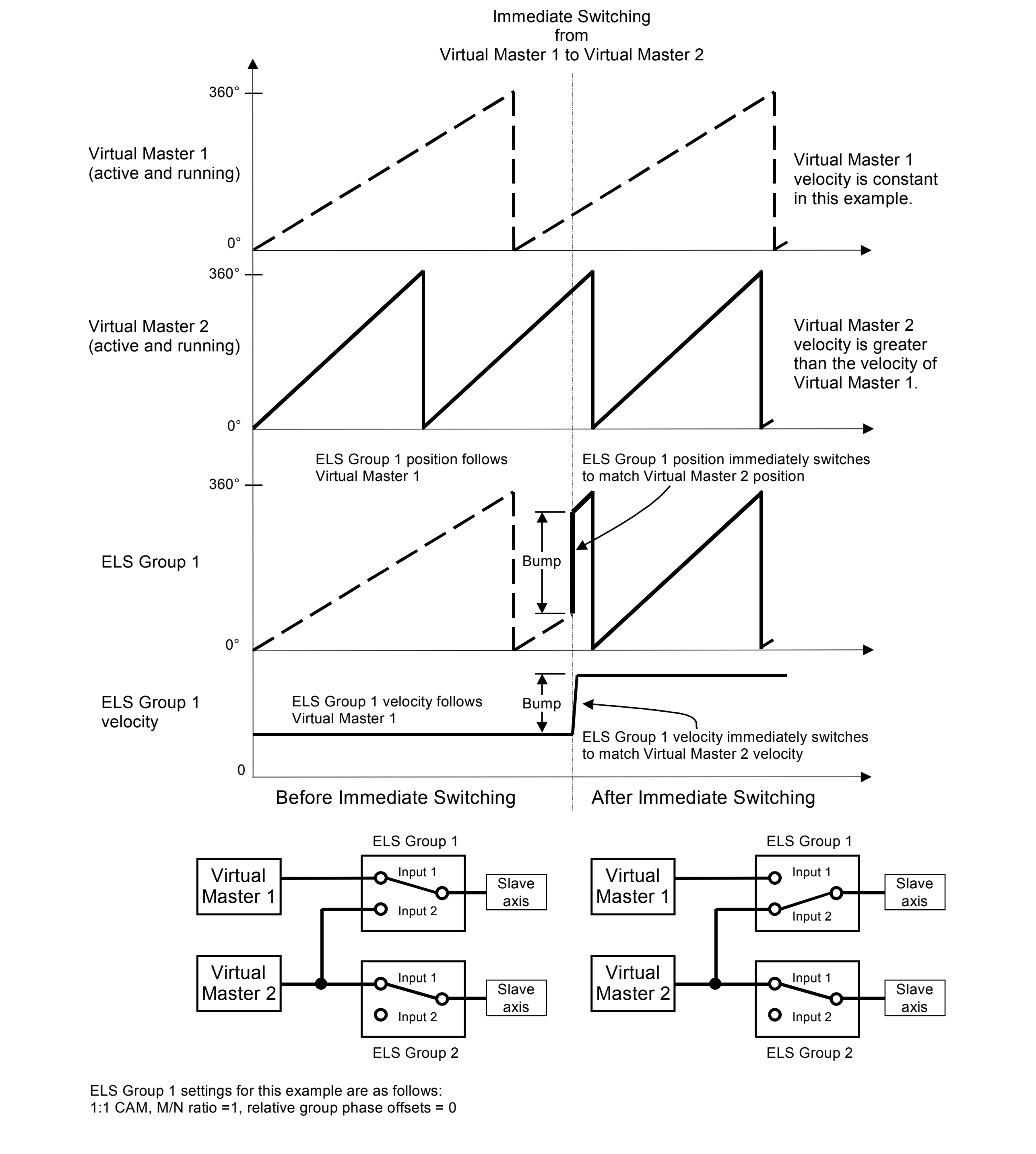


Fig. 6-20: Immediate Switching of ELS Group 1

Immediate Switching to an Inactive Virtual Master (also known as “On the Fly” Switching)

A special case of immediate transition called “On the Fly” applies only to Virtual Masters that are not connected to any groups. On the Fly is an immediate, bumpless transition achieved by initializing the Virtual Master’s position and velocity with a sampling of the group’s current position and velocity. This transition is only possible when the following conditions exist:

- The Fast Stop bit (VM#_CT_FSTOP) is not active
- The Virtual Master is in velocity mode (VM#_CT_VMODE bit is high) (G#_CT_LM_VMODE for Local Master)
- The Go bit (VM#_CT_GO) is high (G#_CT_LM_GO for Local Master)

Note: It is not possible to use this method to synchronize to a Real Master since instantaneous changes in position, velocity, or acceleration would result in a drive fault.

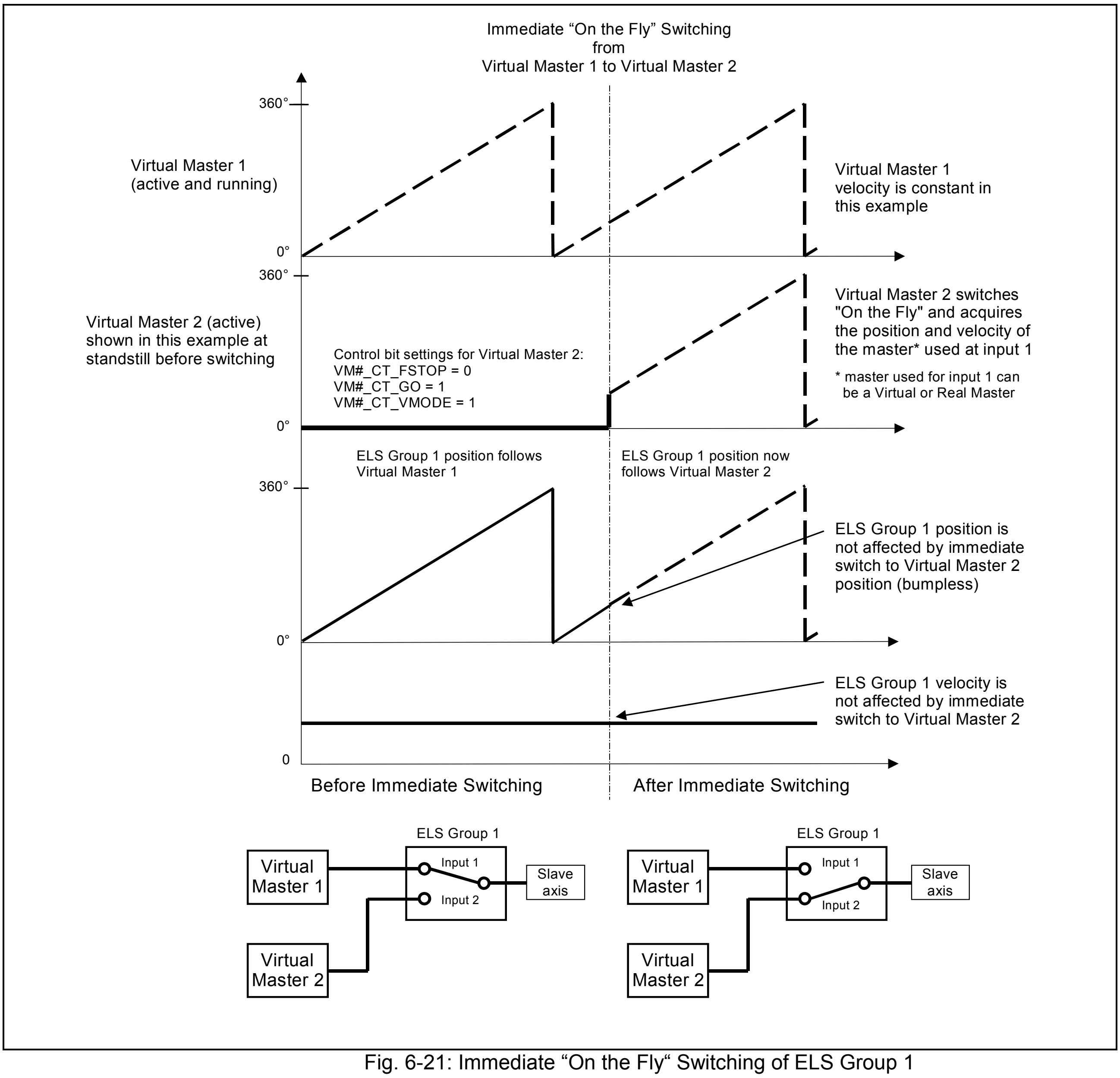


Fig. 6-21: Immediate “On the Fly“ Switching of ELS Group 1

Dynamic Synchronization

Typically, this general-purpose method synchronizes an ELS Group with a real, virtual or ELS Group Master to another real, virtual or ELS Group Master.

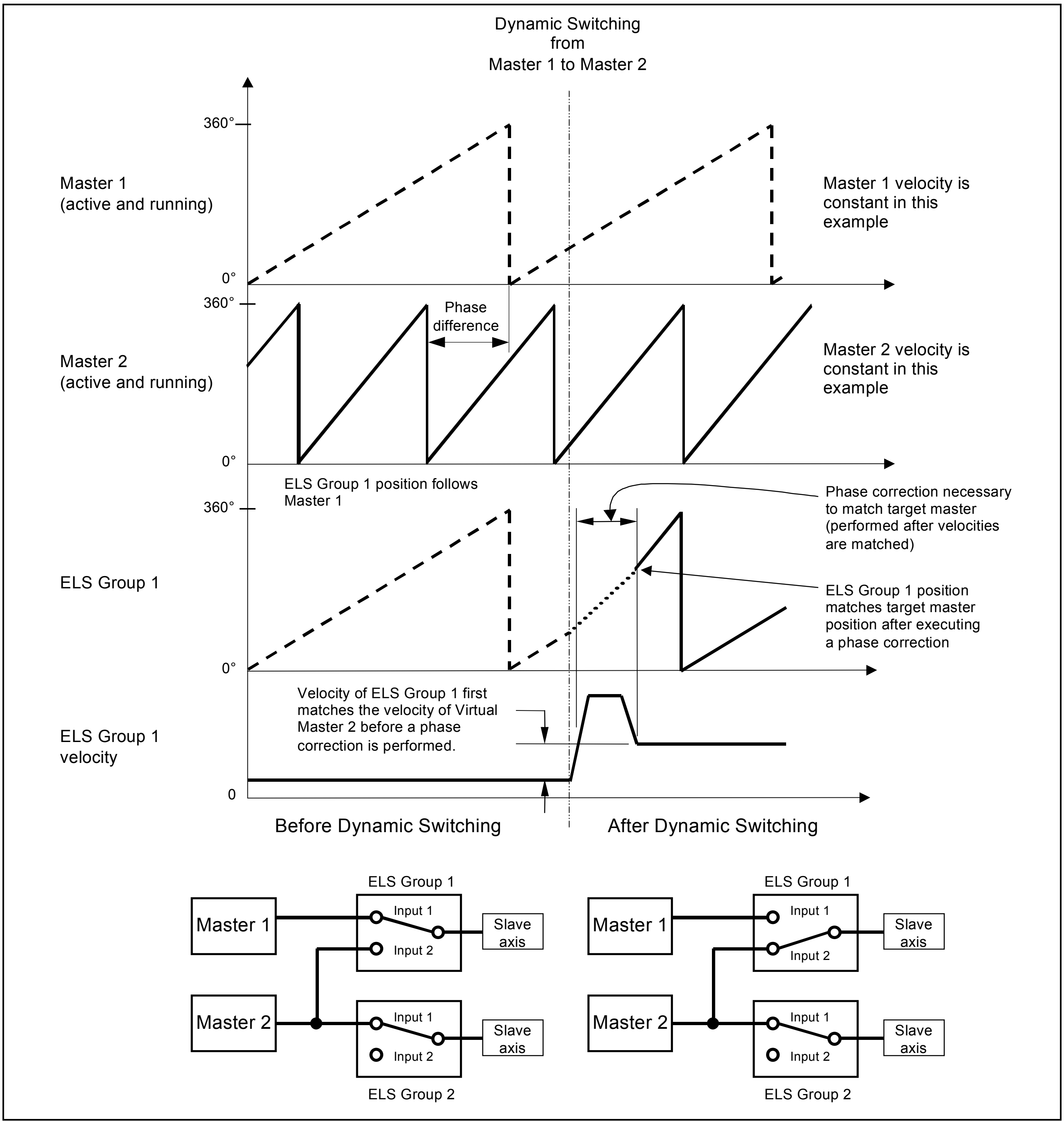


Fig. 6-22: Dynamic Switching of ELS Group 1

Dynamic synchronization allows for a rapid switch to a temporary (internal) velocity mode Virtual Master which then ramps and locks onto the new target master. The temporary master immediately disconnects the group’s connection to the first master and allows for a smooth transition to the next master. Ramping is performed using the ELS Group synchronization's acceleration and velocity rates. These rates are also used for dynamic group master and slave phase corrections.

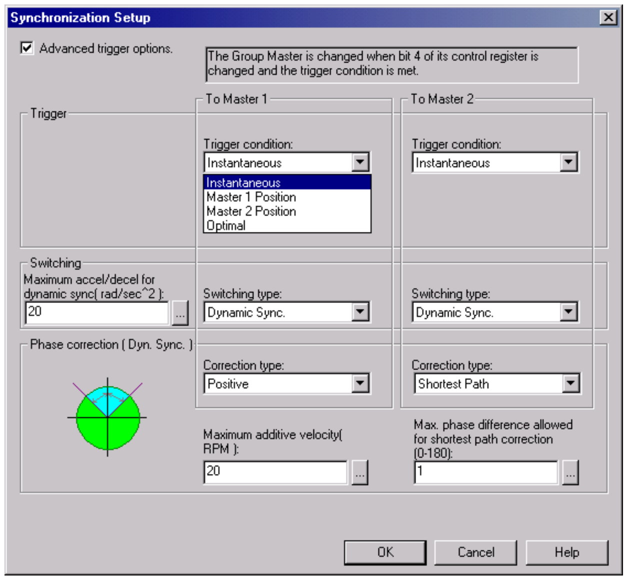
After ramping (velocity synchronization), a phase adjust compensates for any position error. Once the phase adjust is complete, the ELS Group’s master is switched to the new input master. The temporary, internal master is dissolved after the transition is complete.

Rexroth
Bosch Group

Note: Any attempt to switch masters again during dynamic synchronization is ignored, with the exception of switching to local mode (stop ramp).

Switching Synchronization with Advanced Trigger Options

Switching Synchronization between masters can be controlled to a greater extent with the **Advanced trigger options** feature in the *Synchronization Setup* window, see Fig. 6-23. The advanced trigger feature includes settings to specify when switching occurs. It sets up a two-step process for switching. The first step is the Switching Trigger which determines exactly when the switching is to start (Advanced trigger feature) and the second step is the Switching Type (standard switching feature) which determines exactly how the switching is performed.



Synch_Setup_Adv.tif

Fig. 6-23: Synchronization Setup with Advanced Trigger Options

Four types of switching triggers can be used for group switching to the Master 1 or Master 2 signal:

- **Instantaneous** – Disables the switching trigger functionality.
- **Master 1 Position** – Starts switching when group master 1 has passed a specific trigger position.
- **Master 2 Position** – Starts switching when group master 2 has passed a specific trigger position.
- **Optimal** – Starts switching when a certain relationship between the group masters occurs, for example, when the group's master positions coincide or when the group's constant acceleration ramp results in synchronization to the target master's position and velocity simultaneously.

Note: The position trigger used is independent of which master the group is switched to. For example, the Master 1 position switching trigger can be used when switching to group master 1 or 2.

The following Diagrams illustrate the group master switching examples with triggering mechanisms. These methods are used in VisualMotion 11 software. Switching triggers make it possible to start the actual master switching process when a certain condition is met. The master 1 position and master 2 position switching triggers start the switching process when one of the group masters has met a specific trigger condition.

Immediate Switching Using the Master 1 and 2 Switching Triggers

The immediate switching trigger disables the switching trigger functionality so the master switch will begin as soon as the group's control register bit is transitioned.

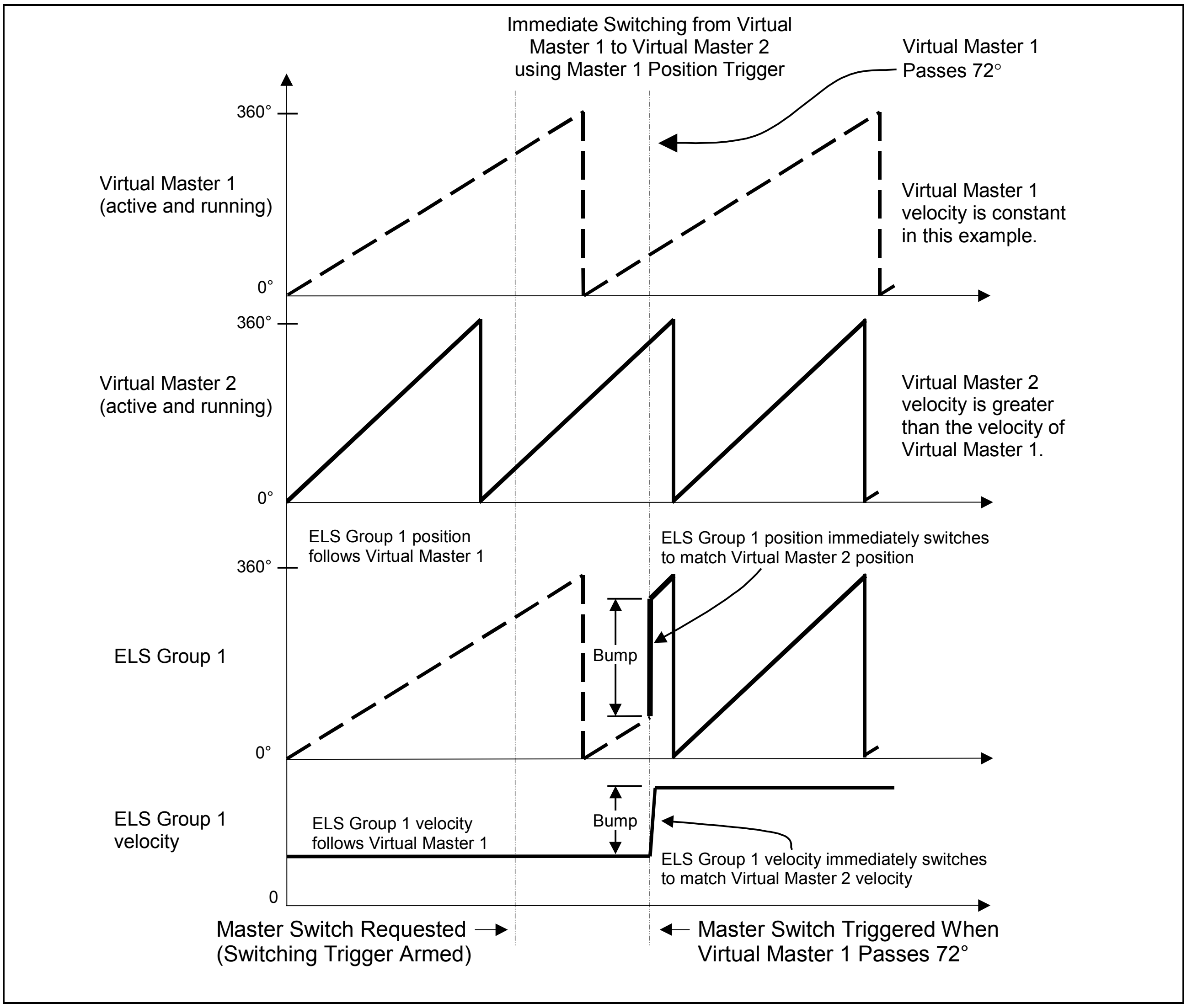


Fig. 6-24: Immediate Switching Using the Master 1 and 2 Switching Triggers

Rexroth Bosch Group

Immediate Switching using Optimal Switching Trigger

This master switch is triggered when the master positions are equal. As soon as the two masters reach the same position, the group matches the velocity of the target master. Switching is dependent on how long it takes until the masters to reach the same position.

Note: If the velocity of the master is equal to the velocity of the group, switching will not occur until the master 1 velocity changes.

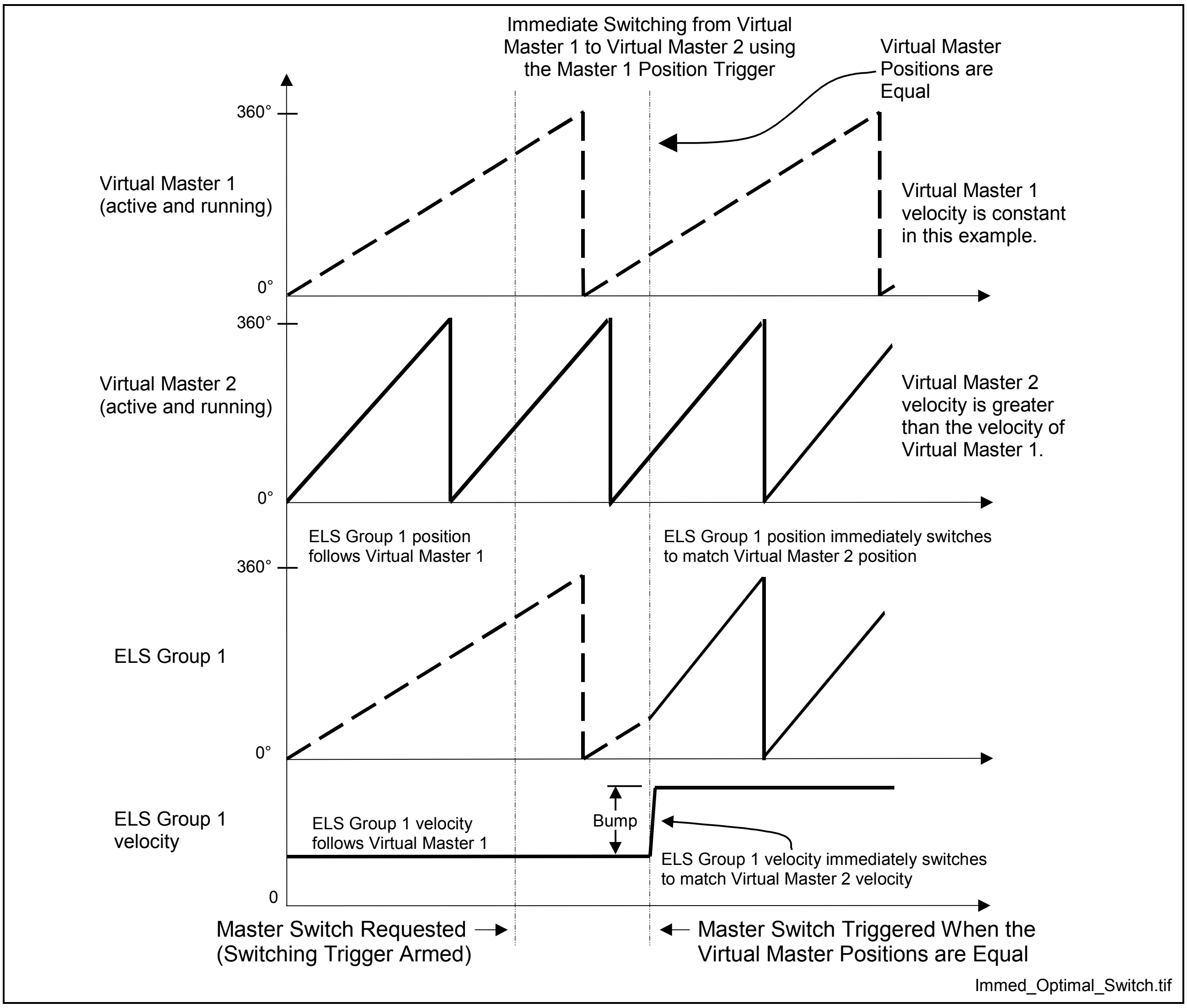


Fig. 6-25: Immediate Switching using the Optimal Switching Trigger

Immediate “On the Fly” Switching using Master 1 and 2 Switching Trigger

Master 2 is a virtual master that is active, but in standby mode until the master switch is triggered. Once the switch is triggered, master 2 switches on the fly and acquires the position and velocity of master 1. The ELS group switches from master 1 to master 2 when master 2 is at the same position and velocity as master 1, producing a bumpless group transition.

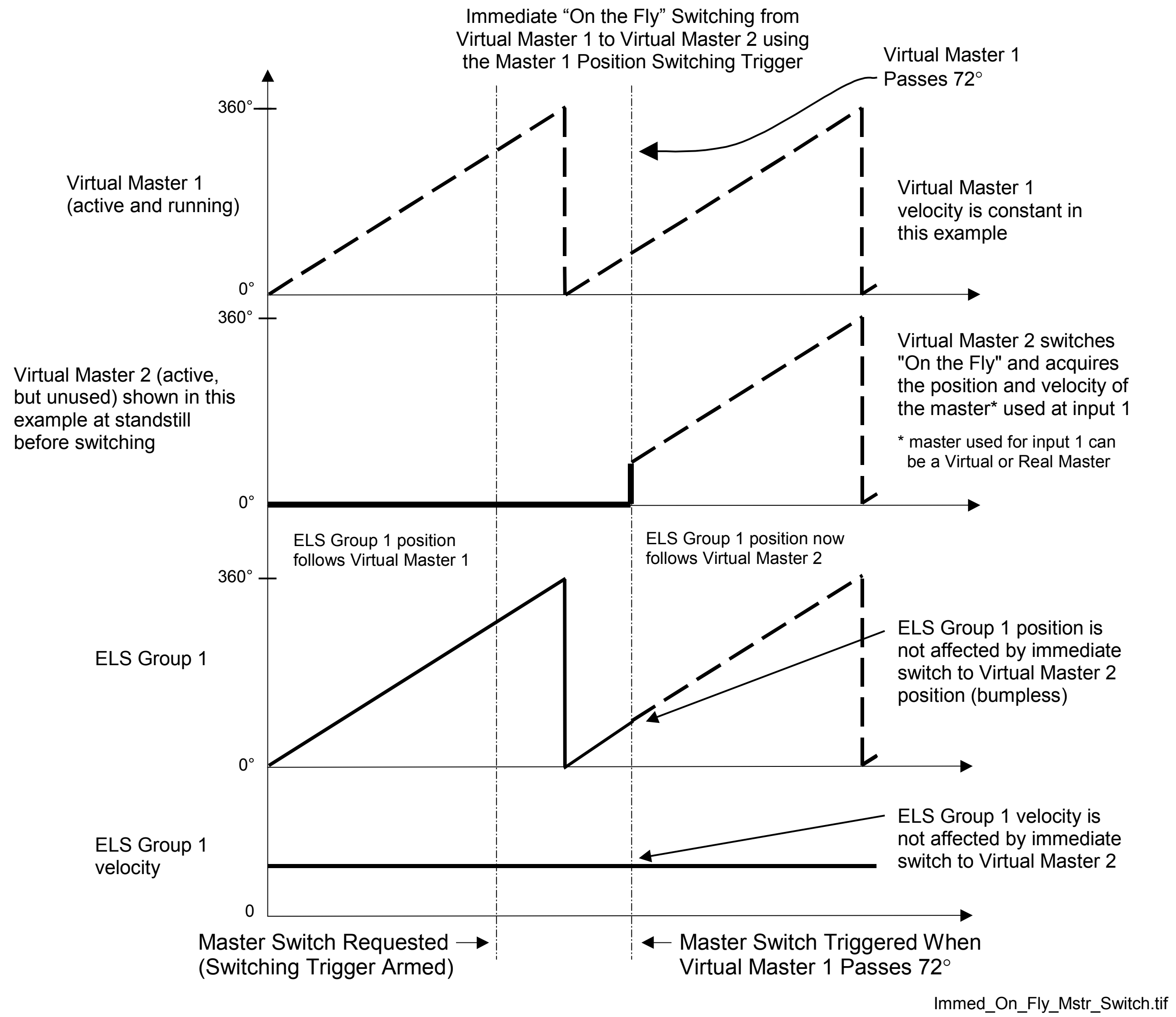


Fig. 6-26: Master 1 and 2 Immediate “on the Fly” Switching

Immediate “On the Fly” Switching using the Optimal Switching Trigger

Master 2 is an active virtual master in standstill mode. When triggered by the master switch, Master 2 acquires the velocity and position of master 1 when master 1 reaches the standstill position of master 2. The switching point of the masters is also the point at which the ELS Group switches from master 1 to master 2.

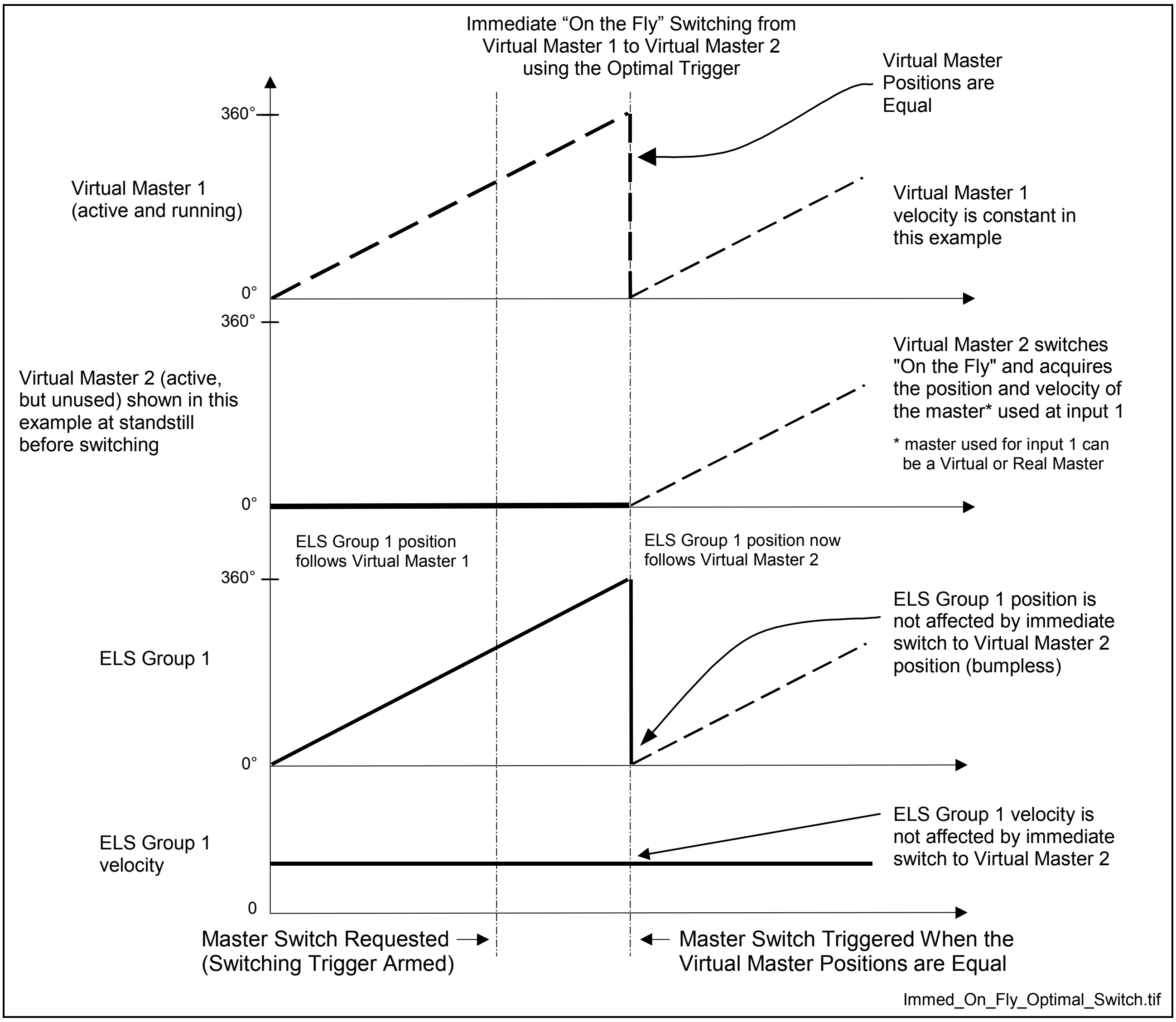


Fig. 6-27: Immediate “On the Fly” Switching using the Optimal Switching Trigger

Dynamic Switching using Master 1 and 2 Switching Trigger

Both master 1 and master 2 remain at a constant velocity. The ELS Group follows master 1 until the master switch triggers the group to switch to master 2. When the group is triggered, it ramps speed, either accelerating or decelerating until it reaches the target position at which point it will be close to the velocity of master 2. At the point of switching, the group will experience a phase adjustment to match master 2.

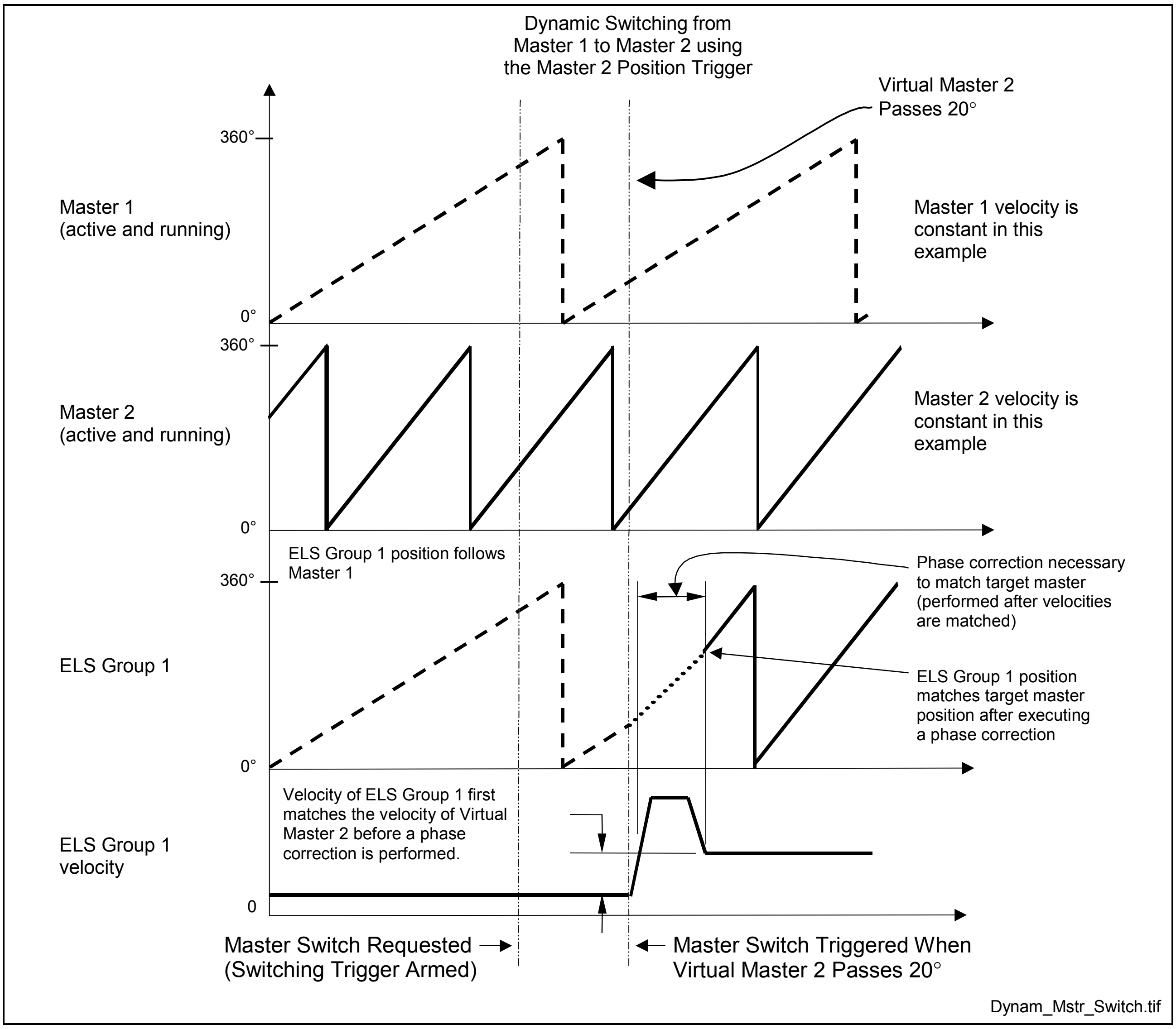


Fig. 6-28: Dynamic Switching using Master 1 and 2 Switching Trigger

Dynamic Switching using Optimal Trigger

Master 1 and 2 maintain their velocities and positions. The master switch triggers the ELS group to switch to target master. The project calculates when the group should begin changing velocity to match the velocity of master 2. When the ELS group's constant acceleration ramp will result in a merge with the target master's position and velocity, the master switch triggers ELS group to switch to the target master. After the ELS group has switched to the target master, a slight phase correction, of up to two Sercos cycles of the closing velocity of the two masters may be necessary to exactly match the target master.

Note: Changes in the velocity of either group master during Optimized Dynamic Synchronization

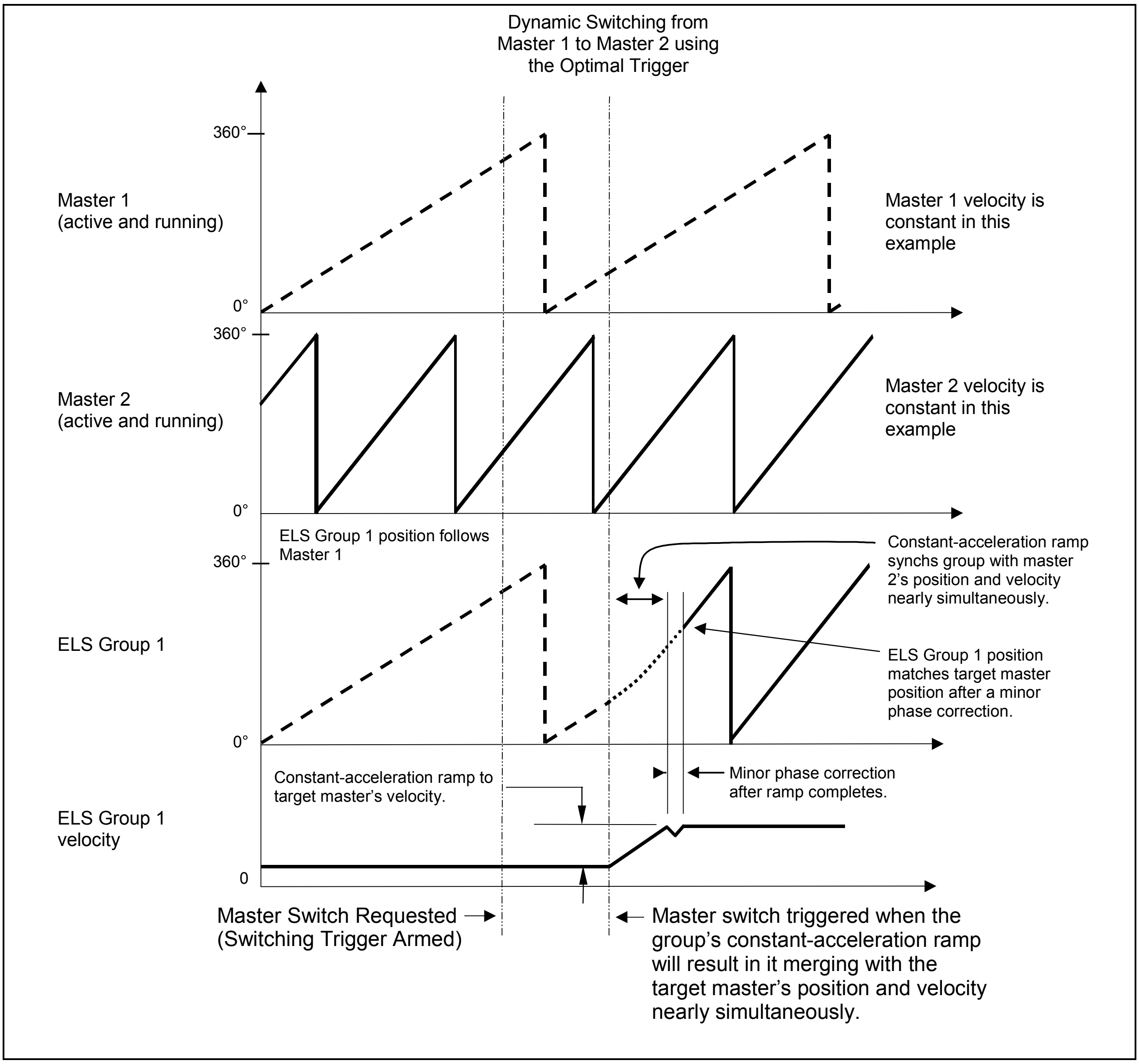


Fig. 6-29: Dynamic Switching Using Optimal Trigger

Synchronized "Lock On/Lock Off" of ELS Group Master

VisualMotion using GPP firmware can be used to stop and restart an ELS Group for one or more cycles of the group's input master. This function is performed using three CAM profiles that are synchronized with the group's input master. Synchronization between CAM profiles and master input eliminates the need for phase corrections. This function also allows the project to stop a group's process without disrupting the other groups running in the project.

The following is an application example of the Lock On/Lock Off feature in GPP firmware. This example monitors the presence of a gap between products in a horizontal wrapper.

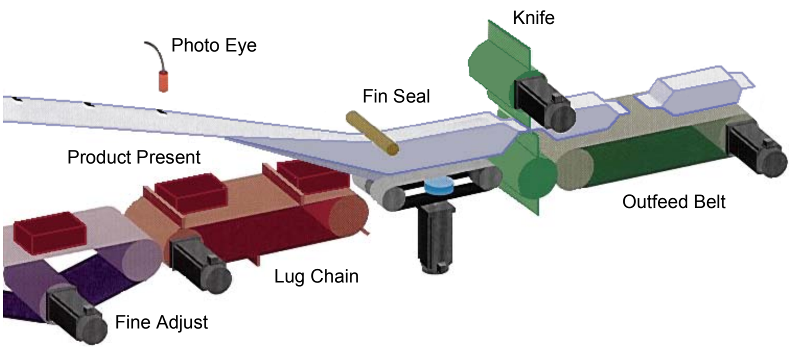


Fig. 6-30: Horizontal Form, Fill and Seal Wrapper

The Lock On/Lock Off feature in GPP is activated by the state condition of bit 1 (G#_CT_LOCK_OFF) in the ELS Group control register.

VisualMotion has three default CAM profiles for the Lock On/Lock Off feature. However, you can create and download customized CAM profiles using the CAM builder function in VisualMotion.

One-to-One CAM Profile "Synchronized to Master"

The one-to-one CAM profile is normally active and synchronized to the master input, unless the Lock On/Lock Off feature is not active. Under normal operating conditions, this CAM profile is active and follows the group's active master input.

State of Lock On/ Lock Off bit

G#_CT_LOCK_OFF = ***0***

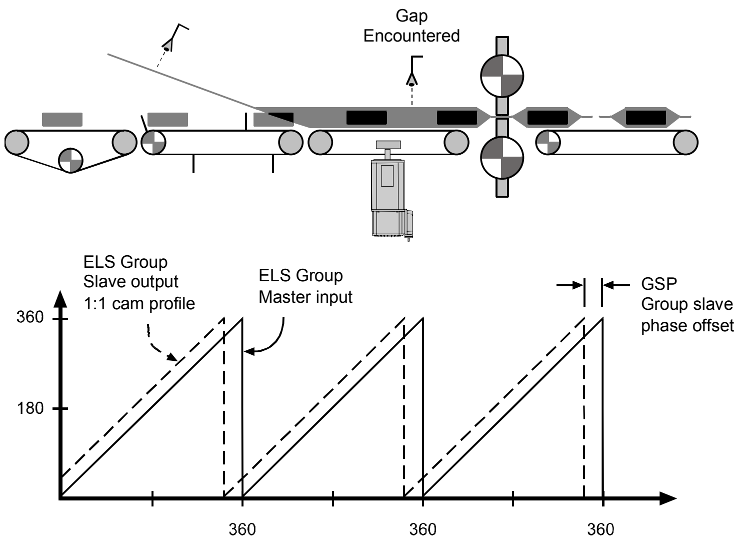


Fig. 6-31: Normal Operation of Wrapper Application

Rexroth
Bosch Group

Lock Off The Lock Off CAM profile decelerates to a stop over one cycle of the master. After this cycle, the group's velocity is stopped and will not restart unless the LOCK OFF bit is toggled.

Note: All motion to the ELS Group Master, as well as any cascading groups, will stop.

State of Lock On/Lock Off bit

G#_CT_LOCK_OFF = **0 to 1**

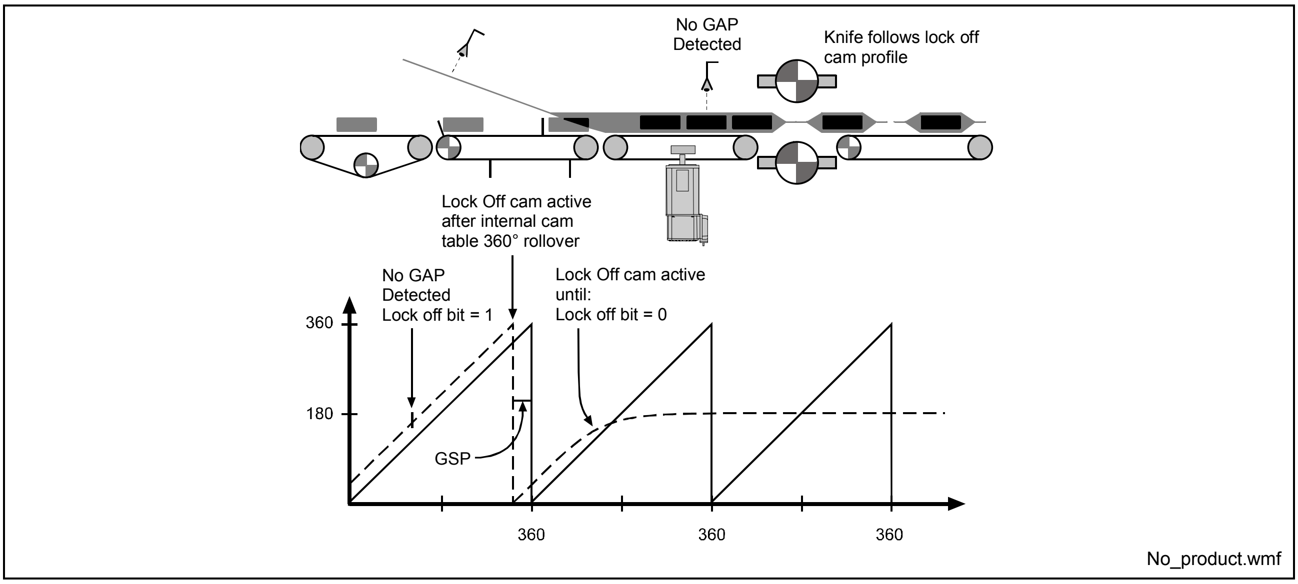


Fig. 6-32: Lock Off CAM Active, *No Product - No Seal*

Lock On The Lock On CAM profile is active and accelerates from a stopped position to match the velocity of the master input over one cycle of the master (360 degrees). After this cycle, the velocity of the group matches that of the master.

State of Lock On/Lock Off bit

G#_CT_LOCK_OFF = ***1 to 0***

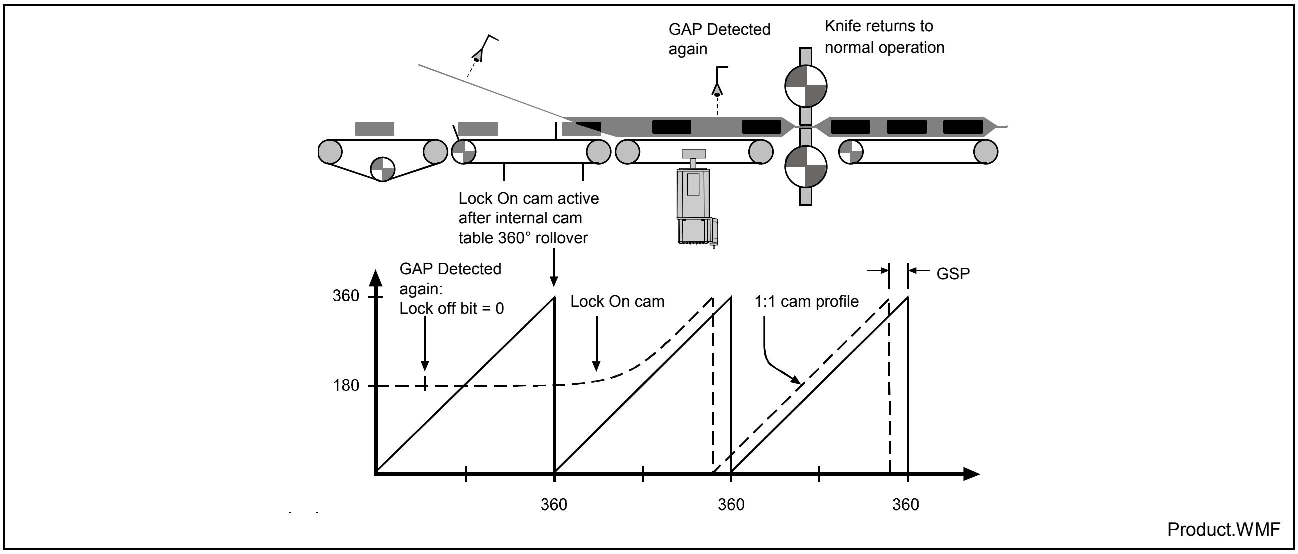


Fig. 6-33: Lock On CAM Active, Product is Present Once Again

Phase Control

The group master and slave phase adjust defaults to a trapezoidal velocity profile using:

- dynamic synchronization acceleration
- dynamic synchronization velocity

Relative phase adjusts are written to the following ELS Group float variables

- G#_REL_M_PH (group master relative master phase adjust)
- G#_REL_S_PH (group master relative slave phase adjust)

These variables are triggered with bit 2 or 3 in the ELS group control register. After execution of the velocity profile, the absolute group master or group slave phase adjust is updated. The phase adjust can also be configured to execute in one step by selecting **Immediate** in the *Phase* window.

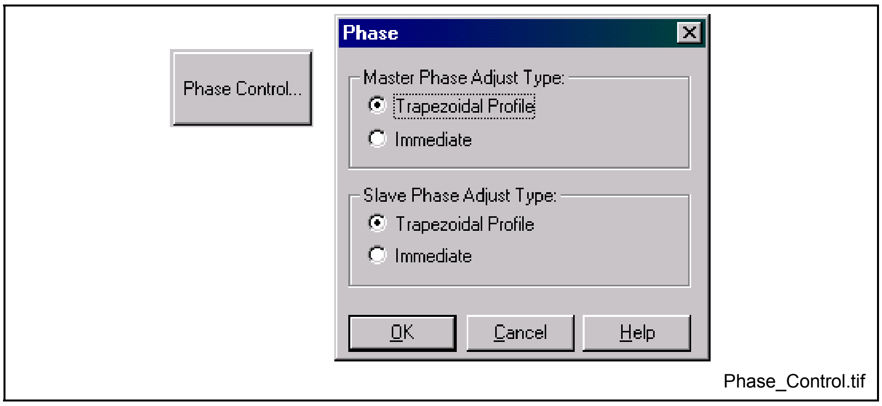


Fig. 6-34: Phase Control

The absolute group master and group slave phase adjust is triggered by a 0 → 1 bit transition. The bit goes to 1 at the start of the phase adjust and to 0 when the phase adjust is complete. The direction of the move is dependent on the settings of the ELS Group Configuration:

Bit 11: Group Master Absolute Phase Adjust

Bit 7	Bit 6	Description
0	0	Shortest path
0	1	Positive phase lock, if phase difference is greater than G#_LOCK_WIN, or use shortest path
1	0	Negative phase lock, if phase difference is greater than G#_LOCK_WIN, or use shortest path
1	1	Phase lock disabled (provides velocity synchronization only)

Bit 12: Group Slave Absolute Phase Adjust

Bit 7	Bit 6	Description
0	0	Shortest path
0	1	Positive phase lock, if phase difference is greater than G#_LOCK_WIN, or use shortest path
1	0	Negative phase lock, if phase difference is greater than G#_LOCK_WIN, or use shortest path

The absolute phase adjust values can only be read except when the stop ramp is active and the group master is at standstill. In this case the absolute phase adjust values can be overwritten and forced.

If an ELS Group is switched to local or manual mode during a phase adjust with a trapezoidal velocity profile, the phase adjust will be completed.

If an ELS Group is switched to parameter mode during a phase adjust, the phase adjust will be aborted and the group will revert back to its prior offset.

Initialization Control

The Initialization window is used to configure the ELS group master operation during project initialization. The configuration set in this window is set in parameter mode.

Sync to ELS Group master on task switching from Manual to Automatic Mode

This can be set at automatic or you can switch the group switch the ELS group to the master manually. The group will remain in local mode when the project comes out of parameter mode until it is manually switched.

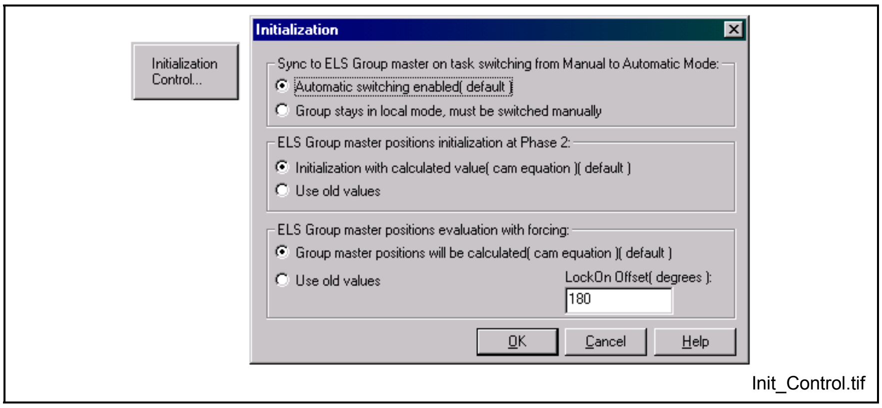


Fig. 6-35: Initialization Control

ELS Group master position initialization at Phase 2

During phase 2 of power up, you can select to use either old values or one of the following:

- the variables initialized by the internal group master being set to the value of the active group input master
- the CAM table input position set to the internal group input master position plus the group master offset
- the state of the state machine set to 1
- the old values kept for absolute master and group slave offset
- the ELS group master position calculated with the CAM equation

ELS Group master position evaluation with forcing

In phase 4 of power up, the ELS group can be reinitialized when the local mode is active and the ELS group master is at standstill (G#_ST_MOTION is 0). Under these conditions, the following variables are not updated by the control and can be overwritten:

- Internal group input master position
- Group CAM table input position (used to calculate the ELS group master position)
- ELS group master position (only when bit 9 in the ELS configuration word is set to 1)
- State of the state machine for lock on/lock off (used to calculate the ELS group master position)
- Absolute group master and group slave offset (used to calculate the ELS group master position)

You can select to use the old values and no calculation of the CAM equation will be performed or you can select the Group master position to be calculated. The Group master is calculated using the group CAM table input position, the absolute group master and group slave offset, and the selected state of the state machine.

Lock On Offset The default setting of 180 degrees matches the default profile in VisualMotion. This number can be changed, but the lock on/lock off profile must be completed in 1 revolution of the master cycle.

Standard ELS Group Phase Adjust

The standard ELS Group master (GMP) and slave (GSP) phase adjust functionality operates such that a move profile is superimposed on the group's current motion, based on the acceleration and velocity specified in program variables G#_SYNC_ACCEL and G#_SYNC_VEL.

The time it takes for the phase adjustment to execute is a function of the synchronization acceleration and velocity and is independent of the group's input master position. Therefore, the phase adjustment executes to completion regardless of whether the group's input master is moving or not. For example, a negative phase adjustment can cause the output of an ELS Group to move backwards even though the group's input master is at standstill. Depending upon the requirements of the application, this behavior can be undesirable.

Program Variables for Standard Phase Adjust

The following ELS Group program variables are used to set and store position values when using a standard ELS Group phase adjust:

ELS Group Program Variable	Description	Update Mode
G#_PROG_M_PH	Value added to current ELS Group master input when a phase adjust is performed.	Phase 4
G#_ABS_M_PH	Total or absolute value of master phase adjust performed on ELS Group input master.	Phase 4 (read-only) and Forcing
G#_PROG_S_PH	Value added to current ELS Group output position when a phase adjust is performed.	Phase 4
G#_ABS_S_PH	Total or absolute value of master phase adjust performed on ELS Group output position.	Phase 4 (read-only) and Forcing

Table 6-27: ELS Group Phase Adjust Program Variables

Register Bits for Standard Phase Adjust

The following ELS Group control and status registers bits set and monitor ELS Group phase adjust:

ELS Group Control Register	Description
Bit 2	Group Master Relative Phase Adjust
Bit 3	Group Slave Relative Phase Adjust
Bit 11	Group Master Absolute Phase Adjust
Bit 12	Group Slave Absolute Phase Adjust

Table 6-28: ELS Group Phase Adjust Control Register Bits

ELS Group Status Register	Description
Bit 2	Group Master Phase Adjust Status
Bit 3	Group Slave Phase Adjust Status
Bit 11	Group Master Absolute Phase Adjust Status
Bit 12	Group Slave Absolute Phase Adjust Status

Table 6-29: ELS Group Phase Adjust Status Register Bits

Note: Refer to control registers 152-159 and status registers 243-250 in chapter 16, volume 2 of the VisualMotion 11 Functional Description for details.

Master-Following ELS Group Phase Adjust

The ELS Group master-following phase adjust option is new to VisualMotion 11 and expands on the standard behavior such that the execution of a phase adjust is tied to the Group master’s motion through a triangular velocity CAM profile.

The completion of the phase adjust is dependent upon the forward movement of the Group’s input master over a pre-defined distance (G#_MST_DIST_M_PH or G#_MST_DIST_S_PH). The behavior is such that phase adjustments only occur when the Group’s master moves. Therefore, it is not possible for the Group’s output to move (forwards or backwards) during the execution of a phase adjustment when the Group’s master is at standstill.

The profile of a Master-Following Phase Adjust is not affected or controlled by G#_SYNC_ACCEL or G#_SYNC_VEL. Rather, the phase adjustment profile is a function of the specified master distance and the motion of the Group’s input master.

Note: It is important to note that the velocity of large Master-Following Phase Adjusts may exceed the maximum velocity of axes attached to the Group because of the use of a triangular velocity CAM profile.

Program Variables for Master-Following Phase Adjust

The following ELS Group program variables are used to set and store position values when using the ELS Group master-following phase adjust:

ELS Group Program Variable	Description
G#_MST_DIST_M_PH	This variable specifies the forward Group Master distance over which the master phase adjust will be performed. For instance, if G#_MST_DIST_M_PH = 100º and the phase adjust was triggered when the master was at 0°, the phase adjustment would occur such that it is complete when the Group Master passes 100º. This distance value must be positive – if a negative value is specified, its absolute value will be used. This value is editable in Run Mode (phase 4). However, changing the value after a master phase adjust has begun will have no effect until another master phase adjust is begun (the value is *captured* when the phase adjust is triggered). If the distance is set to zero, an immediate phase adjust will be performed. The phase adjust will occur within one Sercos cycle, unless the phase adjustment is very small. The drives should be disconnected from the Group during the phase adjust in order to prevent faults.
G#_MST_DIST_S_PH	This variable specifies the forward Group Master distance over which the slave phase adjust will be performed. For instance, if G#_MST_DIST_S_PH = 100º and the phase adjust was triggered when the master was at 0°, the phase adjustment would occur such that it is complete when the Group Master passes 100º. This distance value must be positive – if a negative value is specified, its absolute value will be used. This value is editable in Run Mode (phase 4). However, changing the value after a slave phase adjust has begun will have no effect until another master phase adjust is begun (the value is *captured* when the phase adjust is triggered). If the distance is set to zero, an immediate phase adjust will be performed. The phase adjust will occur within one Sercos cycle, unless the phase adjustment is very small. The drives should be disconnected from the Group during the phase adjust in order to prevent faults.

Table 6-30: ELS Group Master-Following Phase Adjust Program Variables

Register Bits for Master-Following Phase Adjust

The following ELS Group control registers bits are used to enable and abort an ELS Group master-following phase adjust.

ELS Group Control Register	Description
Bit 15	G#_CT_MSTR_FOL_PH
Bit 16	G#_CT_MSTR_ABORT_PH

Table 6-31: ELS Group Phase Adjust Control Register Bits

Note: The control and status of an ELS Group master-following phase adjust is performed using the standard ELS Group phase adjust control and status register bits. Refer to Standard ELS Group Phase Adjust for details.

Rexroth
Bosch Group

General Functionality of Master-Following Phase Adjust

A master-following phase adjust occurs from the position that the Group's master was at when the phase adjust was triggered through the distance specified in G#_MST_DIST_M_PH or G#_MST_DIST_S_PH. This range of master positions defines a sort of *phase adjustment window*. The starting position of the window is the position the Group's master is at when the phase adjust is triggered. The ending position of the window is the start position plus the distance defined in G#_MST_DIST_M_PH or G#_MST_DIST_S_PH.

For example, if a slave phase adjust is triggered when the Group's master was at 100º, and G#_MST_DIST_S_PH is set to 100º, the start position of the window is 100º and the end position of the window is 200º.

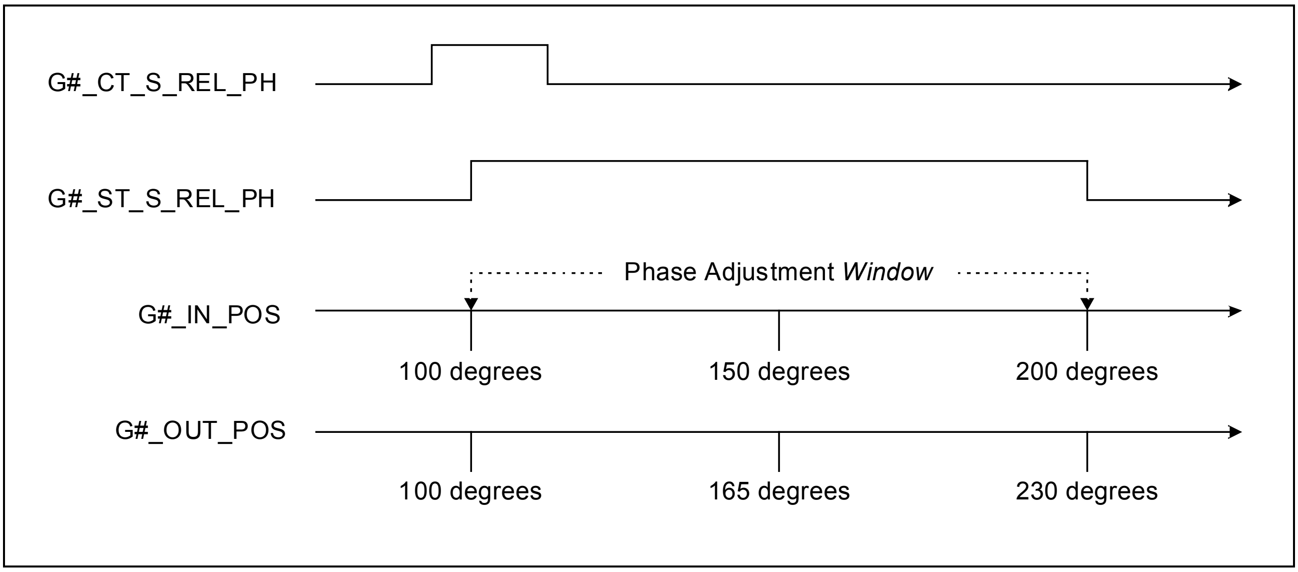


Fig. 6-36: ELS Group Master Following Phase Adjust Example

In the example of Fig. 6-36 (above), G#_ABS_S_PH is assumed to be 30º, G#_MST_DIST_S_PH is assumed to be 100º, and G#_CT_MSTR_FOL_PH is assumed to be set (1).

The master-following slave phase adjust is triggered via the G#_CT_S_REL_PH control bit when the Group's master is at 100º. The G#_ST_S_REL_PH status bit then goes high to confirm that the phase adjust is in progress and stays high until the phase adjust is completed when the Group's master passes 200º.

The phase adjustment will not complete until the Group's master has moved outside of the phase adjustment window in the forward direction. The ELS Group status register phase adjustment bits can be used (as shown in the figure above) to confirm that the phase adjust has completed.

Note: It is important to note that the execution of a master-following slave phase adjust is dependent upon the motion of the Group's master. Therefore, it is important to consider the Group's M/N gear ratio when specifying the G#_MST_DIST_S_PH value.

Group Master Reverse Motion

Master-following phase adjusts can only complete execution when the Group's master moves in the forward direction. Reverse motion of the Group master during a master-following phase adjust is handled in the following ways:

- **Reverse Motion within the Phase Adjust Window**

  If the reverse motion occurs within the phase adjustment window, the phase adjustment process will simply be *reversed*. For example, in the case of diagram 4.0, the Group achieves a phase adjustment of 15º by the time the master has reached 150º. If the master then moves back to 100º, the Group will be back to a phase adjustment of 0º.

- **Master Moves Backwards Past the Phase Adjust Window**

  If the master reverses past the phase adjustment window start position (the master position at which the phase adjust was initially triggered), the phase adjustment will remain triggered (with the associated status bit remaining high), but the phase adjustment will remain at 0º (none). The phase adjustment will begin execution, again, when the Group master moves past the phase adjustment window start position in the forward direction.

Note: If the master reverses 360º (one revolution) or more past the phase adjustment window start position, the phase adjust will be aborted. In such a case, the associated ELS Group status register bit will go low (0) to confirm that the phase adjust has been aborted.

Phase Adjusting Over Small Master Distances

Care should be taken when specifying small group master distances (G#_MST_DIST_M_PH or G#_MST_DIST_S_PH) for the phase adjust.

In general, master-following phase adjusts that are specified to occur over small master distances can result in large position jumps in the Group's output that may result in drive faults. The following aspect should also be considered:

- **Zero Master Distance is Specified (Immediate Phase Adjust)**

  When the master distance (G#_MST_DIST_M_PH or G#_MST_DIST_S_PH) is set to zero, an immediate phase adjust will be performed – the phase adjust will occur within one SERCOS cycle. Unless the phase adjustment is very small, the drives should be disconnected from the Group during the phase adjust in order to prevent faults.

Aborting a Master-Following Phase Adjust

A master-following phase adjust can be aborted by raising the G#_CT_MSTR_ABORT_PH bit. The Group will retain any partially executed portion of the phase adjust as an offset when a master-following phase adjust is aborted.

For example, if the phase adjust in diagram 4.0 were aborted when the master was at 150º, the Group would retain the 15º phase adjustment that had already been achieved. The associated ELS Group status register phase adjustment bit will go low to confirm that a master-following phase adjust has been aborted.

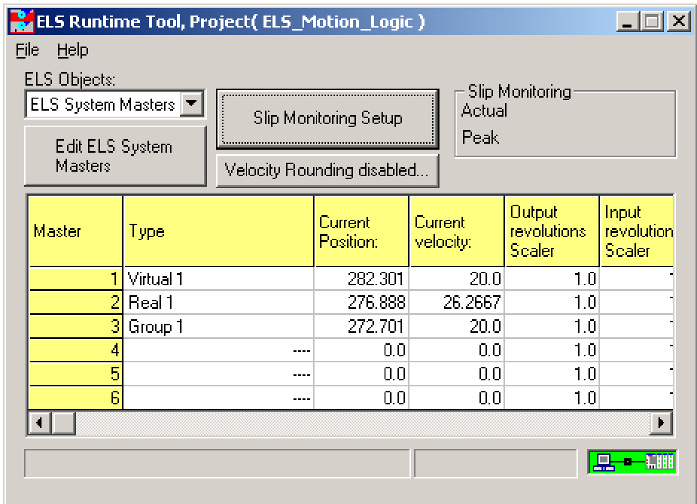
Editing ELS Groups and System Masters

The ELS Runtime tool in VisualMotion can be used to initialize or change the settings of the ELS System Masters, ELS Groups and Virtual Masters in offline or online mode.

In offline mode, ELS system components can be initialized with default or user-defined values. Data is saved with the project and can be synchronized with the control when switching to online mode.

In online mode, modifications to ELS components are saved to both the project and control. These changes will remain with the project on the control's memory unless the project is recompiled and downloaded to the control. If ELS icons contain values that are different from those modified using the ELS Runtime Tool, then the values in the icons will be used when the icon is executed in the logical program flow.

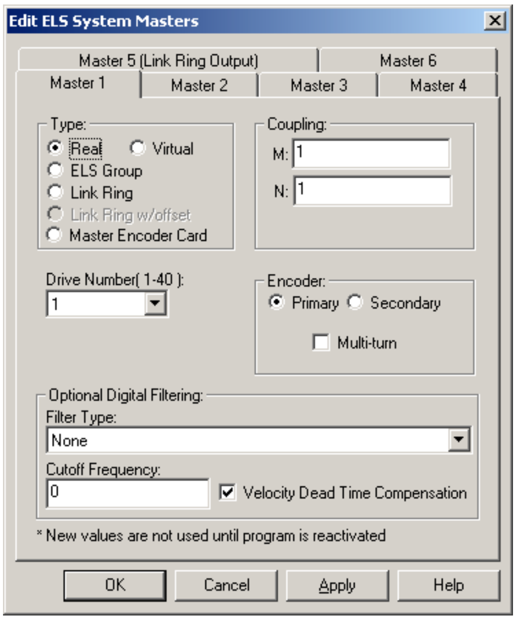
To select a group or master to edit, use the drop-down menu in the ELS Runtime Tool window. When one of the menu items has been selected, a button for that item will appear in the window. Clicking the button will open a window for configuring that group or master.



ELS_Runtime_Tool.tif

Fig. 6-37: ELS Runtime Tool

The association of the ELS System master to its number in the connection interface can be changed by selecting this option in the ELS Runtime Tool. The ELS System Masters can be edited by selecting the Master number tab, which corresponds to the Connection Interface designation for that master, see Fig. 6-38.



ELS_Runtime_Utility.tif

Fig. 6-38: Edit ELS System Masters Window

Changes made in the runtime utility take effect immediately in the control when you click the OK button.

Configuring a Link Ring for ELS System

Link Ring is a fiber optic loop that can be added to an ELS system to provide a means of transferring ELS position data between controls. A Link Ring has the capacity to connect 32 controls together with each control in the ring capable of supporting 64 drives.

Scalable Link Ring Functionality

The Link Ring functionality is scalable based on the maximum number of controls configured in the Link Ring system. The Link Ring addressing for both master and slave must be assigned within the range specified in the table below.

Max. PPC Controls	Addressing Range	Min. Cycle Time (ms)
8	1 to 8	2
16	1 to 16	4
32	1 to 32	8 or 16

Table 6-32: Scalable Link Ring Cycle Times

Based on the maximum number of PPC controls actively participating within the Link Ring, the user is responsible for configuring the following parameters for the Link Ring Master and each corresponding Link Ring Slave(s):

- C-0-0304 Link Ring Address
- C-0-0305 Link Ring Cycle Time

Note: In order to establish Link Ring communication, each Link Ring address must be unique and within the allowable range, the cycle times for all nodes must match, and any changes to Link Ring parameters require a power cycle of the control.

Rexroth Bosch Group

Link Ring Types

A Link Ring can be set up in either a single or double Link Ring configuration. A single Link Ring connects all Link Ring Slaves in series to the Link Ring Master through the primary receive and transmit terminals of each DAQ03 card. A double ring is two separate fiber optic rings connected to the primary and secondary receive and transmit terminals of each DAQ03 card in the Link Ring.

Link Ring Setup

All controls in the Link Ring can run the same or different ELS projects. The Link Ring settings in the ELS project in each control are based on that control's function in the Link Ring. Each control will have a unique Link Ring setting.

To configure a control participating in a Link Ring:

1. Select ***Tools*** ⇒ ***Control Settings*** in VisualMotion Toolkit and set the **Link Ring Address** (C-0-0003) of each node in the *Link Ring* tab. Valid addresses for a Link Ring are 1 to 32.
2. In the *Link Ring* Tab of the *Control Settings* window, select the Link Ring type (single or double) and the control type (master, slave, or repeater).

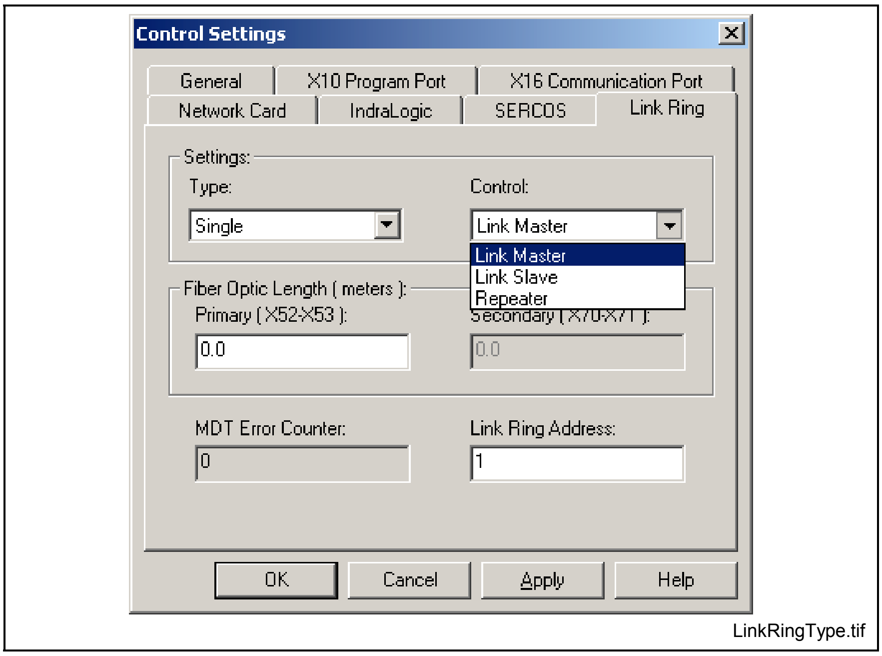


Fig. 6-39: Link Ring Control Type

Fiber Optic Length

The length of the fiber optic cables will determine the required data transmission power for the Link Ring. The length entered in this field corresponds to the length of the fiber optic cable connected to the transmit (TX) terminals of the DAQ03 card in the control. The length entered for a primary ring is stored in card parameter C-0-0301, *Link Ring Primary Fiber Optic Length.* The length entered for the secondary ring is stored in parameter C-0-0302, *Link Ring Secondary Fiber Optic Length.* Parameter C-0-0302 is not used when the system is configured as a single ring. Refer to the *VisualMotion 11 Project Planning* manual for information on error reactions associated with fiber optic length settings.

MDT Error Counter

The master data telegram (MDT) carries data from the Link Ring Master to the Link Ring Slaves. The **MDT Error Counter** field displays the number of MDT errors the slave received from the master. This value can also be accessed in control parameter, C-0-0303.

Project Settings for Link Ring

A Link Ring is configured in a VisualMotion project with the ELSMstr1 and ELSGrp3 icons.

ELS System Master Icon

Within the *Setup ELS System Master Assignment* window, the Link Ring Master source is selected. The master source is either a Virtual Master, Real Master, or ELS Group whose position data is to be made available to other Link Ring nodes. By sending the master source signal through ELS System Master 5, it becomes available to other Link Ring nodes, see Fig. 6-40.

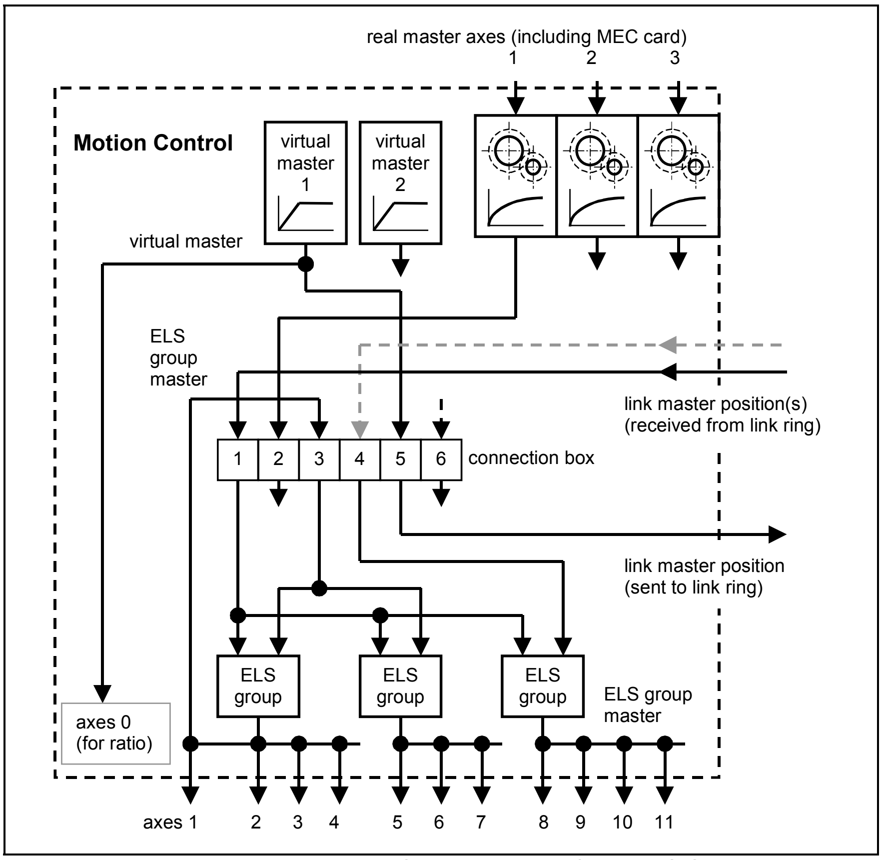


Fig. 6-40: Link Ring Data Routing to Connection Interface (ELS System Master)

To establish the Link Ring Master:

1. Select Master 5 in the *ELS System Master Assignment* window and click the ***Edit…*** button to open the *Setup ELS System Master 5 (Link Ring Output)* window.
2. Select the master type and the settings for it:
   - Real Master – Select Drive Number and filter type
   - Virtual Master – Select the Virtual Number (1 or 2)
   - ELS Group – Select the Group (1 – 8)
3. Select Link Ring as the Type and an External Number from 1 to 32 to indicate which Link Ring Master position (identified by the node's number designation in Link Ring) the ELS System Master will be receiving.

The Link Ring Master output from ELS System Master 5 can be used locally or exported to another system. As shown in Fig. 6-41, ELS System Master 1 of Link Ring Master (node #1) receives it's local Link Ring Master output through the Link Ring. ELS System Master 1 of Link Ring Slave (node #2) receives its signal from Link Ring Master (node #1) and

ELS System Master 2 receives its signal from its local Link Ring Master through the Link Ring.

While only one control can be designated as a Link Ring Master, each control (master and slave) can export a Link Ring Master position. This master position can be used locally, exported to another system (each system can import five master positions), or both so that there can exist up to 32 master signals being transmitted through the Link Ring.

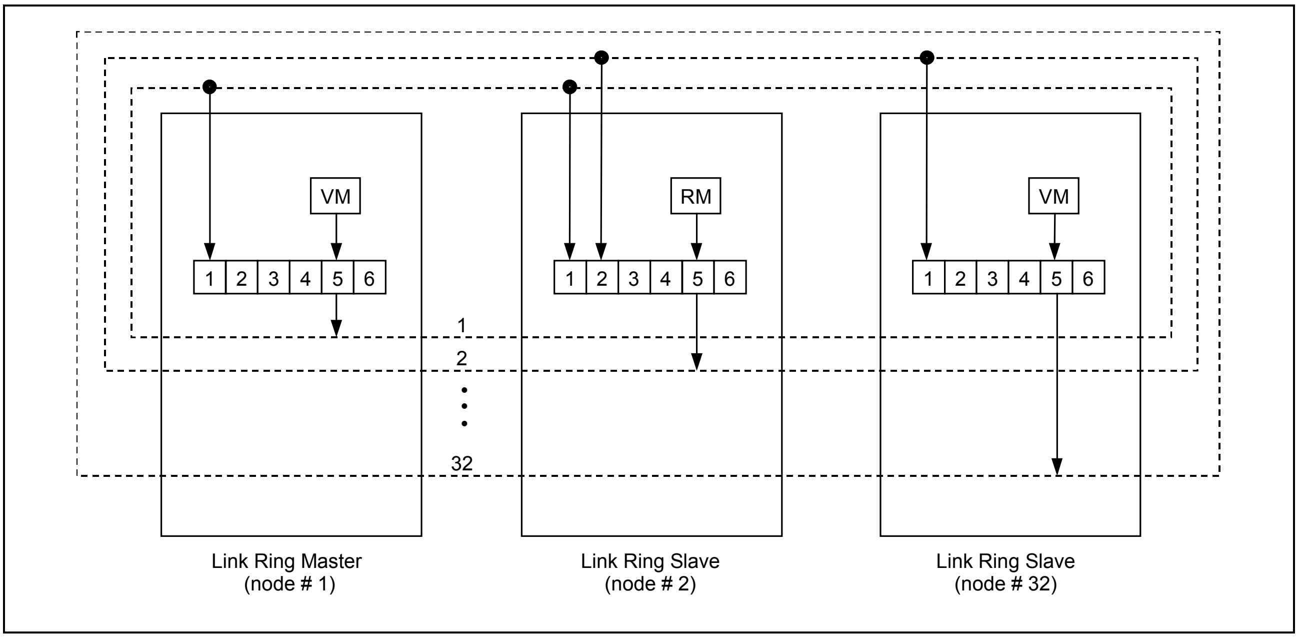
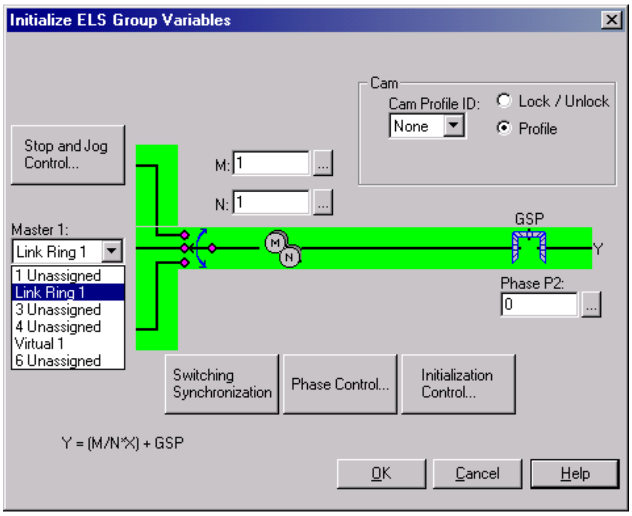


Fig. 6-41: Link Ring Example

ELS Group Icon

Any ELS Group participating in a Link Ring can import a Link Ring Master position from any ELS System. The ELS System Master position that the ELS Group imports is set in the *Initialize ELS Group Variables* window, see Fig. 6-42, which is opened by clicking the ***ELS Group Variables compile time initialization…*** button in the *ELS Group Setup* window. The Link Ring Master set in step 3 of the Link Ring Master setup (Link Ring 1) is listed in the drop-down menus of the Master 1 and Master 2 edit boxes.

Any ELS Group participating in a Link Ring configuration should receive master position data from the Link Ring and not locally from the node's master source. Deriving a position from the master source introduces a phase difference between a local ELS Group and an ELS Group receiving position data through the Link Ring.



ELS_Group_Import_LRMaster.tif

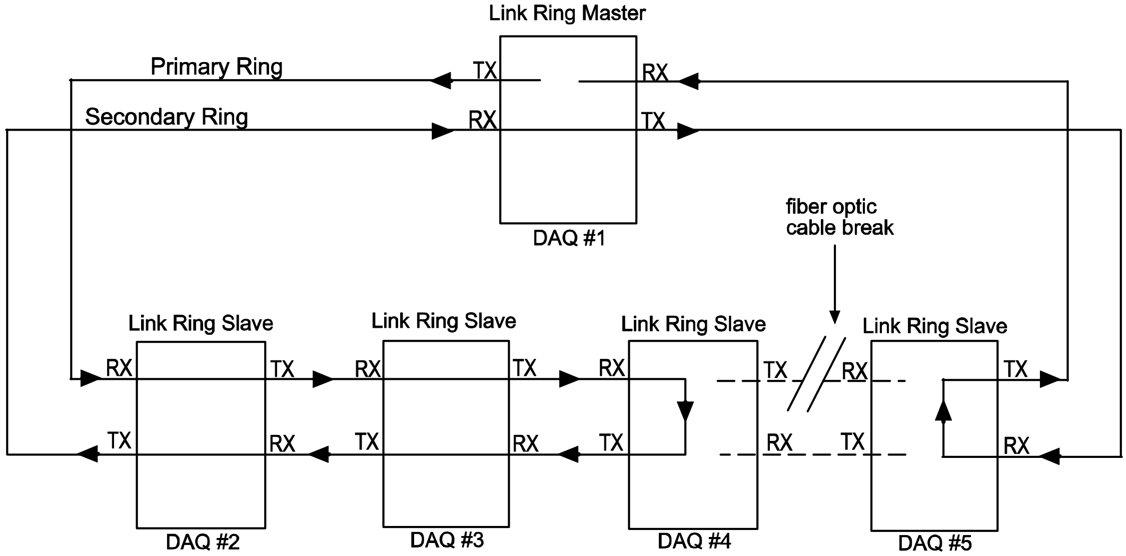
Fig. 6-42: ELS System Master selection for ELS Group

Fault Tolerance in Double Ring

In a double Link Ring, a fault (such as a break in the cable) in the primary or secondary ring, or in both the primary and secondary rings between two adjacent controls, will not disrupt signal transmission. Under normal operating conditions, communication takes place on the primary ring while the secondary ring transmits only a diagnostic signal for error detection. When a fault occurs, the secondary ring begins transmitting the signal as needed then reroutes the signal back to the primary ring to complete the loop.

Fault in Primary Ring

When an fault occurs in the primary ring, the Link Ring data position signal is redirected to the secondary ring, see Fig. 6-43.



Single_Fault_Primary.tif

Fig. 6-43: Single Fault in Primary Ring

In this example, a fault occurs in the primary ring between DAQ #4 and DAQ #5. DAQ #4 reacts by turning on the H18 LED, which indicates an error in the primary ring between itself and DAQ#5. The Redundancy_Loss, Bit 7, of register 40 (Link_Ring_Status) and Error_Prim_Opt_Ring and/or Error_Sec_Opt_Ring, bits 5 and 6 of Register 40, will go high in at least one of the nodes of the Link Ring. The

Rexroth Bosch Group

signal switches between the primary and secondary rings at DAQ #4 and DAQ #5 on either side of the break.

Fault in Secondary Ring

A fault in the secondary ring will not affect the signal transmission in the primary ring. The primary ring will continue to function, but redundancy will be lost when the secondary ring is no longer functional. Bit 7 of Register 40 in at least one of the nodes will go high, indicating that redundancy has been lost, and Bit 6 of Register 40 will go high, indicating that an error has occurred in the secondary ring.

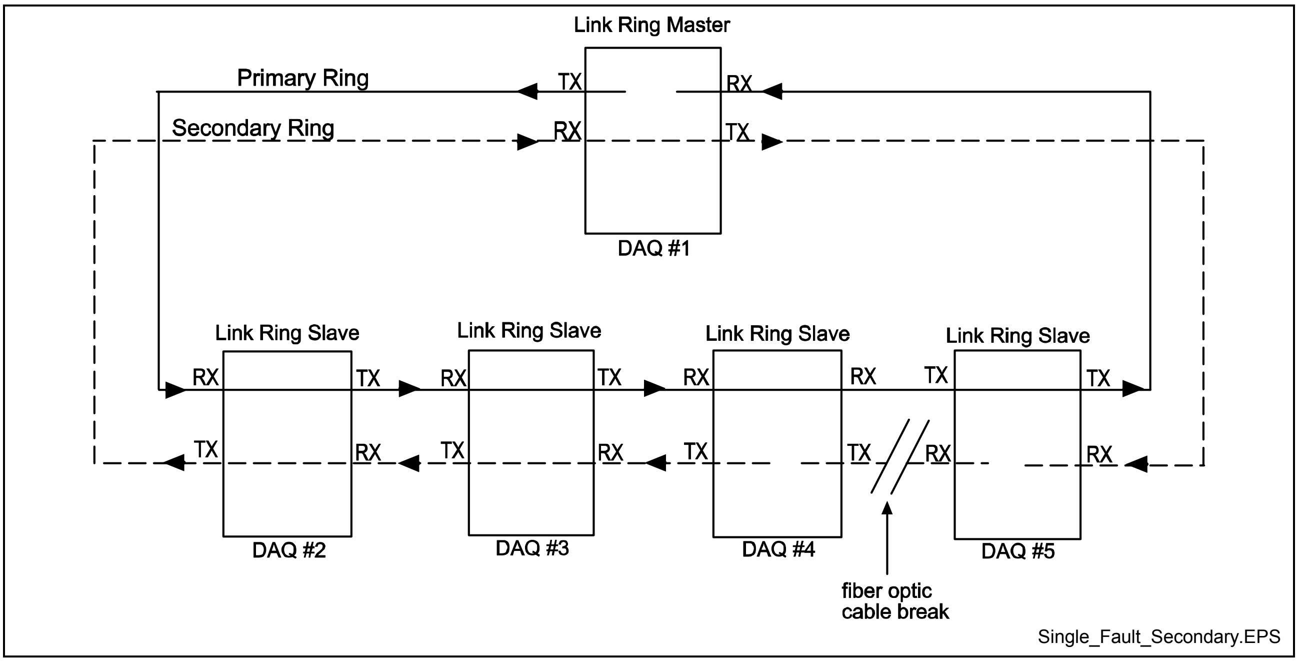


Fig. 6-44: Fault in Secondary Ring

Fault in Primary and Secondary Rings

When a fault occurs in both the primary and secondary rings between the same two nodes, the signal is routed through the secondary ring to complete the loop. Bit 5 of Register 40 will go high to indicate a redundancy loss and either bit 5 or bit 6 of Register 40 will go high to indicate an error in the primary or secondary rings.

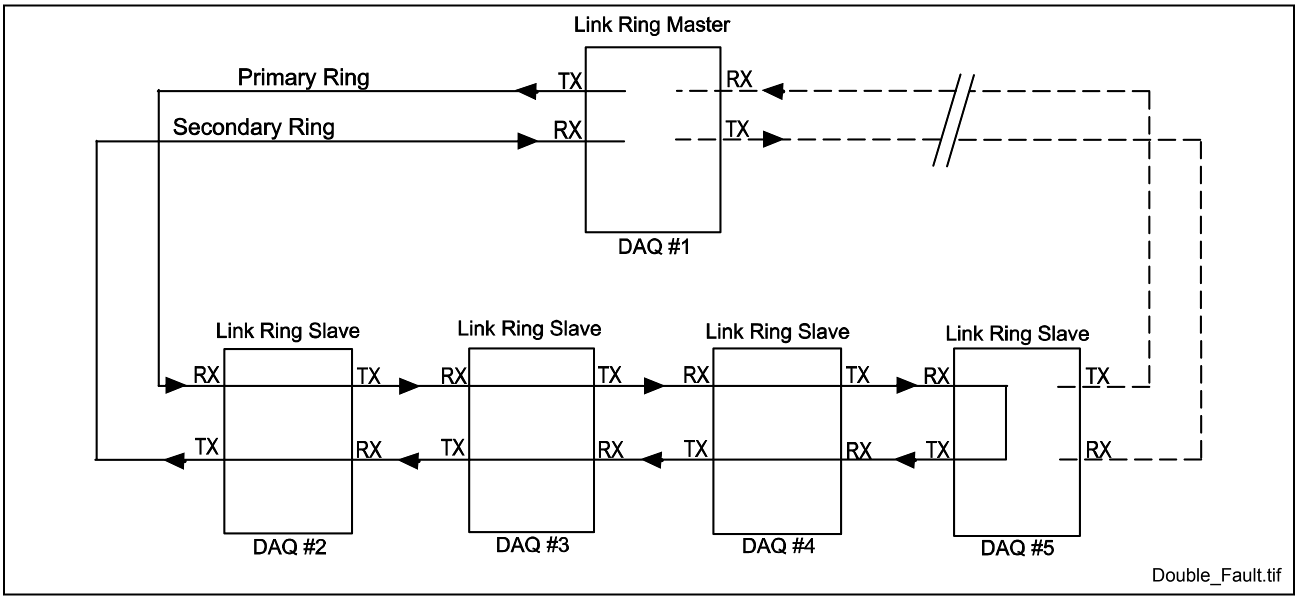


Fig. 6-45: Faults in Primary and Secondary Rings

If a fault occurs in the primary and secondary rings between different controls, the system will shut down. If more than one fault occurs in the Primary ring, the system will shutdown.

Clearing A Redundancy Loss

If a redundancy loss (Register 40, Bit 7) or 541 Link Ring Error (Transmission path defective) is indicated on any node of a given Link Ring, use the following procedure to clear the error.

1. Locate and correct any fiber optic cable problems.
2. Hold the Rebuild Double Ring input (Register 1, Bit 7) high on all nodes until every node's Redundancy Loss bit (Register 40, Bit 7) goes low. You can set a timeout in your logic in the event that the bit does not go low after 10 seconds. This step must be done with all nodes in phase 4.
3. Set the Clear_All_Errors input (Register 1, Bit 5) high for all nodes that have their Error output (Register 21, Bit 5) set high. The Link Error output (Register 40, Bit 4) can also be used as a flag. This bit will only go high for Link Ring-related errors. You can set the Clear_All_Errors input low when the Error output goes low. This step does not need to be performed for all nodes simultaneously, like the Rebuild Double Ring in step 2.

Note: The Clear_All_Errors input does not force a node to execute a Rebuild Double Ring. If performed on its own, without a Rebuild Double Ring, a Redundancy Loss will remain.

6.2 Control CAM System Usage

This section describes Control CAM functionality as used by CAM axes or ELS functionalities such as ELS Groups and Coordinated Articulation.

Switching to a Control CAM

The use of a control CAM by a CAM Axis, ELS Group, or Coordinated Articulation axis before the CAM is ready (i.e., before the CAM was downloaded or built) is not supported.

One of the following errors will be generated if an attempt is made to switch to a control CAM that is not ready:

- “487 CAM # is invalid or not stored”
- "573 CAM # is being built"

If the CAM axis or ELS function (ELS Group, or Coordinated Articulation axis) is not fully decelerated by the time it reaches the CAM’s next zero crossing, the CAM will be switched to the default 1:1 (null) CAM at the next zero crossing.

Downloading and Rebuilding a Control CAM

Control CAMs that are actively being used by a CAM axis, ELS Group, or Coordinated Articulation axis can not be rebuilt or downloaded to the control. If an attempt is made to rebuild an active (in use) control CAM using the CAM Build icon, the operation will be prevented and the following error and task diagnostic will be generated:

- “488 CAM Error: see Task % diag.”
- “488 Cannot store CAM: already active for axis #”

If an attempt is made to download or delete an active (in use) control CAM, the operation will be prevented and the following Direct ASCII Communication error will be returned:

- “!58 Cannot store CAM: already active for axis #”

Note: An *active* CAM refers to a CAM that is being actively used by a CAM Axis, ELS Group, or Coordinated Articulation axis. The axis number (#) referenced in the above errors will show the CAM Axis number if the CAM is being used by a CAM Axis, otherwise the axis number will be zero if the CAM is being used by an ELS Group or Coordinated Articulation axis.

Downloading or Rebuilding a Control CAM used by a CAM Axis

This is an exception to the descriptions made in the section titled "Downloading and Rebuilding a Control CAM" in which it is possible to download or rebuild a control CAM for a CAM Axis. This can be done by *desynchronizing* the CAM axis (placing it in single axis mode). Once the CAM axis is in single axis mode, its current control CAM can be downloaded or rebuilt without generating an error.

The following points should be noted with respect to the above process:

1. The transition between the old and new instances of the CAM can occur at any position – the transition will occur at whatever position the CAM is at when the download or build operation is completed.

2. The default 1:1 (null) CAM will be used during any period of time in which the old instance of the CAM is deleted and the new instance is not ready.
3. Position jumps may occur as a result of points 1 and 2.

Downloading or Rebuilding a Control CAM used by an ELS Group

This is an exception to the descriptions made in the section titled "Downloading and Rebuilding a Control CAM" in which it is possible to download or rebuild a control CAM use by an ELS Group. This can be done by placing the ELS Group into Local Mode. Once the ELS Group is in Local Mode, its current control CAM can be downloaded or rebuilt without generating an error. The Group's slave drives should be *desynchronized* (placing them in single axis mode) during the control CAM download or build operation some position jumps may occur.

The following points should be noted with respect to the above process:

1. The transition between the old and new instances of the CAM can occur at any position – the transition will occur at whatever position the CAM is at when the download or build operation is completed.
2. The default 1:1 (null) CAM will be used during any period of time in which the old instance of the CAM is deleted and the new instance is not yet ready.
3. Position jumps may occur as a result of points 1 and 2, above. In as much, the Group's slave drives should be desynchronized to prevent faulting.

ELS Group G#_ACTIVE_CAM Program Variable

The ELS Group G#_ACTIVE_CAM program variable accurately reflects the control CAM currently being used by the Group. A new CAM is written to program variable G#_ACTIVE_CAM at the next zero crossing for the following CAM types:

CAM Type	ELS Group Integer Variable
Profile CAM	G#_USER_CAM
Lock on / Lock off CAM	G#_LOCKON_CAM
	G#_RUN_CAM_ID
	G#_LOCKOFF_CAM

Table 6-33: ELS Group CAM Type

Switching to a Control CAM in Sercos Phase 4 (Run Mode)

A CAM used for a CAM Axis, ELS Group, or Coordinated Articulation axis can be switched while in Sercos phase 4. A CAM axis, ELS Group, or Coordinated Articulation axis will not transition to the new CAM until the next zero crossing. This creates a smooth transition from the old CAM to the new CAM. In the case of ELS Groups, the G#_ACTIVE_CAM program variable can be used to confirm that the Group has transitioned to the new CAM.

However, it is possible to immediately (without waiting for the next zero crossing) switch a CAM axis or ELS Group to another CAM while in single axis or Local Mode, accordingly.

The procedure for each is as follows:

CAM Axis:

1. Place the drive in single axis mode.
2. Change the CAM number in axis parameter A-0-0034 (control CAM currently active).

ELS Group:

1. Place the ELS Group in Local Mode (set G#_CT_LOCAL to 1).
2. Change the CAM number in the G#_USER_CAM (if the Group is configured for Profile operation) or G#_RUN_CAM_ID (if the Group is configured for Lock on / Lock off operation).
3. Transition the Group’s forcing bit (G#_CT_VAR_CLK) to 1 and wait for the forcing acknowledgement bit (G#_ST_VAR_ACK) to go high.
4. Transition the Group’s forcing bit back to 0.
5. Optionally, confirm that the CAM switched by checking the Group’s G#_ACTIVE_CAM variable.

Note: For step 3, slave drives should not be synchronized during this operation since a position jump may occur that could be large enough to result in drive faults.

Switching to an Index CAM More Than Once

CAM axis and ELS functions such as ELS Group and Coordinated Articulation can not switch to an Index CAM that is already in use. An error will always be generated when an attempt is made to use a specific Index CAM more than once.

The following fatal error will be generated, if an attempt is made to switch to an Index CAM that is already in use:

- "533 Multiple instances of index CAM: # found"

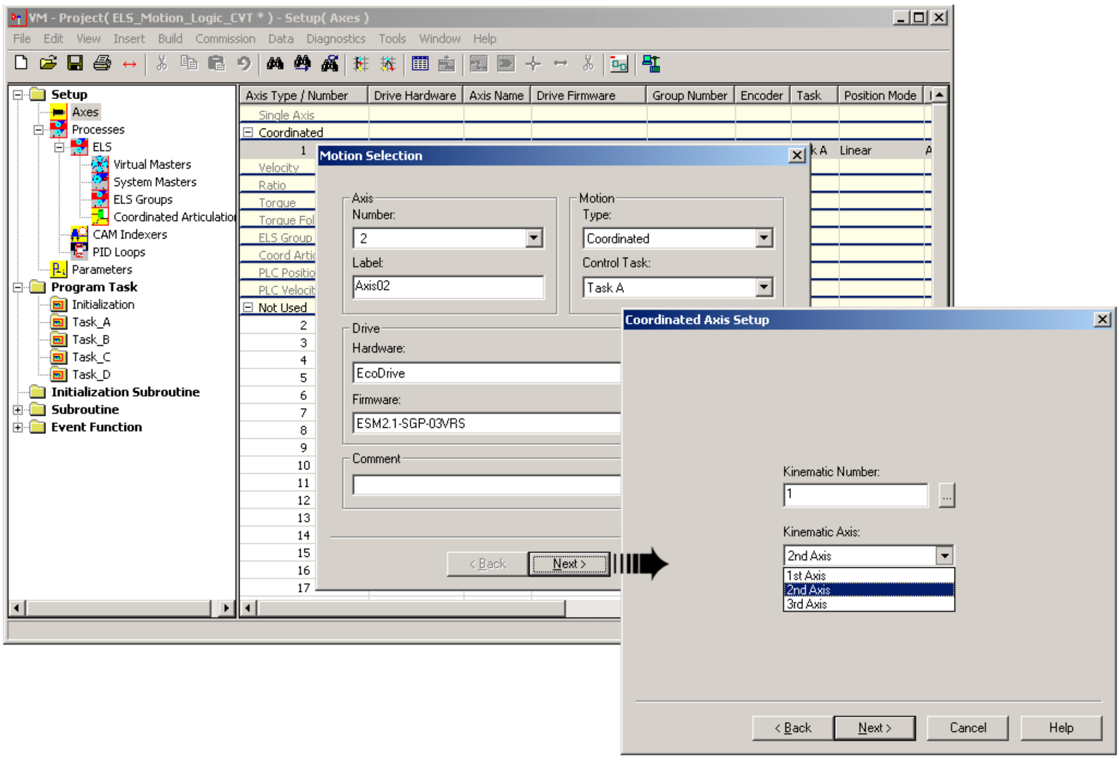
At the next zero crossing, the CAM axis or ELS function will switch to the default 1:1 CAM instead of the specified Index CAM.

6.3 Coordinated Motion

Standard Coordinated Motion

VisualMotion provides predefined kinematic libraries for controlling industrial robots in coordinated motion applications to achieve accurate high-speed positioning over geometric paths.

The assignment of a Kinematic number is performed in the *Project Navigator* window under ***Setup*** ⇒ ***Axes*** for both Coordinated and Coord Articulation axis types, as shown in Fig. 6-46.



Task_Axes_Setup.tif

Fig. 6-46: Assigning a Kinematic (Standard Coordinated Motion)

Associated Task Parameters

When a Kinematic (number) is assigned and downloaded to the control, the associated task parameters are given default values. The associated task parameters are:

- T-0-0010 ;Kinematic number
- T-0-0011 through T-0-0013 ;Coordinated axis X, Y and Z
- T-0-0050 through T-0-0059 ;Kinematic value 1 - 10

Task parameter T-0-0010 displays the assigned kinematic number in the axis icon. Task parameters T-0-0011 through T-0-0013 will display the Sercos drive address of all configured axes in a specific kinematic.

Task parameters T-0-0050 through T-0-0059 represent segment lengths in a robotic arm. Any segment task parameter can be modified to match the exact segment length for a specific robot. Task parameters are modified by selecting ***Data*** ⇒ ***Parameters*** ⇒ ***Edit*** from VisualMotion's main menu and selecting the Task tab.

Rexroth Bosch Group

Normal Case Kinematics

When using kinematics, make certain that all the mechanical settings for each axis corresponds to the physical requirements for the robot in use. Each kinematic is assigned default values for each segment in a robot's design (K values). These K values can be modified by the user to meet the requirements of their particular robot. The unit of measurement for each kinematic is defined in parameters A-0-0005 and T-0-0005. Feed constant (k) and gear ratios for each axis are unique to each machine and are defined in the *Drive n Mechanical* window. The exceptions to normal case kinematics are defined under Special Case Kinematics.

Special Case Kinematics

To maximize positioning resolution for kinematics 2, 4, 5 and 9, the user should set the mechanical setting of each axis to the following selections.

Description	Selection	Parameter
Unit of measure for position data	degree	A-0-0005
Type of scaling	Rotary	A-0-0004, bit 2
Feed constant k	6.283185 (2π)	S-0-0123

Table 6-34: Mechanical Settings

Coordinated Articulation

Coordinated Articulation is an advanced feature in VisualMotion 11 used to move up to six axes in coordinated fashion based on world coordinate inputs from an ELS Group CAM output or manual positions. This feature provides the ability to link cyclic coordinated motion to an ELS master.

Differences from Normal Coordinated Motion

The following outlines the difference in features in Coordinated Articulation from standard Coordinated Motion:

- Points table are not supported
- Motion types supported: Coordinated and ELS Group
- Zones are not supported
- Up to 6 axis supported
- Up to 4 Coordinated Articulation configurations supported (only one configuration allowed per any task A-D)
- No BTC06 support
- Control task register supported
- Coordinated limit parameters supported

Applications using Coordinated Articulation

Coordinated Articulation is used for applications requiring the fast cyclic positioning output of coordinated axes for the moving or picking up of products. Applications examples are high speed transfer tables or cyclic robotic arms.

Block Functionality

This function can be visualized consisting of four types of blocks:

Note:	Forward transformations refer to deriving world coordinates from axis positions. The forward transform is only for display purposes.

- A CAM section, taking ELS Group position and transforming it based on the CAM H factor, and Offset.
- A ramp generator, providing a trapezoidal profile for manual moves.
- A switch, selecting CAM output or manual moves for input to kinematic.
- A kinematic, converting between world coordinates and axis positions (Not all kinematics support forward transformations).

World Coordinated View

For each of the six world coordinates, x, y, z, roll, pitch and yaw, a CAM ELS Group and manual trapezoidal ramp is provided.

In CA_Sync Mode, at zero position of the ELS Group output, the CAM number, H, and offset are copied. A check is made to insure the output will be within the minimum and maximum of the world coordinate input. The offset must be equal or greater than the minimum; offset plus H factor must be less than maximum. If output is outside limits, a diagnostic warning is entered, and the output limited.

In CA_Sync Mode, at each Sercos interrupt, the ELS Group output position is looked up in the selected CAM table and scaled with the H factor and the offset is added. If the CAM number is zero, the output of the CAM section will the same as the offset.

If in CA_Local Mode and enabled, the ramp generator will use the linear/rotary programmed accel, decel, and velocity to move its world coordinate input to the target position.

In CA_Local Mode, axis CAM switching takes place immediate, no zero crossing of the master is required.

The switch directs the output of the CAM section or ramp generator to the world input of the kinematic section, or switches configured axis into single axis mode.

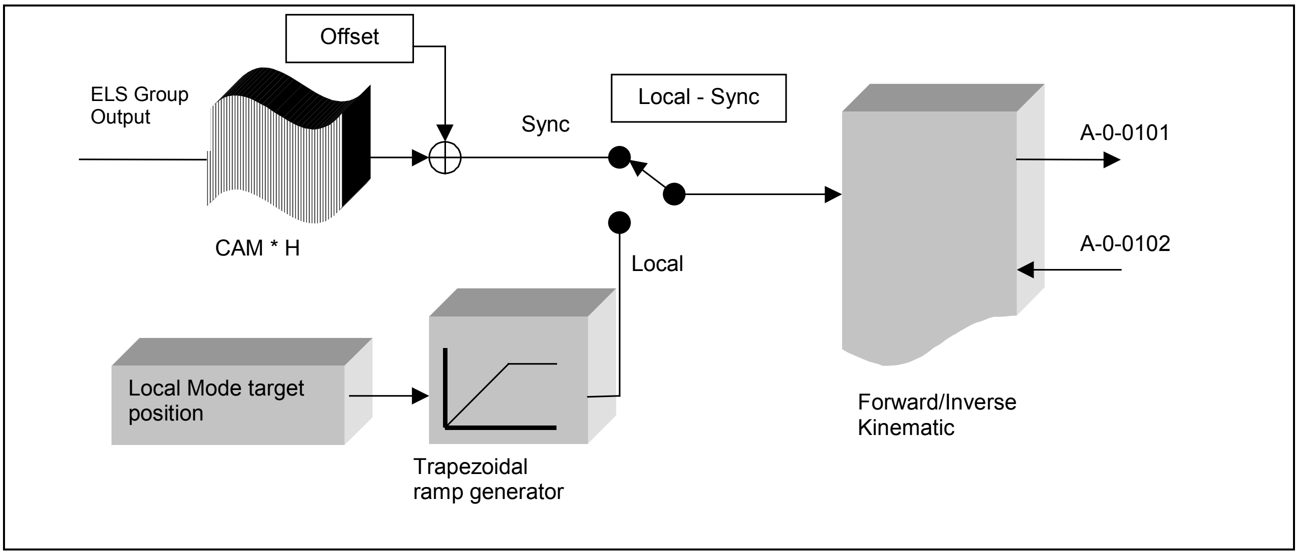


Fig. 6-47: Coordinated Articulated Axis Configuration

Trapezoidal Ramp Generator (Normal Local Mode)

The ramp generator is used for absolute moves when its task is in run.

The target position is checked against minimum and maximum limits at the start of the move (0->1 on Enable Ramp Generator). If the target position is outside the limits, a warning is issued and output limited. The target position, velocity, acceleration, and deceleration are copied at the start of the move; changing these values during the move will not affect the move once started.

An E-STOP, switch to “task manual” or “task stop” during the move will result in ramping down the velocity to zero using the deceleration value that was copied at the start of move.

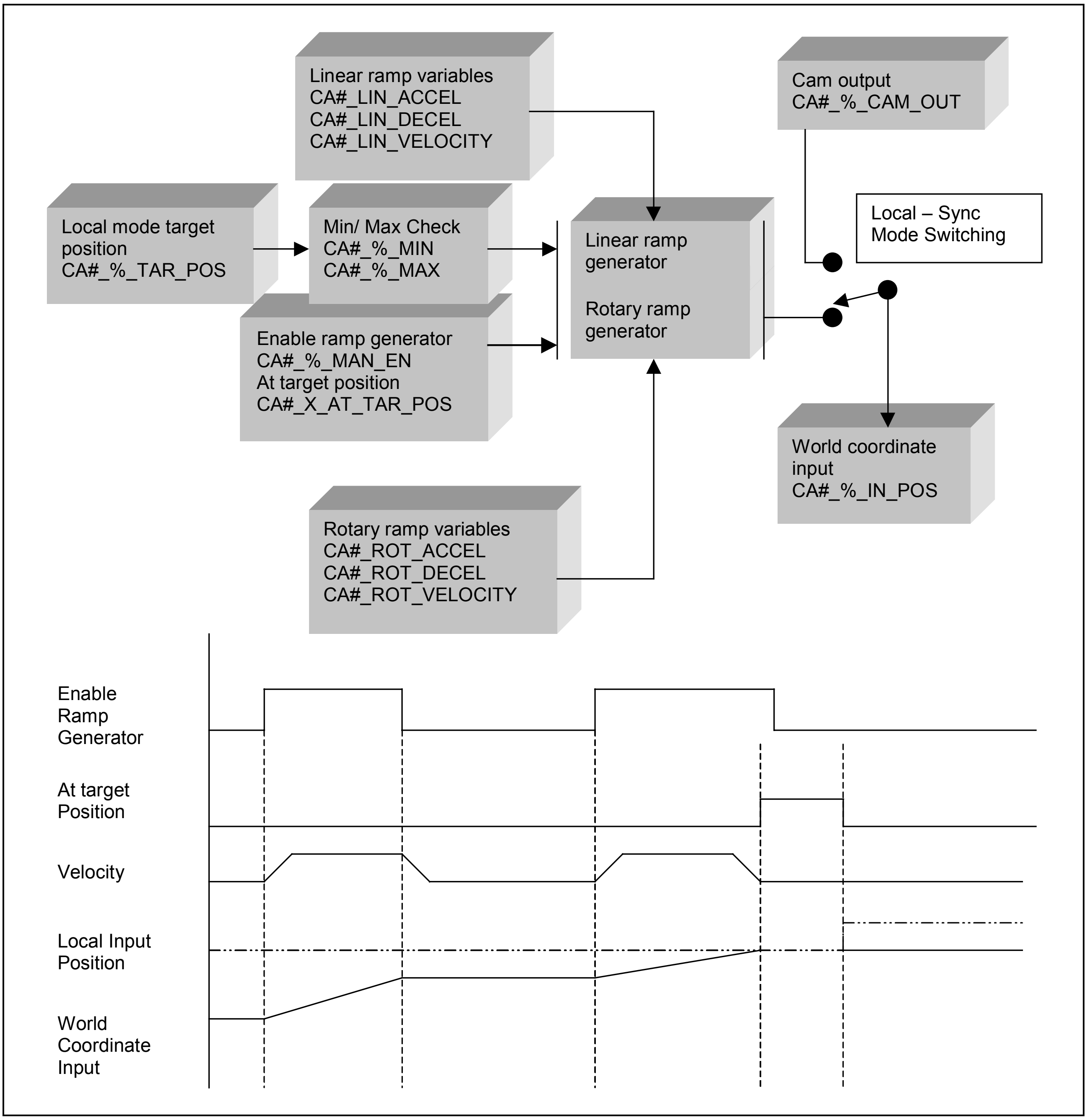


Fig. 6-48: Normal Local Mode

System Considerations

Axis Motion Type

Axis parameter A-0-0003 is set to axis motion type 9 (Coordinated Articulation Axis). This axis motion type has the following configuration (S-0-0258 & S-0-0047 in the MDT):

- Primary Operating Mode: Position Command
- Secondary Operating Mode 1: Single Axis
- Secondary Operating Mode 2: Velocity
- Secondary Operating Mode 3: Torque

Task Control of Coordinated Articulation Axes

All ELS functionality is controlled by task A. All Coordinated Articulation axes are controlled in the task they are declared. Assigned axes primary mode is single axis, secondary mode is position.

Initial State	New State	Comment
2 Parameter	5 Manual	“Single axis” mode
5 Manual	6 Automatic	“Single axis” mode
6 Automatic	7 Task Run	“Single axis” mode. Mode is changed to “position” with “ELS Mode” icon.
7 Task Run	6 Automatic	Axes remain in their current mode, “position” or “single axis”. Icon in “CA_Local” mode, axes are ramped to 0 velocity. Icon in “CA_SYNC” mode, axes remain in ELS Group controlled motion.
6 Automatic	5 Manual	Axes remain in their current mode, “Position” or “Single axis”. If icon in “CA_SYNC” Mode, ELS Group switches to “Local” mode and stops motion.
5 Manual	2 Parameter	“Single axis” mode

Table 6-35: Task Control

Feedback Support

The Coordinated Articulation function supports only the primary feedback device of the drive.

E-Stop of Coordinated Articulation Axes

Axes remain in their current mode, “position” or “single axis”. The ELS Group ramps to a stop in “Local Mode” using stop deceleration.

Control and Status Registers

The following user defined control and status registers are used to control and monitor Coordinated Articulation. These control and status registers are independent of those assigned to Virtual Masters, System Masters and ELS Groups.

Control Registers

Two registers are used to set the mode of operation of an axis. The user assigns the first starting register and VisualMotion assigns the next consecutive register. Default labels are provided by VisualMotion.

Coordinated Articulation supports the following two modes of operation:

- Synchronized Mode (Relative or Absolute)
- Local Mode (Normal or Immediate)

Starting Control Register Number

A Coordinated Articulation configuration uses two control registers. One control register is used for synchronized mode and one for local mode. When a starting control register number is entered, the next consecutive number is used for the local mode control register. The following tables list the functions used in each control register.

Synchronized Mode Control Register

The first control register selects the operation mode (sync or local) for each axis.

Register Bit	Label	Description
1	CA#_X_SYNC	*Synchronized Mode Bits*
2	CA#_Y_SYNC	0 -> 1 transition enables each axis to Sync mode.
3	CA#_Z_SYNC	This transition is delayed until the ramp generator's velocity is 0.
4	CA#_ROLL_SYNC	1 -> 0 transition set each axis to Local Mode.
5	CA#_PITCH_SYNC	Local mode is used for the manual positioning of each axis.
6	CA#_YAW_SYNC	
7	CA#_X_REL_MOD	*Positioning Mode Bits*
8	CA#_Y_REL_MOD	Enable before setting Synchronized Mode Bit.
9	CA#_Z_REL_MOD	1 = Relative Mode
10	CA#_ROLL_REL_MOD	0 = Absolute Mode (default)
11	CA#_PITCH_REL_MOD	
12	CA#_YAW_REL_MOD	

Table 6-36: Synchronized Mode Control Register

Local Mode Control Register

The second control register enables the ramp generator in local mode for each axis. Local mode allows the user to manually position the world coordinate in-position value to the current CAM output position before synchronizing.

Register Bit	Label	Description
1	CA#_X_MAN_EN	*Ramp Generator Enable Bits [used when Sync mode is inactive]*
2	CA#_Y_MAN_EN	**Normal Local Mode**: 0 -> 1 transition enables the ramp generator, creates a trapezoidal move profile for variables CA#_%_TAR_POS value to variable CA#_%_IN_POS (where # = Unit number and % = X, Y, Z, ROLL, PITCH or YAW).
3	CA#_Z_MAN_EN	1 -> 0 transition disables the ramp generator. If a move is at velocity, the velocity is ramped down using the value in variable CA#_LIN_DECEL.
4	CA#_ROLL_MAN_EN	**Immediate Local Mode**:
5	CA#_PITCH_MAN_EN	0 -> 1 transition stores the value of variable A#_%_TAR_POS to variable CA#_%_IN_POS (essentially performing an immediate move).
6	CA#_YAW_MAN_EN	1 -> 0 transition has no effect
7	CA#_X_MAN_IMD	*Immediate Mode Bits [used when Sync mode is inactive]*
8	CA#_Y_MAN_IMD	Enable before setting Local Mode Bit.
9	CA#_Z_MAN_IMD	1 = Immediate Mode
10	CA#_ROLL_MAN_IMD	0 = Normal Local Mode
11	CA#_PITCH_MAN_IMD	
12	CA#_YAW_MAN_IMD	

Table 6-37: Synchronized Mode Control Register

Starting Status Register Number

A Coordinated Articulation configuration uses two status registers. One status register is used for synchronized mode and one for local mode. When a starting status register number is entered, the next consecutive number is used for the local mode status register. The following tables list the functions used in each status registers.

Synchronized Mode Status Register

The first register is used to monitor the status of each axis enabled to synchronized mode. It also monitors the equivalence between variables CA#_%_IN_POS and CA#_%_TAR_POS (where % = X, Y, Z, ROLL, PITCH or YAW).

Register Bit	Label	Description
1	CA#_X_SYNCED	1 = In Sync Mode. Using CAM output world coordinate input value. 0 = In Local Mode. Use target position for world coordinate input value.
2	CA#_Y_SYNCED	
3	CA#_Z_SYNCED	
4	CA#_ROLL_SYNCED	
5	CA#_PITCH_SYNCED	
6	CA#_YAW_SYNCED	
7	CA#_X_READY	1 = CAM output value (CA#_%_CAM_OUT) equals world coordinate input value (CA#_%_IN_POS) and ramp generator is at 0 velocity. 0 = Not ready for synchronization
8	CA#_Y_READY	
9	CA#_Z_READY	
10	CA#_ROLL_READY	
11	CA#_PITCH_READY	
12	CA#_YAW_READY	

Table 6-38: Synchronized Mode Status Register

Local Mode Status Register

The second register is used to monitor the status of CA#_%_IN_POS equal to CA#_%_TAR_POS.

Register Bit	Bit Label	Description
1	CA#_X_AT_TAR_POS	Only valid in Local Mode. (CA#_%_SYNC = 0) 1 = CA#_X_IN_POS at CA#_X_TAR_POS position and ramp generator at 0 velocity. (where % = X, Y, Z, ROLL, PITCH or YAW) 0 = not, goes to 0 when new target position entered. ***NOTE***: This bit is not set to 1 if the ramp generator move is interrupted (dropped Enable, user task stopped, etc.) before the target position is reached. In such an instance, it is not possible to 'force' the bit to 1 by setting the ramp generator's target value equal to the current input value. Rather, another ramp generator move must be completed before this bit is set to 1.
2	CA#_Y_AT_TAR_POS	
3	CA#_Z_AT_TAR_POS	
4	CA#_ROLL_AT_TAR_POS	
5	CA#_PITCH_AT_TAR_POS	
6	CA#_YAW_AT_TAR_POS	

Table 6-39: Synchronized Mode Status Register

Coordinated Articulation Program Variables

The following program variables (20 integer variables, 93 float variables) are used for Coordination Articulation. In the following tables, “#” denotes the unit number.

Integer Variables

The following 20 integers are available for Coordination Articulation. The integer label column displays the default label issued by VisualMotion.

Number Offset	Default Value	Access	Integer Label	Description
0	0	read-only	CA#_NUMBER	Unit number of Coordinated Articulation function 1-4 ***Note***: This feature supports 4 robots however they must be defined in different user tasks.
1	0	read/write in phase 2 read-only in phase 4	CA#_OUT_AXIS1	Number of axis used for output 1
2			CA#_OUT_AXIS2	Number of axis used for output 2
3			CA#_OUT_AXIS3	Number of axis used for output 3
4			CA#_OUT_AXIS4	Number of axis used for output 4
5			CA#_OUT_AXIS5	Number of axis used for output 5
6			CA#_OUT_AXIS6	Number of axis used for output 6
7	0	read/write in any phase	CA#_X_CAM_NUM	Input CAM number for % world coordinate position
8			CA#_Y_CAM_NUM	
9			CA#_Z_CAM_NUM	
10			CA#_ROLL_CAM_NUM	
11			CA#_PITCH_CAM_NUM	
12			CA#_YAW_CAM_NUM	
13	0	read/write in phase 2 read-only in phase 4	CA#_TASK_ID	
14	0	read/write in any phase	CA#_Reserved14	
15	0	Read only	CA#_KINEMATIC	Number of kinematic equation used in transformation
16	1	read/write in phase 2 read-only in phase 4	CA#_ELS_GROUP	ELS Group number used as CAM input
17	0	read/write in any phase	CA#_RESERVED17	
18	0	read/write in any phase	CA#_RESERVED18	
19	0	read/write in any phase	CA#_RESERVED19	

Table 6-40: Program Integers

Float Variables

The following 93 floats are available for Coordination Articulated. The float label column displays the default label issued by VisualMotion.

Number Offset	Default Value	Access	Float Label	Description
0	0.0	read-only	CA#_X_IN_POS	X world coordinate input value in linear units, initialized to 0 at program activation.
1			CA#_Y_IN_POS	Y world coordinate input value in linear units, initialized to 0 at program activation.
2			CA#_Z_IN_POS	Z world coordinate input value in linear units, initialized to 0 at program activation.
3			CA#_ROLL_IN_POS	Roll world coordinate input value in rotary units, initialized to 0 at program activation.
4			CA#_PITCH_IN_POS	Pitch world coordinate input value in rotary units, initialized to 0 at program activation.
5			CA#_YAW_IN_POS	Yaw world coordinate input value in rotary units, initialized to 0 at program activation.
6	0.0	read/write in any phase	CA#_X_TAR_POS	Local mode, X target position for “single axis” move in world coordinates, linear units
7			CA#_Y_TAR_POS	Local mode, Y target position for “single axis” move in world coordinates, linear units
8			CA#_Z_TAR_POS	Local mode, Z target position for “single axis” move in world coordinates, linear units
9			CA#_ROLL_TAR_POS	Local mode, Roll target position for “single axis “move in world coordinates, rotary units
10			CA#_PITCH_TAR_POS	Local mode, Pitch target position for “single axis” move in world coordinates, rotary units
11			CA#_YAW_TAR_POS	Local mode, Yaw target position for “single axis” move in world coordinates, rotary units
12	100.0	read/write in phase 2	CA#_LIN_ACCEL	Local mode, linear acceleration used for “single axis” move(units/sec^2)
13	100.0	read-only in phase 4	CA#_LIN_DECEL	Local mode, linear deceleration used for “single axis” move(units/sec^2)
14	50.0	read/write in phase 4	CA#_LIN_VELOCITY	Local mode, linear velocity used for “single axis” move(units/min)
15	100.0	read/write in phase 4	CA#_ROT_ACCEL	Local mode, rotary acceleration used for “single axis” move(units/Sec^2)
16	100.0	read/write in phase 2	CA#_ROT_DECEL	Local mode, rotary deceleration used for “single axis” move(unitss/Sec^2)
17	50.0	read-only in phase 4	CA#_ROT_VELOCITY	Local mode, rotary velocity used for “single axis” move(RPM)
18	0.0	read-only	CA#_X_CAM_OUT	(CAM[#]*H) + Offset), for % world coordinate
19			CA#_Y_CAM_OUT	
20			CA#_Z_CAM_OUT	
21			CA#_ROLL_CAM_OUT	
22			CA#_PITCH_CAM_OUT	
23			CA#_YAW_CAM_OUT	

Table 6-41: Program Floats (1 of 2)

Rexroth
Bosch Group

Number Offset	Default Value	Access	Float Label	Description
24	1.0	read/write in any phase	CA#_X_CAM_H	H CAM factor for % world coordinate
25			CA#_Y_CAM_H	
26			CA#_Z_CAM_H	
27			CA#_ROLL_CAM_H	
28			CA#_PITCH_CAM_H	
29			CA#_YAW_CAM_H	
30	0.0	read/write in any phase	CA#_X_CAM_OFF	Offset for % world coordinate
31			CA#_Y_CAM_OFF	
32			CA#_Z_CAM_OFF	
33			CA#_ROLL_CAM_OFF	
34			CA#_PITCH_CAM_OFF	
35			CA#_YAW_CAM_OFF	
36	-100.0	read/write in phase 2 read-only in phase 4	CA#_X_MIN	Minimum value for % world coordinate
37			CA#_Y_MIN	
38			CA#_Z_MIN	
39	-180.0		CA#_ROLL_MIN	
40			CA#_PITCH_MIN	
41			CA#_YAW_MIN	
42	100.0	read/write in phase 2 read-only in phase 4	CA#_X_MAX	Maximum value for % world coordinate
43			CA#_Y_MAX	
44			CA#_Z_MAX	
45	180.0		CA#_ROLL_MAX	
46			CA#_PITCH_MAX	
47			CA#_YAW_MAX	
48 … 86	0.0	read/write in phase 2 read-only in phase 4	CA#_KINEMATIC_V1 through CA#_KINEMATIC_V39	Kinematic setup data value 1 through Kinematic setup data value 39
87	0.0	read-only	CA#_X_FDBK_POS	X world coordinate forward transform value in linear units, updated every Sercos cycle in Phase 4.
88			CA#_Y_ FDBK _POS	Y world coordinate forward transform value in linear units, updated every Sercos cycle in Phase 4.
89			CA#_Z_ FDBK _POS	Z world coordinate forward transform value in linear units, updated every Sercos cycle in Phase 4.
90			CA#_ROLL_ FDBK _POS	Roll world coordinate forward transform value in rotary units, updated every Sercos cycle in Phase 4.
91			CA#_PITCH_ FDBK _POS	Pitch world coordinate forward transform value in rotary units, updated every Sercos cycle in Phase 4.
92			CA#_YAW_ FDBK _POS	Yaw world coordinate forward transform value in rotary units, updated every Sercos cycle in Phase 4.

Table 6-42: Program Floats (2 of 2)

Initializing and Synchronizing Coordinated Articulation

Note: The following procedure is provided as a guide in demonstrating the necessary steps to initialize and synchronize a Coordinated Articulated system. It may not contain all the steps required for your specific application.

1. Ensure that all configured axes are in single axis mode by using a Mode Change icon.
2. Set *CA#_%_SYNC* to 0 (Local Mode)
3. Set each axis target position variable *CA#_%_TAR_POS* to a known world coordinate machine reference position.
4. Set *CA#_%_MAN_IMD* to 1 (Immediate Local Mode)
5. Transition the manual enable bit CA#_%_MAN_EN from 0 ⇒ 1 to immediately store the value of *CA#_%_TAR_POS* to *CA#_%_IN_POS*.

[The value of *CA#_%_IN_POS* is passed through the Coordinated Articulation Kinematic and the resulting value is written to axis commanded position parameter A-0-0101]

6. In single axis, use a Move icon to position each configured axis to the calculated axis commanded position A-0-0101 and wait for moves to complete.
7. Using a Change Mode icon, Synch to Master.

[Now that all axes are positioned to a known machine reference point, the Articulated Coordinated axes can now be synchronized to the Coordinated Articulation CAM output]

8. Copy the CAM output position variable, CA#_%_CAM_OUT, to the local mode target position variable CA#_%_TAR_POS to synchronize each axis position to the CAM output position.
9. Disable immediate local mode by setting *CA#_%_MAN_IMD* to 0. (Normal Local Mode)
10. Transition the manual enable bit CA#_%_MAN_EN from 0 ⇒ 1 to now synchronize the coordinated axes, using the ramp generator, to the Coordinated Articulation CAM output and wait for coordinated move to complete.
11. Now set the Coordinated Articulation function to Sync mode by transiting CA#_%_SYNC from 0 ⇒ 1 and run program.

Kinematics for Standard Coordinated Motion

In addition to standard linear and circular segments, VisualMotion supports the execution of forward and reverse kinematic movement by using an application specific library of kinematic functions. Kinematics are developed by Bosch Rexroth to customer specifications.

Coordinated Jogging

When a drive is referenced, the G*P 11 control firmware detects the travel limits for kinematic 1 only. Once started, jogging automatically stops before the drive travel limit is reached. For all other kinematics, if the drive is referenced, jogging will continue if commanded until the drive issues a travel limit fault.

Coordinated axes may be jog in any direction while a task is in manual mode by using Task Jog Registers 7-10. Refer to chapter 14, *Registers*, in volume 1 of the *Rexroth VisualMotion 11 Functional Description*.

Kinematic 1

This kinematic represents a standard 3 axes gantry (up to 3 axes, Cartesian coordinates X, Y, and Z) arrangement used with basic coordinated motion applications not requiring complex paths.

Note: Task parameters T-0-0050 through T-0-0059 are not used with kinematic #1.

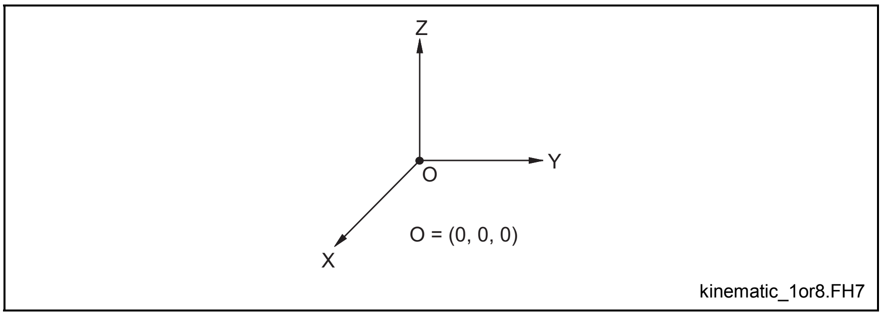


Fig. 6-49: Kinematic 1 Graphic

Kinematic 2

This kinematic represents a 2 axes (C1 and C2) parallelogram robot.

Task Parameters	Kinematic Constants	Default Values	Units
T-0-0050	K1	33.0	Inches
T-0-0051	K2	34.5	Inches
T-0-0052	K3	12.0	Inches
T-0-0053	K4	34.7279	Inches
T-0-0054	K5	6.0	Inches
T-0-0055	K6	8.0	Inches

Fig. 6-50: Kinematic 2 Task Parameters Values

Note: Kinematic 2 requires special mechanical setup. Refer to Special Case Kinematics on page 6-74 for details.

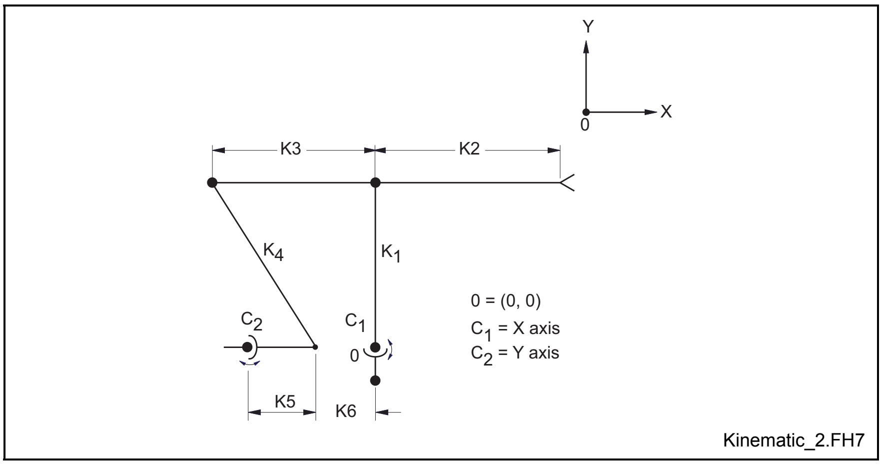


Fig. 6-51: Kinematic 2 Graphic

Kinematic 3

This kinematic represents a 2 axes (C1 and C2) XY tandem motor robot.

Note: Task parameters T-0-0050 through T-0-0059 are not used in kinematic 3.

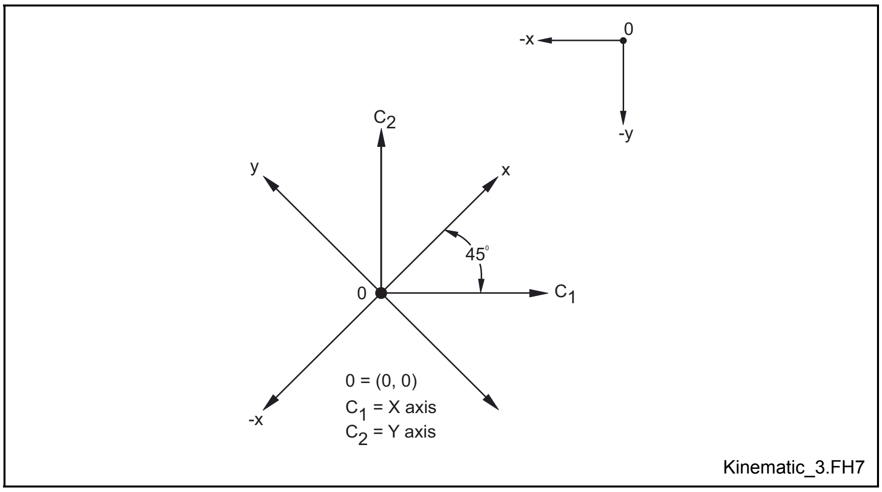


Fig. 6-52: Kinematic 3 Graphic

Kinematic 4

This kinematic represents a 2 axes (C1 and C2) parallelogram robot.

Task Parameters	Constants	Default Values	Units
T-0-0050	K1	850.0	mm
T-0-0051	K2	850.0	mm
T-0-0052	K3	200.0	mm
T-0-0053	K4	850.0	mm
T-0-0054	K5	200.0	mm

Fig. 6-53: Kinematic 4 Task Parameters Values

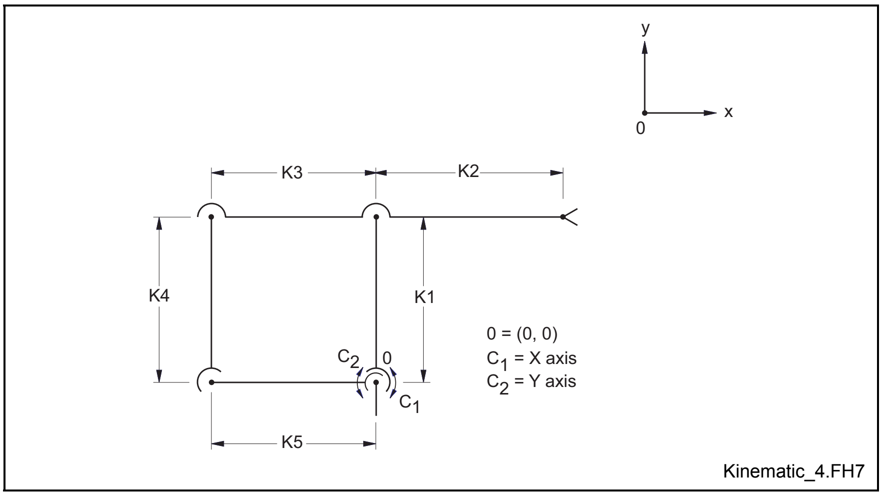


Fig. 6-54: Kinematic 4 Graphic

Note: Kinematic 4 requires special mechanical setup. Refer to Special Case Kinematics on page 6-74 for details.

Kinematic 5

This kinematic represents a 2 axes (C1 and C2) XY tandem motor robot.

Task Parameters	Constants	Default Values	Units
T-0-0050	K1	12.0	Inches
T-0-0051	K2	12.0	Inches
T-0-0052	K3	13.0	Inches
T-0-0053	K4	13.0	Inches
T-0-0054	K5	12.0	Inches
T-0-0055	K6	8.0	Inches
T-0-0056	K7	3.5	Inches
T-0-0057	K8	4.0	Inches

Fig. 6-55: Kinematic 5 Task Parameters Values

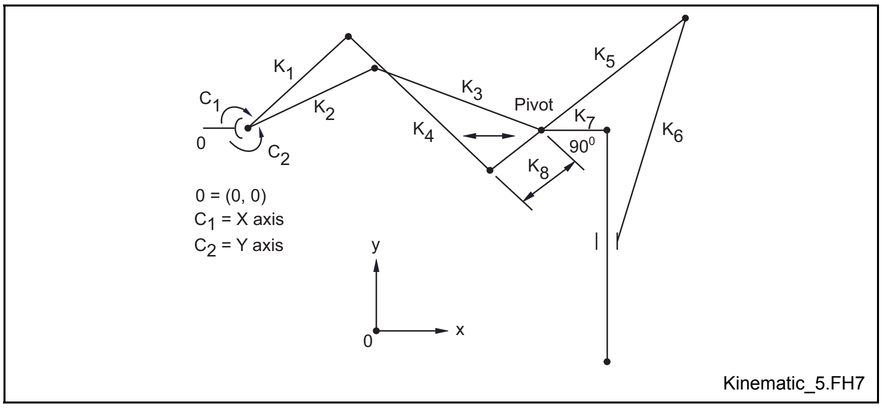


Fig. 6-56: Kinematic 5 Graphic

Note: Kinematic 5 requires special mechanical setup. Refer to Special Case Kinematics on page 6-74 for details.

Kinematic 8 (with Velocity Precision)

This Kinematic represents a 3 axes (X, Y, and Z) Gantry robot.

Note: Task parameters T-0-0050 through T-0-0059 are not used in kinematic 8.

The rate field in kinematic 8 is used to more precisely enter the velocity for each commanded point. The *%Sp* (percentage of speed) field does not have any affect in speed control.

Note: If a zero value is used for rate, the maximum path velocity (T-0-0020) will be used.

The path velocity is a resultant of all the individual axes' velocities for a kinematic. The control continuously calculates the individual axes' velocities to accomplish the programmed path. If the calculation for a given axis exceeds parameter A-0-0020, then the resultant path velocity for that given path segment will be scaled down.

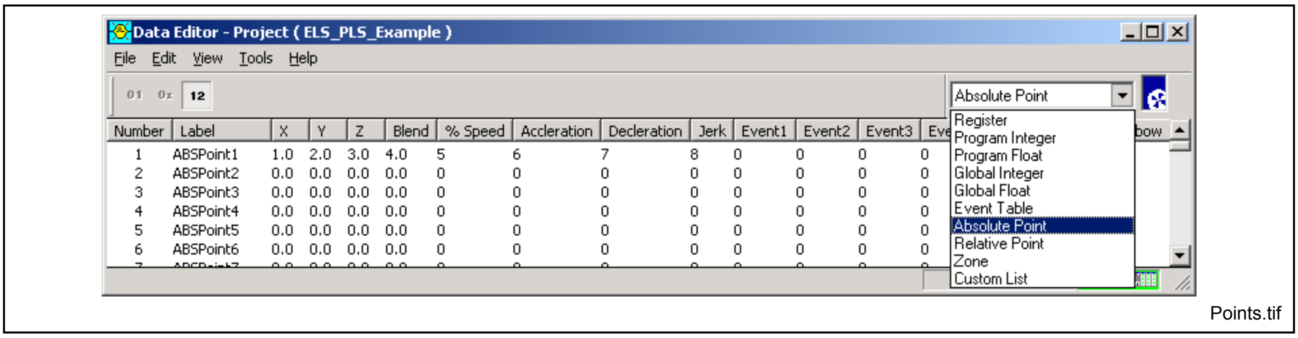


Fig. 6-57: Example Points Table

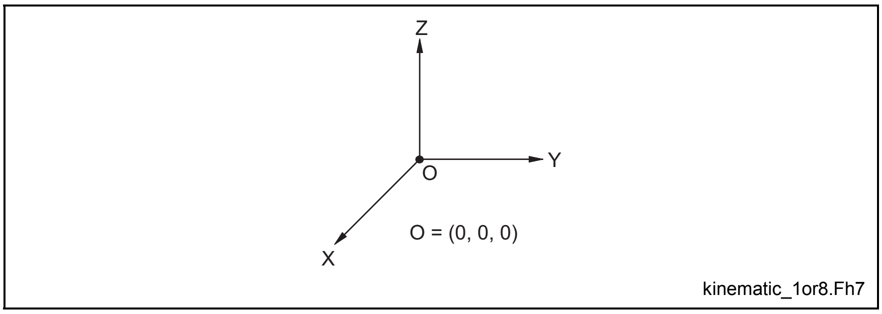


Fig. 6-58: Kinematic 8 Graphic

Kinematic 9

This Kinematic represents a 2 axes (C1 and C2) articulated arm robot.

Task Parameters	Constants	Default Values	Units
T-0-0050	K1	35.669	Inches
T-0-0051	K2	35.669	Inches

Fig. 6-59: Kinematic 9 Task Parameters Values

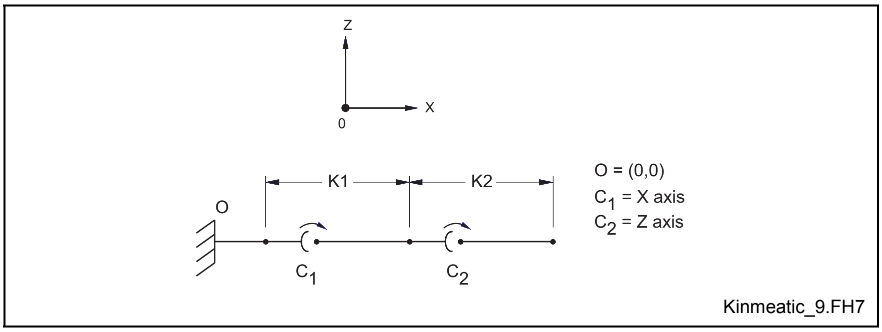


Fig. 6-60: Kinematic 9 Graphic

Note: Kinematic 9 requires special mechanical setup. Refer to Special Case Kinematics on page 6-74 for details.

Kinematic 10

This Kinematic represents a 3 axes (X, Y and Wrist) robot.

Note: Task parameters T-0-0050 through T-0-0059 are not used in kinematic 10.

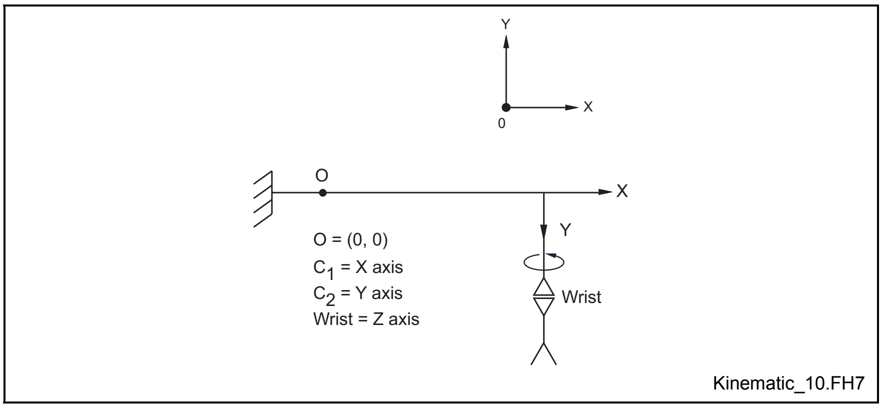


Fig. 6-61: Kinematic 10 Graphic

Note: Jogging of kinematic 10 is controlled by the following bits of Task Jog registers 7-11:

- The X-axis is jogged using bit 9 (Jog_X_Coord)
- The Y-axis is jogged using bit 10 (Jog_Y_Coord)
- The Z-axis (Wrist) is jog using bit 14 (Jog_Joint_6)

Kinematic 12

This Kinematic represents a 2 axes (X and Z) Loader robot.

Task Parameters	Constants	Default Values	Units
T-0-0050	K1	0.0	mm
T-0-0051	K2	711.200	mm
T-0-0052	K3	1642.28	mm
T-0-0053	K4	0.0	mm
T-0-0054	K5	0.0	mm

Fig. 6-62: Kinematic 12 Task Parameters Values

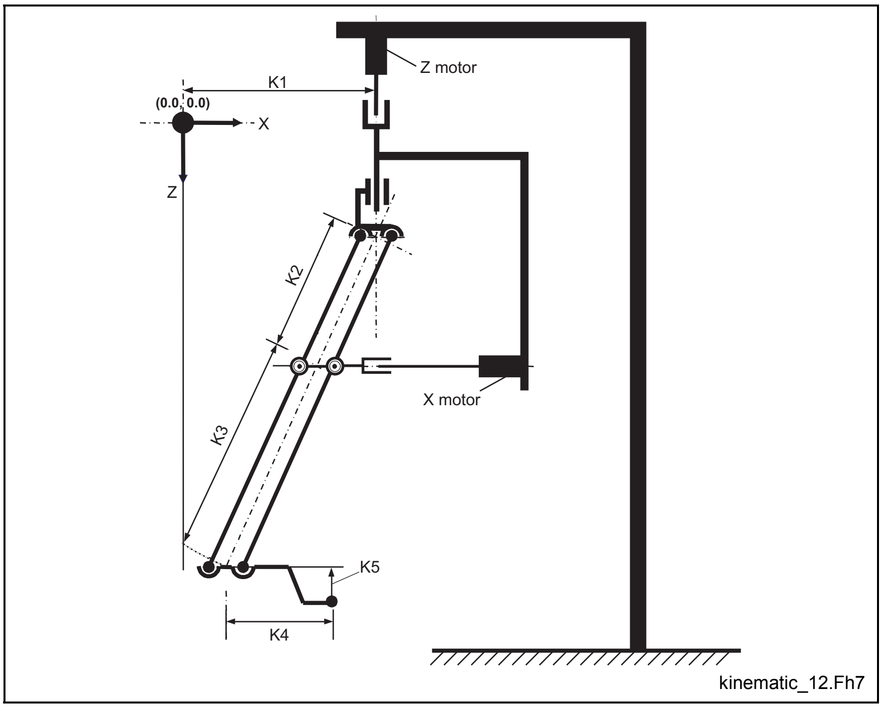


Fig. 6-63: Kinematic 12 Graphic

Kinematics for Coordinated Articulation

This section describes the available kinematic libraries that are support by VisualMotion's Coordinated Articulation functionality. Kinematics are developed by Bosch Rexroth to customer specifications.

Kinematic 13

Kinematic 13 is a pass through kinematic. It can be used for up to 6 axes in an Coordinated Articulation application. The target position value for each axis must be assigned in the respective float variable CA#_%_TAR_POS (where % is x, y, z, roll, pitch or yaw). There is no output kinematic resolving the position of the axes based on a coordinated move.

This kinematic is not assigned in the axis icon but in the *Coordinated Articulation* icon in the *Kinematic Setup* button. Since this kinematic is a pass through, any values assigned as Kinematic Data has no effect on the output. The float variable values assigned are pass through to each axis.

Kinematic 14

Kinematic 14 is a customer specific kinematic used for new Hexapod robot.

Kinematic 15

Kinematic 15 is customer specific and proprietary.

Kinematic 16

This Kinematic represents a 2 axes (X and Z) Loader robot used in a Coordinated Articulation application.

Task Parameters	Constants
CA#KINEMATIC_V1	K1
CA#KINEMATIC_V2	K2
CA#KINEMATIC_V3	K3
CA#KINEMATIC_V4	K4
CA#KINEMATIC_V5	K5

Fig. 6-64: Kinematic 16 Task Parameters Values

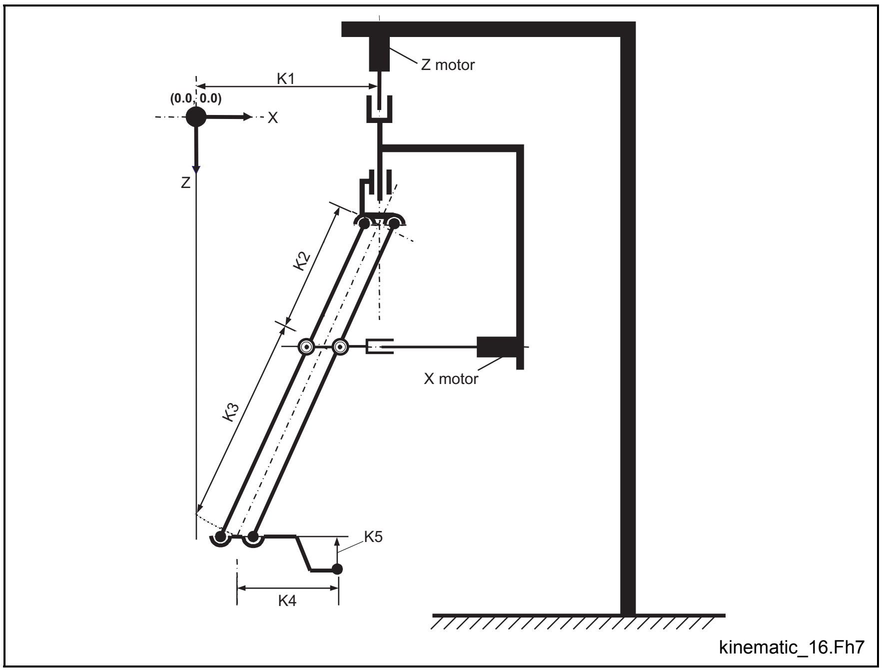


Fig. 6-65: Kinematic 16 Graphic

Kinematic 18

This kinematic represents a 2 axes (C1 and C2) XY tandem motor robot used in a Coordinated Articulation application.

Note: Task parameters T-0-0050 through T-0-0059 are not used in kinematic 18.

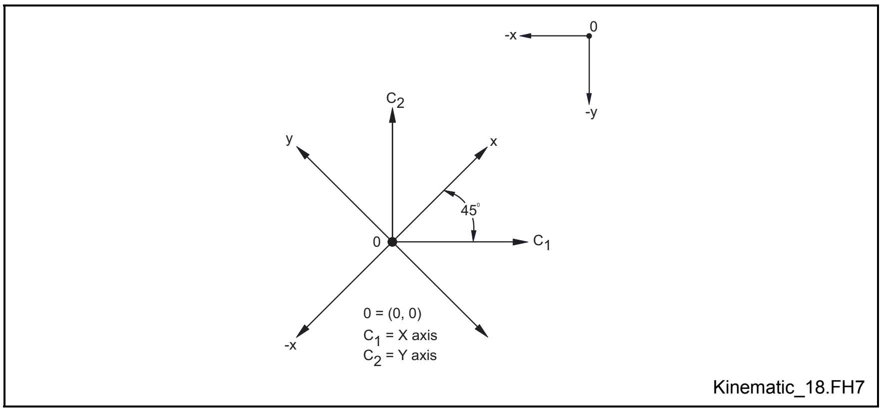


Fig. 6-66: Kinematic 18 Graphic

7 Commissioning Tools

7.1 IndraLogic

IndraLogic is an IEC 61131-3 programming software tool used for programming the PLC logic portion of the VisualMotion motion and logic system. IndraLogic can be used for varying amounts of programming, from a simple I/O interface to VisualMotion registers, to a complete PLC project. Using supported motion functions and function blocks, IndraLogic can provide a complete motion programming solution for single axis or ELS applications.

The following five IEC 61131-3 programming languages are supported:

- Instruction List (IL)
- Ladder Diagram (LD)
- Function Block Diagram (FBD)
- Sequential Function Chart (SFC)
- Structured Text (ST)

In addition to the IEC specific languages, IndraLogic supports the following product specific programming language:

- Continuous Function Chart (CFC)

The following figure illustrates an overview of the VisualMotion motion and logic system:

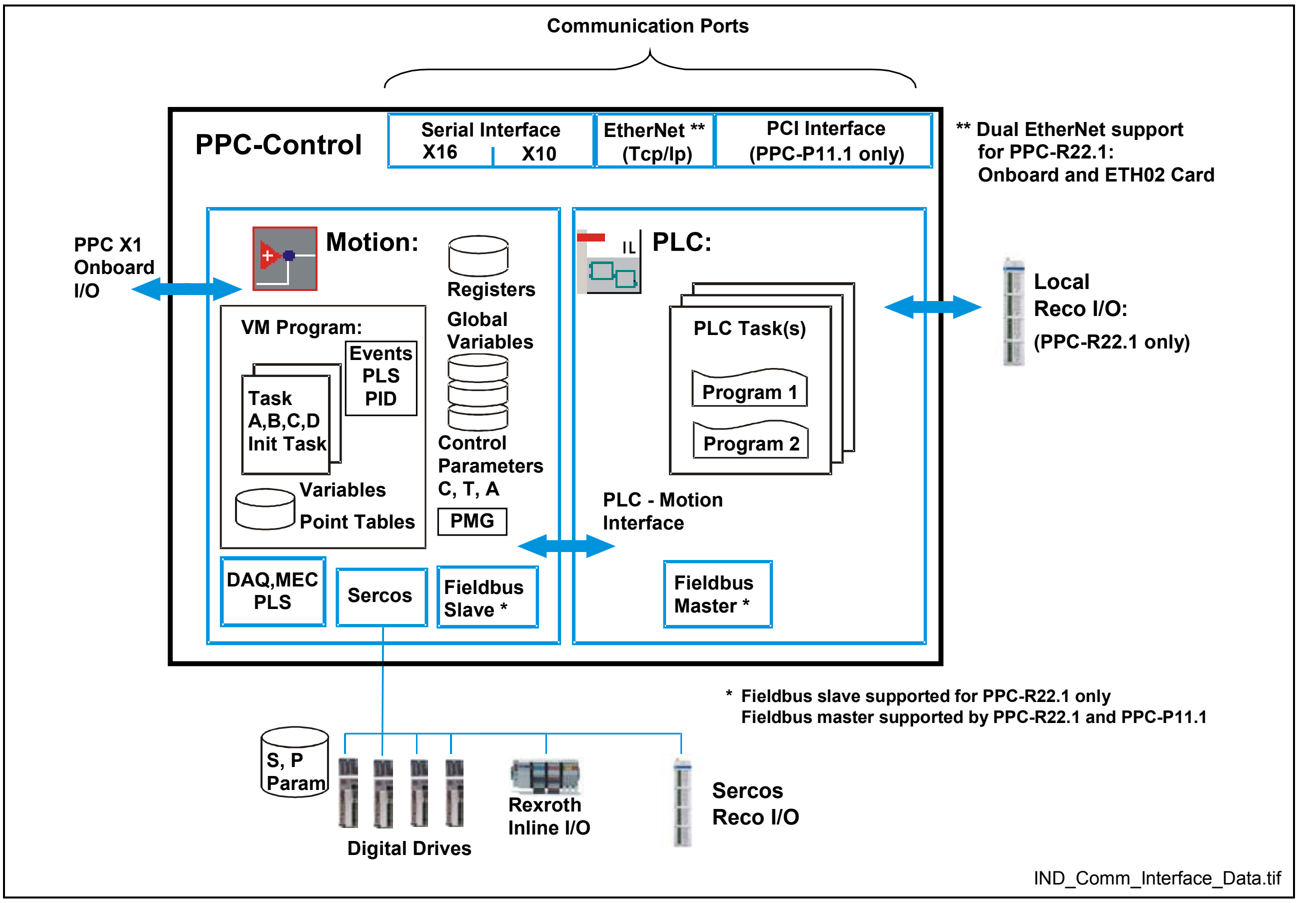


Fig. 7-1: Overview Communication Interface and Data Management

IndraLogic is used to configure and monitor the following devices:

- Local Reco 02 I/O modules (PPC-R22.1 only)
- Profibus fieldbus master
- DeviceNet fieldbus master

Refer to Configuring a Profibus Fieldbus Master on page 7-12 and Configuring a DeviceNet Fieldbus Master on page 7-38 for details.

VisualMotion Toolkit is used to configure and monitor the following devices:

- Rexroth IndraDrive, Diax 04, and EcoDrive 03 digital drives
  - Diax04 and EcoDrive 03 I/O modules
- Sercos Reco I/O modules
- Sercos Inline I/O modules
- Fieldbus Slave support with PPC-R22.1
  - Profibus slave fieldbus
  - DeviceNet fieldbus slave
  - Interbus fieldbus slave
  - EtherNet/IP fieldbus slave (onboard and ETH02 card interfaces)
- Cross communication (DAQ)
- Option Card PLS
- Encoder Interface Card (MEC)
- PCI bus interface (PPC-P11.1 only)

IndraLogic Project Definition

An IndraLogic project contains the following programmable logic controller (PLC) program objects.

- Program Organization Units (POUs)
- Data Types
- Visualizations
- Resources
- Libraries

Detailed information about objects is provided in the document, *PLC Programming with Rexroth IndraLogic 1.0*. A PLC project is stored in the VisualMotion project folder on the PC's hard drive using the following file extension:

- <projectname>**.pro**

The PLC project name is assigned the same project named used in the associated VisualMotion project when IndraLogic is launched from VisualMotion Toolkit.

Task Scheduling in Sercos Cycle

The following figure indicates the priority of the PLC time slot in the processing sequence of a Sercos cycle:

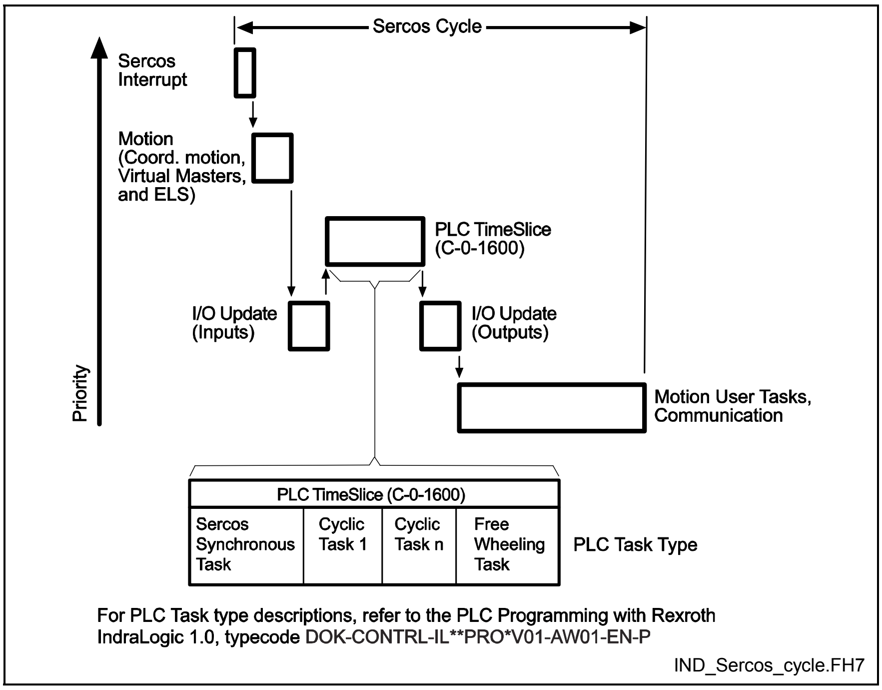


Fig. 7-2: PLC Timeslice in Sercos Cycle

The PLC Timeslice is set at a default value of 25% and can have a maximum value of 75% of the overall Sercos cycle. The percentage setting of the Sercos Timeslice for the PLC is configured in the *Control Settings* window in VisualMotion Toolkit. This setting is stored in control parameter C-0-1600.

Note: The Sercos Timeslice should be increased to 50% for motion related PLC programs.

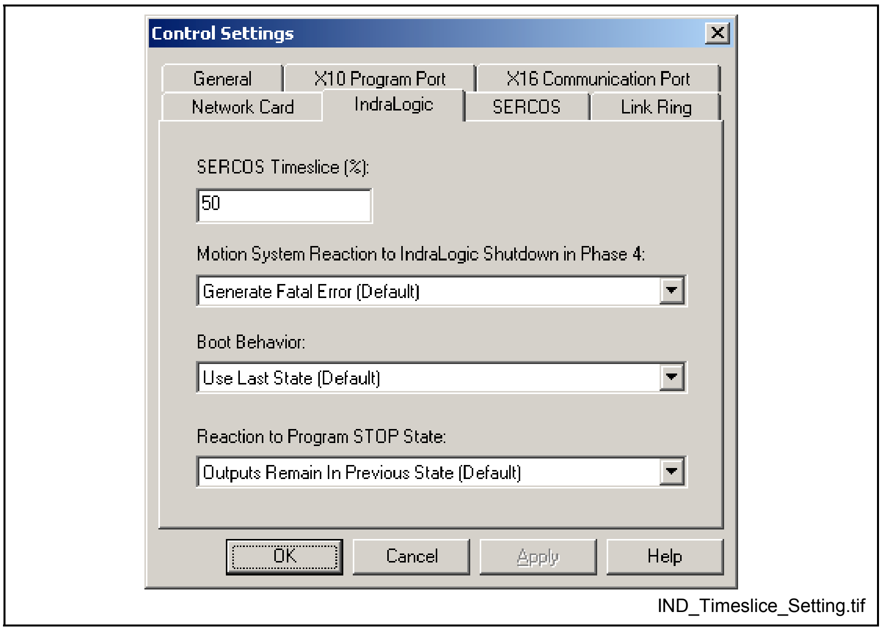


Fig. 7-3: Sercos Timeslice Setting

Rexroth Bosch Group

Note: The Sercos Timeslice value (C-0-1600) is the requested time slice. The system ensures that adequate time is available for processing the motion and logic Sercos interface. In rare instances, the actual PLC processing time available could be limited.

Create a POU in IndraLogic

Before a PLC program can be downloaded to a target (PPC control), it must contain at least one POU (Program Organization Unit) with some data.

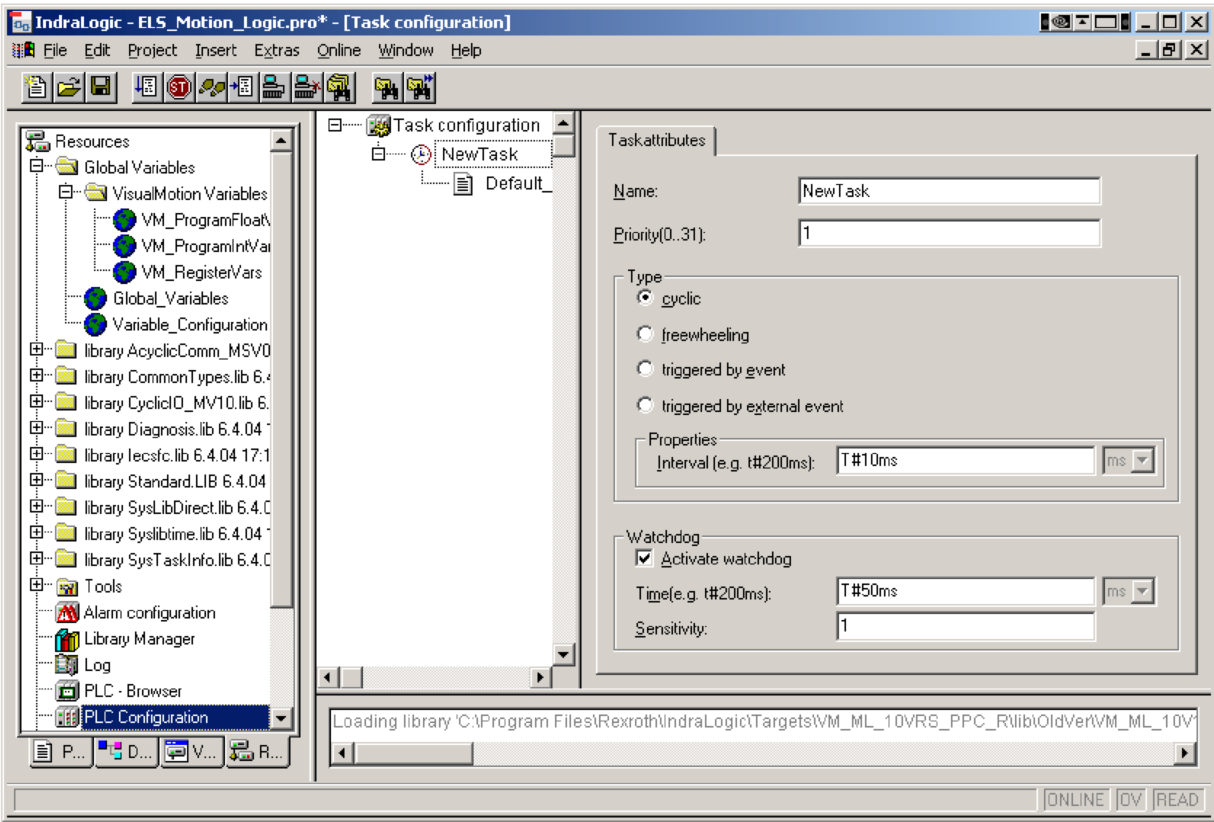
Refer to the *PLC Programming with Rexroth IndraLogic 1.0, typecode DOK-CONTRL-IL**PRO*V01-AW01-EN-P*.

Creating a PLC Task

A PLC task determines when a POU is executed. Tasks must be associated with POUs in your PLC program. Refer to the IndraLogic User manual, DOK-CONTRL-IL**PRO*V01-AW01-EN-P, and IndraLogic online help system for details.

To create a task:

1. Double-click **Task Configuration** under the *Resource* tab.
2. Right-click *Task Configuration* and select ***Append Task***.
3. Set the properties of the task under the Task attributes tab.
4. Enter a **Name** for the Task and set a **Priority** from 0-31 (0 being the highest priority). Refer to Table 7-2: Recommended Task Settings for priority details.



IND_Taskattributes.tif

Fig. 7-4: New Task Configuration

5. Select the task type from one of the following selections:

Task Type	Description
Cyclic	a triggered task that may run to completion within the defined time interval or continue at the next trigger
Freewheeling	a task with no time limit for completion that runs continuously, beginning again immediately following completion
Triggered by Event	the task starts as soon as the system event, which is defined in the *Properties* section, occurs
Triggered by External Event	a time driven task called Sercos Event that runs to completion within one Sercos cycle

Table 7-1: Task Configuration Types

6. The Watchdog time and sensitivity should be activated and set to a value that corresponds to the selected task type and application requirements.

Note: The setting of the Watchdog time and sensitivity is strongly recommended.

7. The following recommendations should be considered:

Task Type	Recommended Watchdog Settings		Task Priority
	Time	Sensitivity	
Cyclic	Set watchdog time to equal the Interval time. e.g. Interval = T#10ms, Watchdog = T#10ms	Set to 1	Less than Sercos Cyclic Task, Greater than Freewheeling
Freewheeling	Set watchdog time to the worst case time duration of the actual freewheeling task used in the application.	Set to 0 or 1	Lowest of all PLC tasks
Triggered by event	similar to freewheeling task		
Triggered by external event (Sercos Cyclic Task)	Set watchdog time to the Sercos cycle time	Set to 0 **	Highest of all PLC tasks
** The sensitivity is set to assume that the Sercos cyclic task is to run to completion every Sercos cycle.			

Table 7-2: Recommended Task Settings

Associate the Task to a POU:

1. Right-click the Task and select **Append Program Call**.
2. Type the name of the program call in the Program Call tab or select the ellipses button to open the *Input Assistant* window to select a POU.

Input Assistant

When opened for a Program Call, the *Input Assistant* window will list all available POUs in the project. The window can also be accessed by pressing the F2 function key.

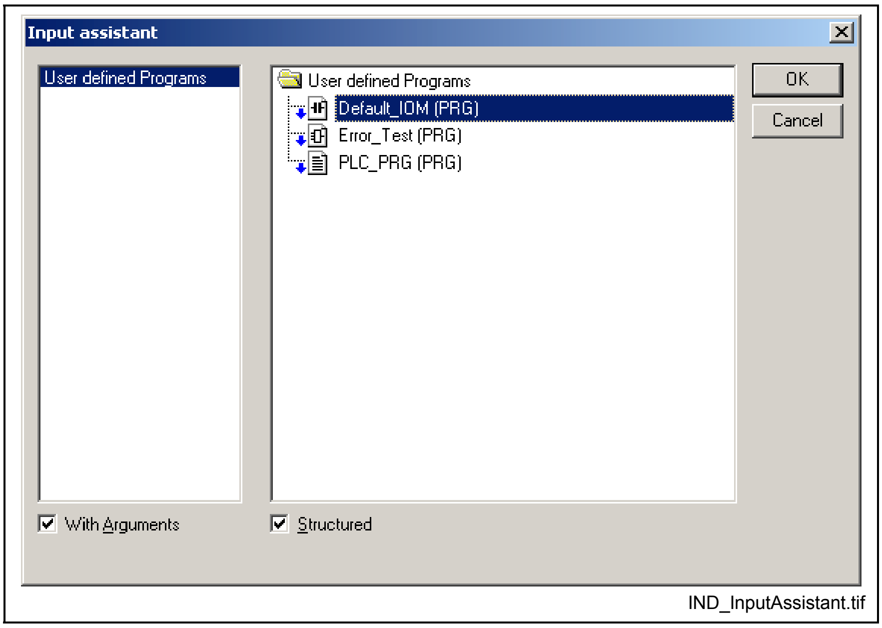


Fig. 7-5: Input Assistant Window

Creating a Boot Project in IndraLogic

A boot project is a compiled PLC program downloaded to the control and stored in flash along with the motion project. This allows the PLC project to automatically start when the PPC control is switched to online mode.

A boot project can be created manually by the user or configured in IndraLogic to be automatically created and downloaded to the control every time the PLC is switched online.

Manual Boot Project Creation

While IndraLogic is logged on and running, select ***Online ⇒ Create boot project***.

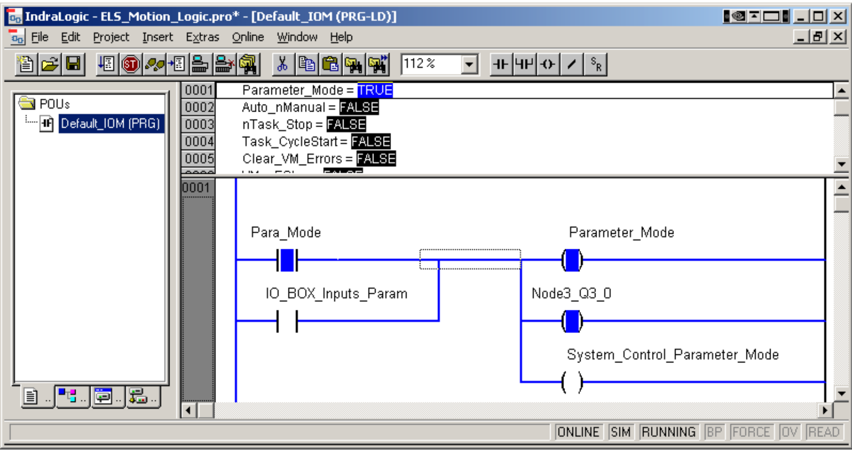
Automatic Boot Project Creation

To have IndraLogic automatically download the current PLC project to the control as a boot project, select the *Resource* tab (located under the tree navigator). Next, double click on **Target Settings** to open the *Target Settings* window. Under the **General** tab, check the box next to "**Load bootproject automatically**".

Simulation Mode

Simulation mode allows you to test your IndraLogic project without actually communicating with your PLC target. In this mode, **only logic functionality will run, motion functionality will be ignored**. Use the following steps, to use simulation mode:

1. Select ***Online ⇒ Simulation Mode*** in IndraLogic.
2. Bring the PLC online by clicking on the **Login** icon or by selecting ***Online ⇒ Login***.
3. Start the PLC project, by clicking on the **Run** icon or by selecting ***Online ⇒ Run***.

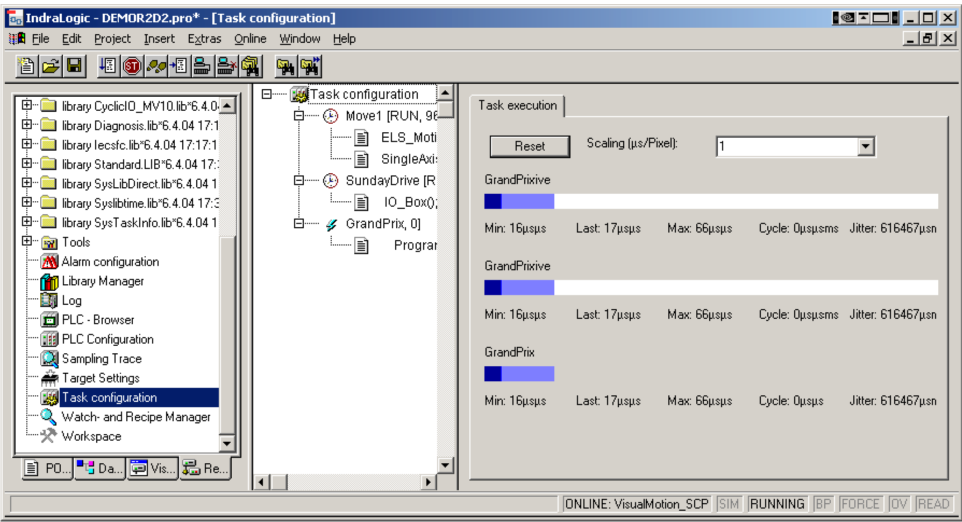


SimMode.tif

Fig. 7-6: Running in Simulation Mode

Monitoring Task Execution

IndraLogic task configurations can be monitored while IndraLogic is online and a PLC program is running. Double-click on **Task configuration** under the *Resource* tab and click on the Task configuration icon with the left mouse button. Refer to "Taskconfiguration in Online Mode" in the IndraLogic Help System for details. The following figure illustrates an example of task execution.



TaskExecution.tif

Fig. 7-7: Task Execution

Rexroth
Bosch Group

IndraLogic Integration with VisualMotion

VisualMotion Toolkit provides an interface for configuring the following data and communication settings shared between VisualMotion and IndraLogic:

- Program variables, registers, and bits that are shared with IndraLogic as SysLibDirect global variables
- Target Libraries for support of functions and function blocks based on control firmware version
- Communication parameters used between motion and logic for configuring the PLC Configuration in IndraLogic

Before launching IndraLogic, select ***Tools ⇒ Options…*** in VisualMotion Toolkit and set the above mentioned options under the IndraLogic tab. Refer to chapter 13, *Menu Descriptions*, in volume 2 of the *VisualMotion Functional Description* for details.

Once the initial settings are set in VisualMotion Toolkit, launch IndraLogic by selecting ***Commission ⇒ IndraLogic…***

PLC Configuration

When IndraLogic is launched from VisualMotion Toolkit, the PLC Configuration in IndraLogic is configured based on the selected target.

Note: PLC Configurations are saved with the setting of the original project target. If the VisualMotion project target is changed, (i.e., GPP 11 to GMP 11), an empty default PLC Configuration is created and would require the user to manually re-create the PLC Configuration. Refer to Managing IndraLogic Files on page 7-57 for recommended procedures for archiving and restoring project files.

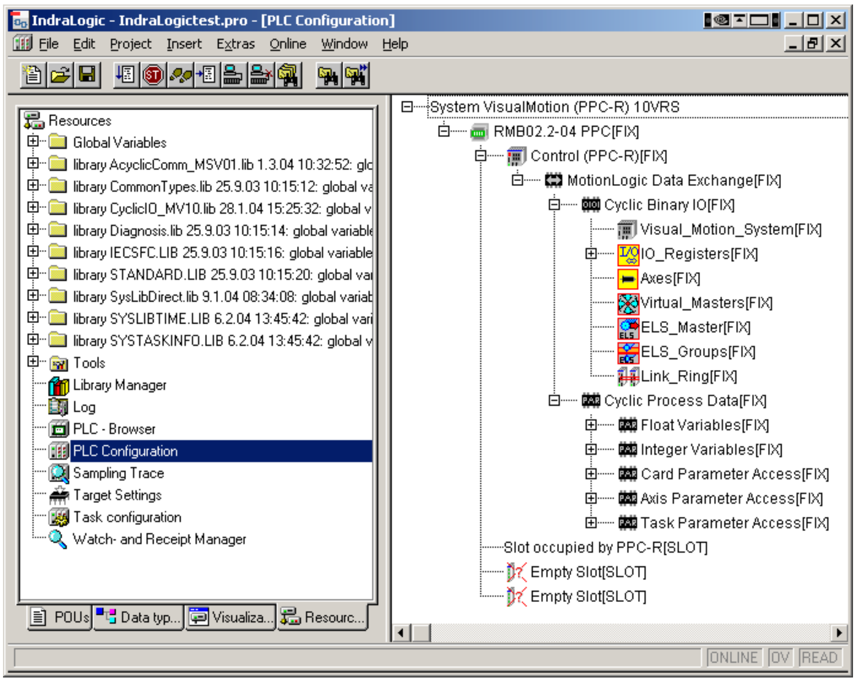
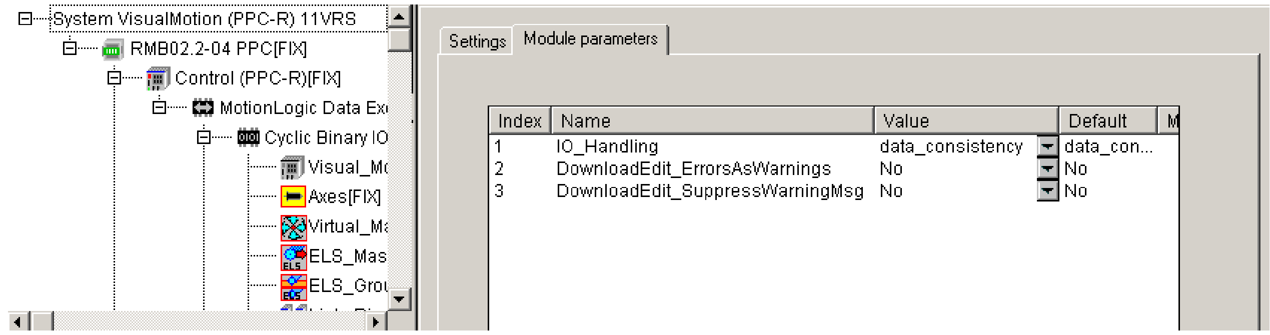


Fig. 7-8: IndraLogic PLC Configuration

Any LocalReco 02 I/O modules installed along with the PPC-R22.1 control in the RMB02.2 rack must be manually configured. Refer to Configuring Local Reco 02 I/O Modules in IndraLogic on page 7-11 for details.

Configuring PLC System Behavior

The handling of I/O data consistency between all PLC tasks and the I/O image as well as the ability to download online changes is configured within the **PLC Configuration** tab.



PLCSystemSetting.tif

Fig. 7-9: PLC Configuration Settings

Use the following steps to configure PLC system behavior:

1. Select *PLC Configuration* from the *Resources* tab and click on **System VisualMotion (PPC) 11VRS**.
2. Select the **Module parameters** tab and set the following options to meet the application requirements:

Name	Value	Default	Meaning
IO_Handling	data_consistency	X	All PLC tasks use the same data from a single I/O image instance. This ensures data consistency across multiple tasks accessing the same data.
	task_priority		I/O image data values are captured as each task accessing the data runs. This setting does not guarantee data consistency but can be more efficient, especially for large I/O images.
DownloadEdit_ErrorsAsWarnings	No	X	This setting does not allow online PLC changes to be downloaded to the control when the drives are enabled and in motion. Refer to Fig. 7-10: Online Change Error.
	Yes		Online changes are possible even with the drives enabled. However, a warning message is displayed requiring user input before proceeding. Refer to DownloadEdit_SuppressWarningMsg.
DownloadEdit_SuppressWarningMsg	No	X	Warning message is displayed when DownloadEdit_ErrorsAsWarnings is set to "Yes" Refer to Fig. 7-11: Online Change Warning
	Yes		Warning message is not displayed and download proceeds without user input when DownloadEdit_ErrorsAsWarnings is set to "Yes"

Table 7-3: PLC Configuration System Behavior

Note: Additional PLC system behavior settings can be set in VisualMotion Toolkit by selecting ***Tools ⇒ Control Settings*** and selecting the **IndraLogic** tab.

The following error message is displayed if online changes are not permitted (DownloadEdit_ErrorsAsWarnings = No):

Fig. 7-10: Online Change Error

The following warning message is displayed if the online change warning message is not suppressed (DownloadEdit_SuppressWarningMsg = No) and DownloadEdit_ErrorsAsWarnings = Yes.

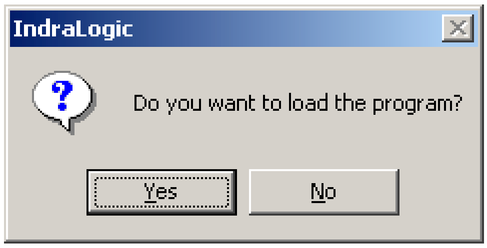


Fig. 7-11: Online Change Warning

Target Settings and Communication Parameters

IndraLogic target settings are located under the **Resource** tab and provide a means for selecting a predefined configuration for the selected target. IndraLogic supposes the following two VisualMotion targets:

- VisualMotion 11VRS (PPC-P)
- VisualMotion 11VRS (PPC-R)

Selecting one of the targets defines the following structures in IndraLogic:

- Hardware limitations (memory, Task quantities and types, etc.)
- IO options (local Reco, IO Image information, etc.)
- Data Exchange options with Motion: SysLibDirect support, IO Image contents (Binary and Process Data IO exchange, System Variables, etc.)
- Fieldbus options
- Libraries supported (defaults loaded)
- Networking abilities
- Communication options (SCP_VM, RS232, TCP/IP)
- PLC Browser commands
- External Tool support (e.g. GSD_Installer)
- Internal visualization support (Visualization tab)
- Compiler Error code definitions and text

Communication parameters defined the method of communication between IndraLogic and the control. Refer to IndraLogic Communication Channels on page 7-40 for details.

VisualMotion Variables

VisualMotion variables such as program variables, global variables and registers can be declared and passed to IndraLogic as SysLibDirect variables. Refer to SysLibDirect on page 7-17 for details.

Configuring Local Reco 02 I/O Modules in IndraLogic

Local Reco 02 I/O modules are only supported with the PPC-R22.1 control and configured using IndraLogic. Local Reco modules reside along side the control in a RMB02.2 rack. Before proceeding with the configuration, make a note of all the Local Reco I/O modules types that are installed. Refer to chapter 5 of the *VisualMotion 11 Project Planning* manual for details on Reco 02 hardware.

Note: Status for Local Reco 02 I/O is not visible unless the data is used in a PLC program.

Use the following steps, to configure Local Reco 02 I/O modules in IndraLogic:

1. From the *Resource* tab, double-click **PLC Configuration**.

Initially, the PLC Configuration displays one RMB02.2 rack with a PPC-R occupying the first two slots and two empty slots available for Reco 02 I/O modules.

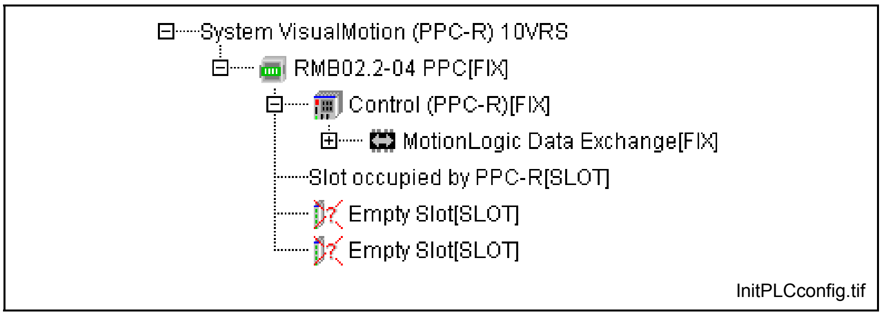


Fig. 7-12: Initial "System VisualMotion" PLC Configuration

2. Right-click on EMPTY Slot[SLOT], select ***Replace element*** and choose the I/O module installed in that slot.

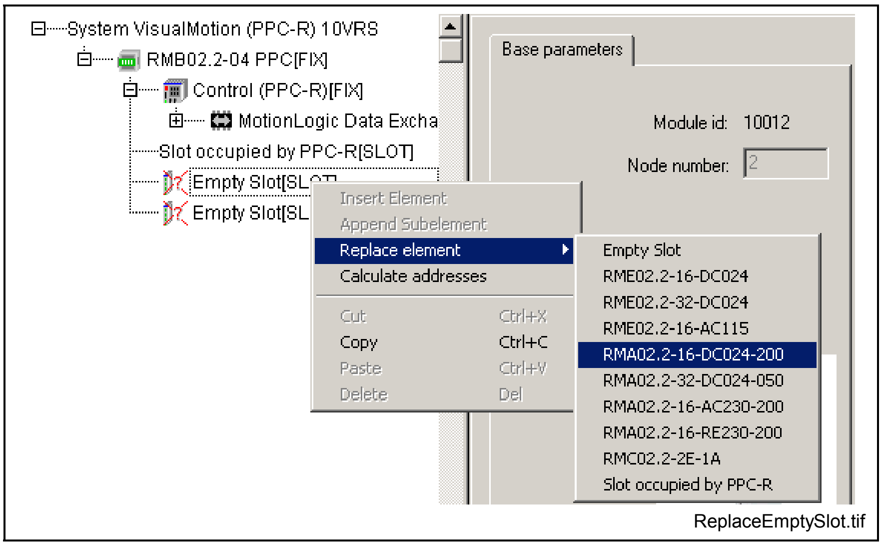


Fig. 7-13: Establish I/O in PLC Configuration

Note:	The I/O modules configured in IndraLogic must match the actual I/O modules in the system or a runtime error will be issued when downloading the PLC project.

Add an RMB02.2 Rack

To add an additional RMB02.2 rack, right-click on *System VisualMotion (PPC-R) 11VRS* and select **Append RMB02.2-04**.

An additional 3 RMB02.2 racks can be added and configured. For a total of 4 RMB02.2 racks in a system. Each additional rack can hold up to 4 Reco 02 I/O modules.

Applying Names to Reco 02 I/O Modules

A symbolic name can be assigned to an individual byte and/or bit of a Reco module defined in the PLC Configuration.

Use the following steps, to assign a name:

1. Expand the I/O module tree structure to view the byte and bits.
2. Open a text field by double-clicking on the word 'AT' before the IEC address. For example, the IEC address can appear as %QB0.
3. Enter a name that will be associated with the module's byte or bit memory location.

Note:	A comment field of a byte or bit can be modified within Base Parameter tab. Comments appear as "(* comment *)".

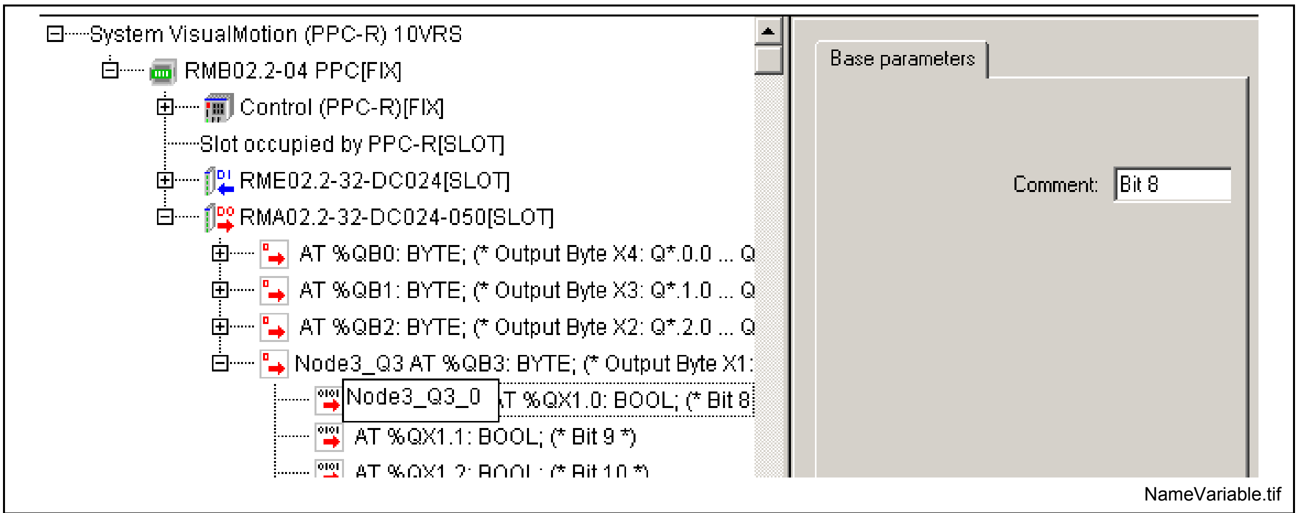
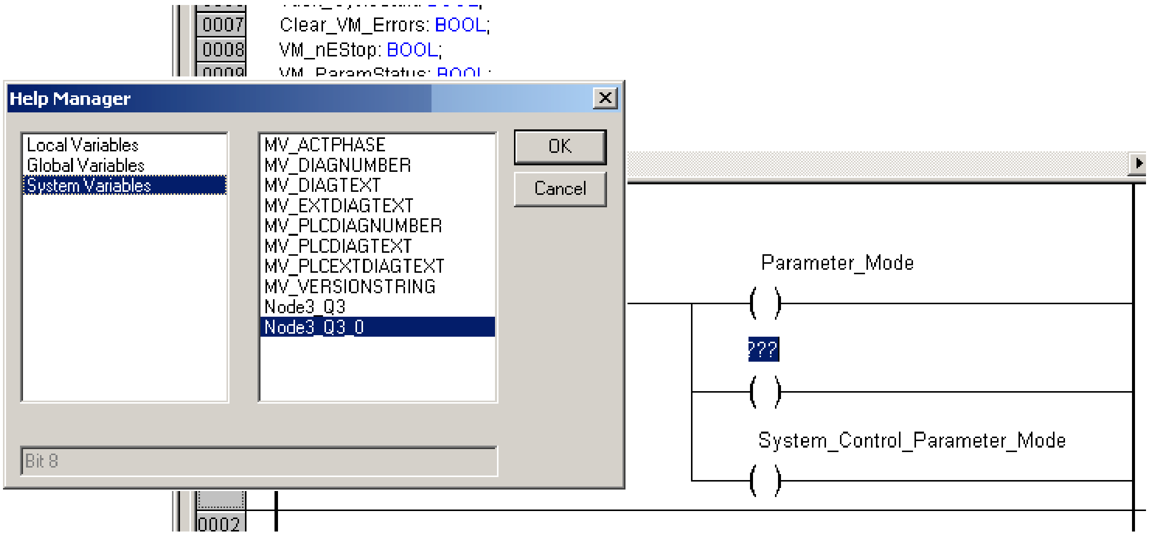


Fig. 7-14: Adding a Name to a Variable

For ease of programming, names assigned in the PLC Configuration can be selected as a *System Variable* from IndraLogic's Input Assistant. Double-click on the name location within the IndraLogic program and press the F2 key.



helpmanager.tif

Fig. 7-15: Selecting a System Variable in Input Assistant

The I/O module can be accessed using the symbolic name or IEC address directly.

Note: **It is highly recommended that any data used in a PLC project be addressed using symbolic names.** IEC memory addresses can be shifted down in memory whenever new data is appended to the Motion/Logic Data Exchange area or if a preceding module type changes. This can produce unexpected behaviors in an application.

Rexroth
Bosch Group

Motion Programming with IndraLogic

When IndraLogic is used for controlling motion, VisualMotion serves to initialize the axes for use by IndraLogic. The commissioning and parameterization of all axes in the system can be performed with *DriveTop* and the *Parameter Overview* tool in VisualMotion Toolkit. Motion commands are implemented through function blocks that are called from the IndraLogic tasks. Motion function block libraries such as MV_Motion.lib and MV_UserMotion.lib are provided for single axis and/or ELS motion. The RMC_PLCopen.lib library also provides standardized function blocks for single axis motion. Refer to IEC Libraries on page 7-43 for a complete list of supported libraries.

With IndraLogic based motion, the error reaction for each axis must be programmed by the user in IndraLogic. Function blocks are provided to enable, disable, and halt axes and other motion components. The programmer must be aware of the Sercos phase and enabled status of an axis before a motion command is issued.

Note: In contrast, when VisualMotion is used to control the motion, IndraLogic serves as a replacement for the I/O Mapper and/or the external PLC. The axes are assigned to user tasks and their error reactions are handled by VisualMotion. Refer to chapter 11 for error reaction descriptions and settings.

Axis Configuration

The assignment of axes in a system is dependent on whether VisualMotion or IndraLogic will control the motion. In either case, all axes must be declared in VisualMotion Toolkit.

Axes are assigned to a VisualMotion task or to IndraLogic within the *Setup Axes List* in the *Project Navigator* window.

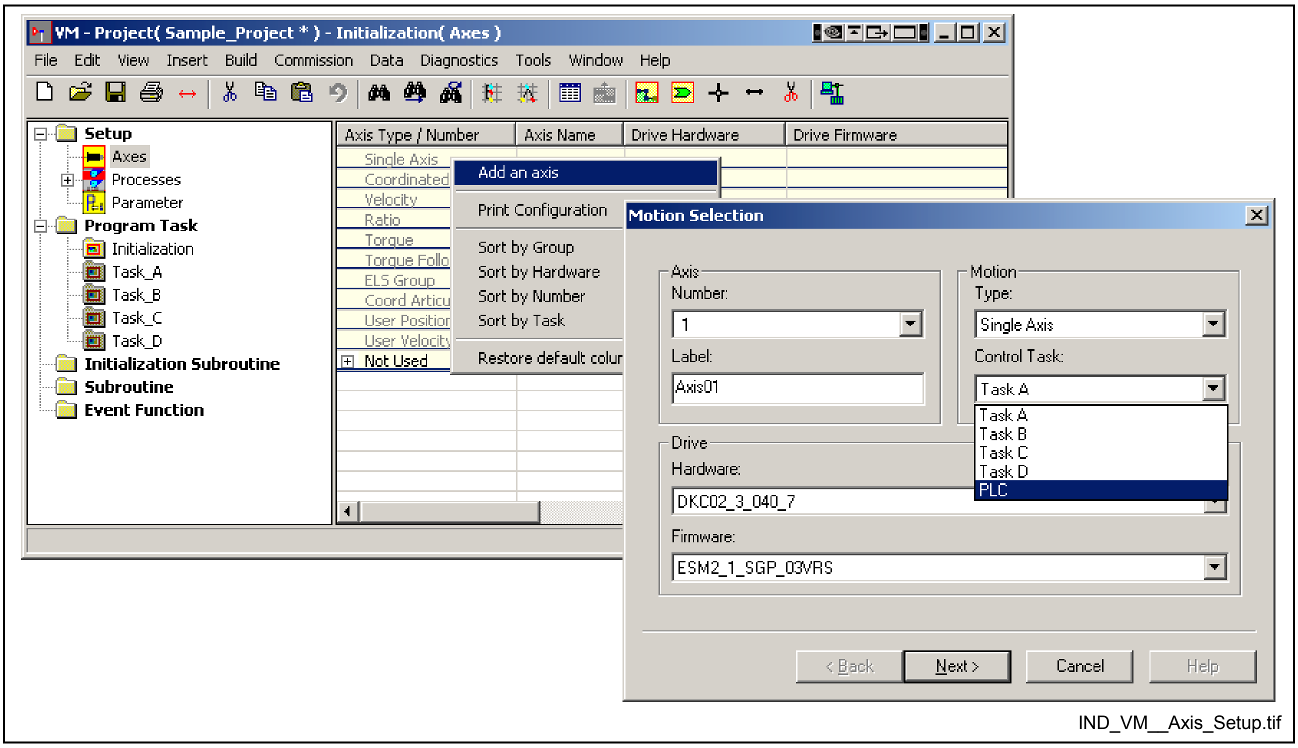


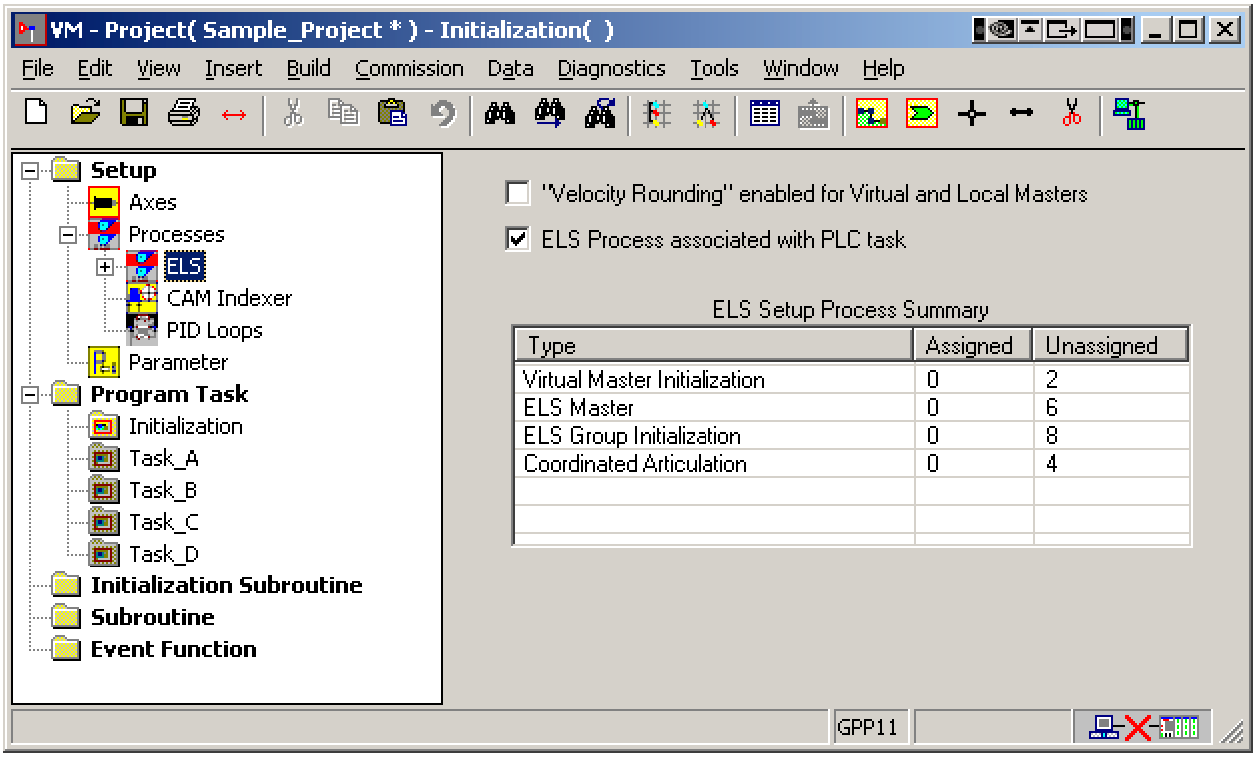
Fig. 7-16: Axis Assignment for IndraLogic

Note: For details on how the setup axes and ELS processes, refer to section 4.3 under headings *Setup Axes* and *Setup Processes*.

Associating the ELS System

When ELS motion is controlled by VisualMotion, the ELS system is associated with task A. The ELS system stops when task A stops.

When ELS motion is controlled by IndraLogic, the association to the PLC must be set within the Setup Processes ELS settings. The following figure shows a check next to the "ELS Process associated with PLC task" option. This option transfers the association of the ELS system from task A to IndraLogic.



AssociateELS.tif

Fig. 7-17: ELS Associated to PLC

Motion and Logic Data Exchange

The methods for exchanging data between motion and logic include:

Access Method	Description
SysLibDirect	data is declared as a global variable and is updated in real-time within the IndraLogic task
I/O Image	data is updated at the beginning and end of an IndraLogic task
Function Blocks	data is accessed through command functions organized in pre-configured groups to perform a potentially complex task
Shared Memory Area	pre-defined data is shared between the motion and logic areas of firmware

Table 7-4: Motion Data Access Methods

The following figure illustrates the type of motion data that can be accessed by the different IndraLogic access methods.

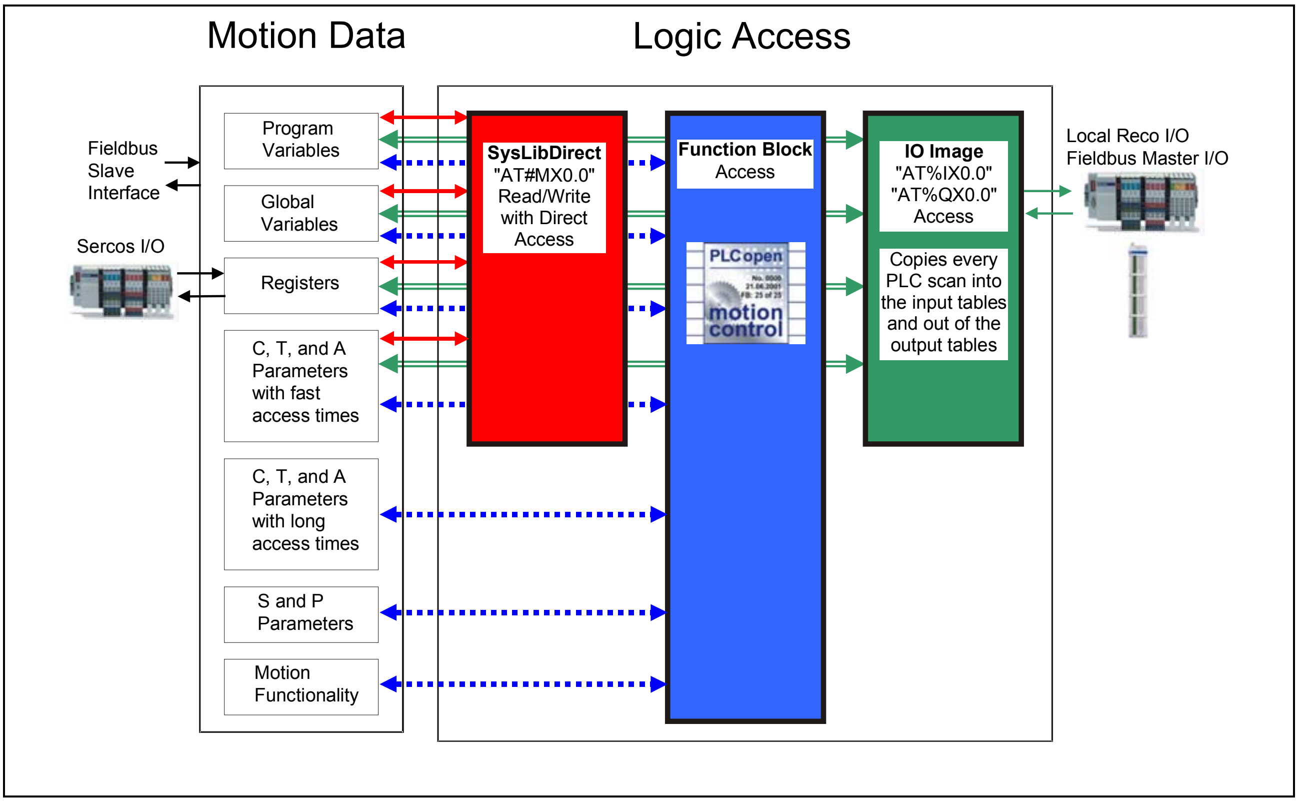


Fig. 7-18: Motion/Logic Data Access

Motion Data	PLC Access: SysLibDirect (read/write with direct access)	PLC Access: I/O Image	PLC Access: Function Block Access
Program Variables	Yes	Yes	Yes
Global Variables	Yes	Yes	Yes
Registers	Yes	Yes	Yes
Control, Task, and Axis Parameters with fast access times	Yes	Yes	Yes

Motion Data	PLC Access		
	SysLibDirect (read/write with direct access)	**I/O Image**	**Function Block Access**
Control, Task, and Axis Parameters with long access times (i.e., list parameters)	No	No	Yes
S and P Parameters	No	No	Yes
Motion Functionality	No	No	Yes

Table 7-5: Summary of Motion Data Access Methods

SysLibDirect (Direct Access)

SysLibDirect is an access method where a special type of address is associated to a variable in IndraLogic. This access method provides IndraLogic with access to VisualMotion registers, register bits, global variables and program variables. SysLibDirect variables must be selected in VisualMotion Toolkit before launching IndraLogic. Refer to Configuration in VisualMotion on page 7-18 for details.

The following table describes the different supported VisualMotion data types:

VM Data Type	IEC Data Type	Data Size	High Word	Low Word	SysLibDirect Address Example
Register Bit	BOOL	X - Bit	Register Number (1 to 1024)	Bit Number (0 to 15)	#MX1.2:BOOL
Register	WORD	W - Word	2000 (Global Constant)	Register Number	#MW2000.1:WORD
Program Integer	DINT	D – Double Word	2001 (Global Constant)	Integer Number	#MD2001.1:DINT
Program Float	REAL	D – Double Word	2002 (Global Constant)	Float Number	#MD2002.1:REAL
Global Integer	DINT	D – Double Word	2003 (Global Constant)	Global Integer Number	#MD2003.1:DINT
Global Float	REAL	D – Double Word	2004 (Global Constant)	Global Float Number	#MD2004.1:REAL
C-Parameter **	varies	varies	3000 (Global Constant)	Parameter Number	#MD3000.125:DWORD
A-Parameter **	varies	varies	3101…3199 (Axis # 1-99)	Parameter Number	#MD3101.100:REAL
T-Parameter **	varies	varies	3201…3204 (Task # 1-4)	Parameter Number	#MD3202.24:REAL
** The allowable control, axis and task parameters are listed in control parameters C-0-1638 and C-0-1639.					

Table 7-6: SysLibDirect Data Types

Note: The access time for SysLibDirect variables is longer than accessing local variables in a PLC program.

SysLibDirect Variable Usage and Status

SysLibDirect variables are declared as global variables in the PLC and accessed from the motion side at the immediate point of execution. This means that when a POU accesses a SysLibDirect variable, that data is read from and written to VisualMotion at that instant. For this reason, data consistency can not be guaranteed if the same variable is being accessed multiple times by the same POU or from a different POU. If data consistency is required, copy the contents of the SysLibDirect variable to a new local or global variable and then use the new variable in the POU or use the I/O Image.

Note: Excessive use of SysLibDirect variables can increase the system load of PLC tasks.

Note: VisualMotion variables used in an IndraLogic POU are not automatically updated if the variable label changes in the VisualMotion project. They will have to be referenced again within IndraLogic POUs.

Once used in a PLC program, the user can monitor the status of SysLibDirect variables just like local or global IndraLogic variables in IndraLogic version 1.4 or greater.

Note: HMI devices can access SysLibDirect variables when using IndraLogic version 1.4 or greater.

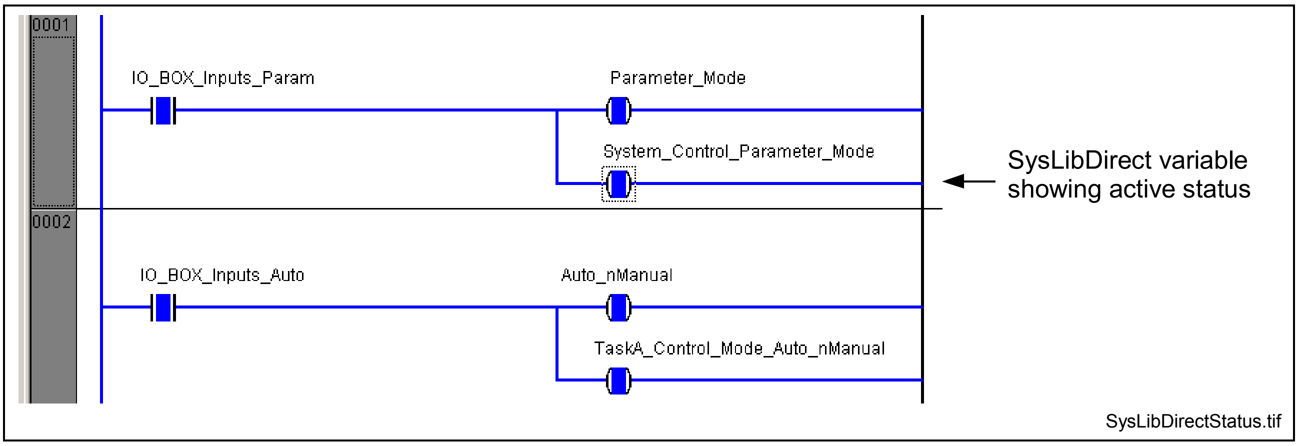


Fig. 7-19: SysLibDirect Variable Status

Configuration in VisualMotion

VisualMotion variables can be shared with IndraLogic using predefined default label names assigned by VisualMotion or by prepending an ID to the front of each default label name. Refer to Prepend Data Type to Label on page 7-21 for details.

The supported addressing format are as follows:

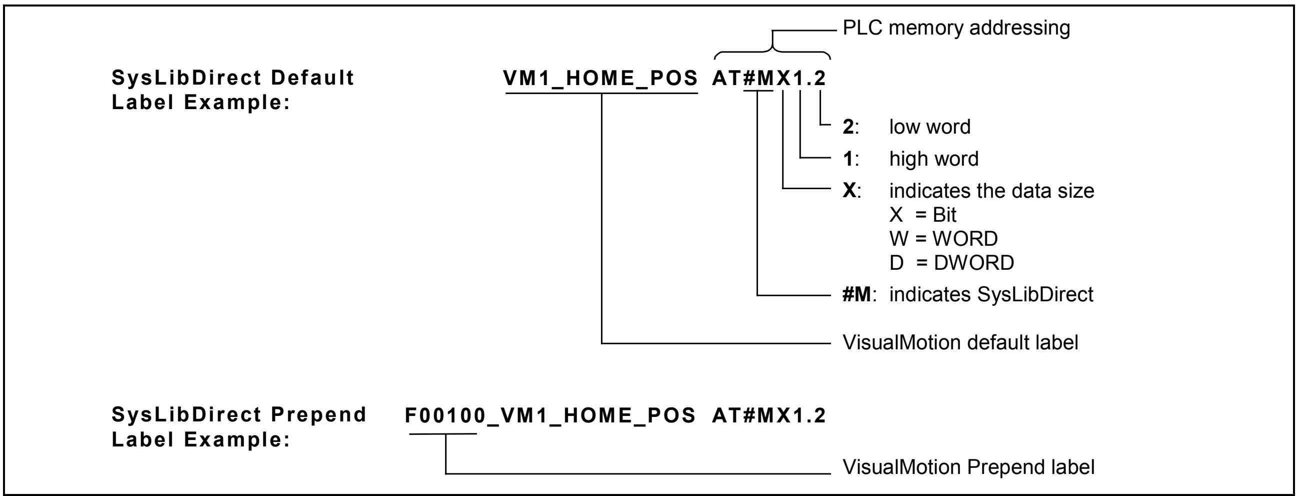


Fig. 7-20: SysLibDirect Access Format Examples

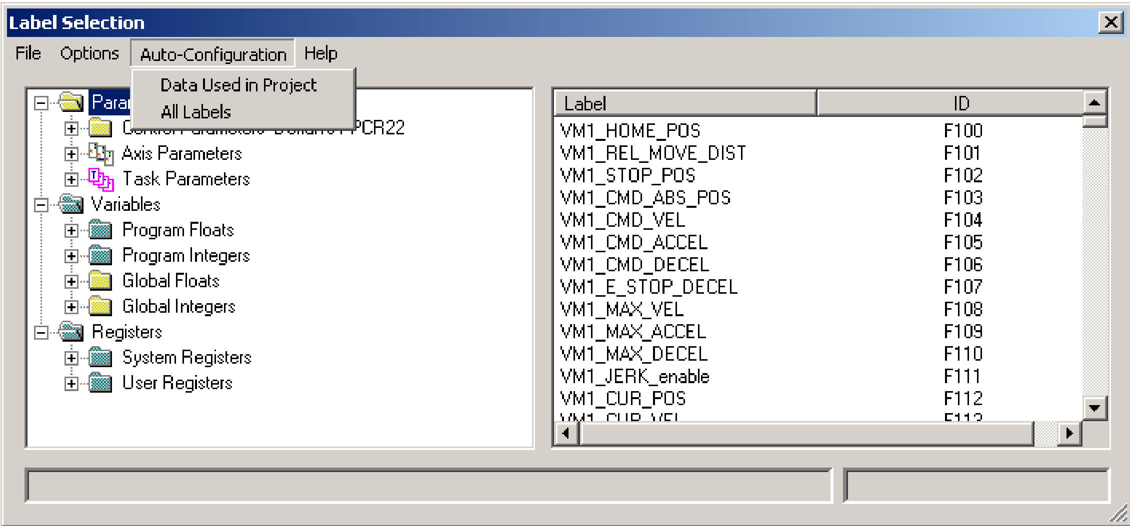
Note: It is highly recommended to use only one addressing format of the VisualMotion variable labels when sharing with IndraLogic.

The following steps describe how to share default VisualMotion labels with IndraLogic:

1. In VisualMotion Toolkit, select ***Tools*** ⇒ ***Options…*** and select the *IndraLogic* tab.
2. Click the **Select Item To Share With IndraLogic** button.
3. Expand the Variables and Registers folders.
4. Further expand the relevant subfolders and place a check next to the variable to be shared with IndraLogic.

Note: A second method would be to have VisualMotion automatically select all the variable and register data used in the project by selecting ***Auto-Configuration*** ⇒ ***Data Used in Project***.

5. Close the *Label Selection* window and launch IndraLogic.

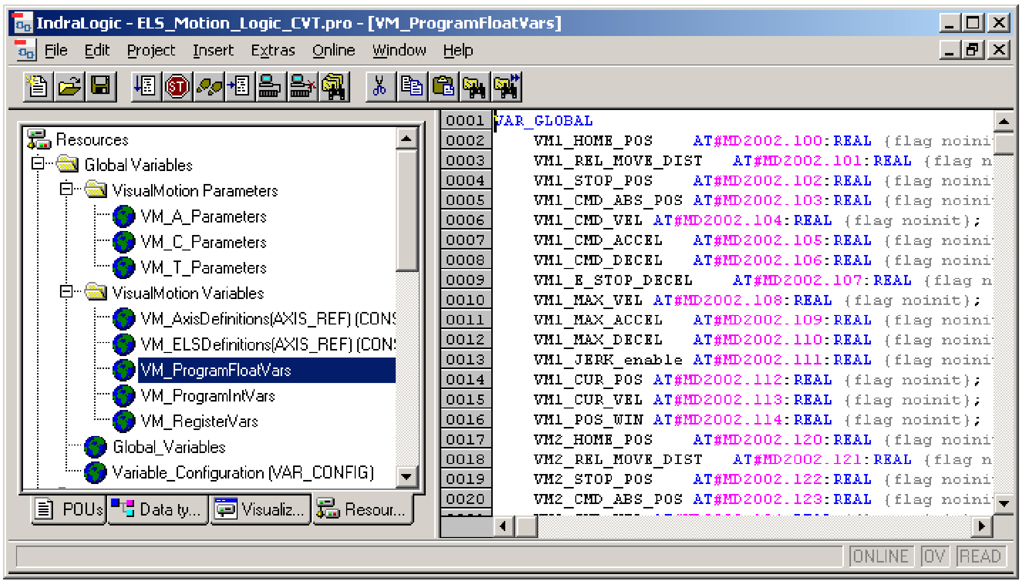


AutoConfigSys.tif

Fig. 7-21: Label Selection

Note: Shared variable labels are not dynamically updated and require that IndraLogic be closed and launched again if any new labels are selected.

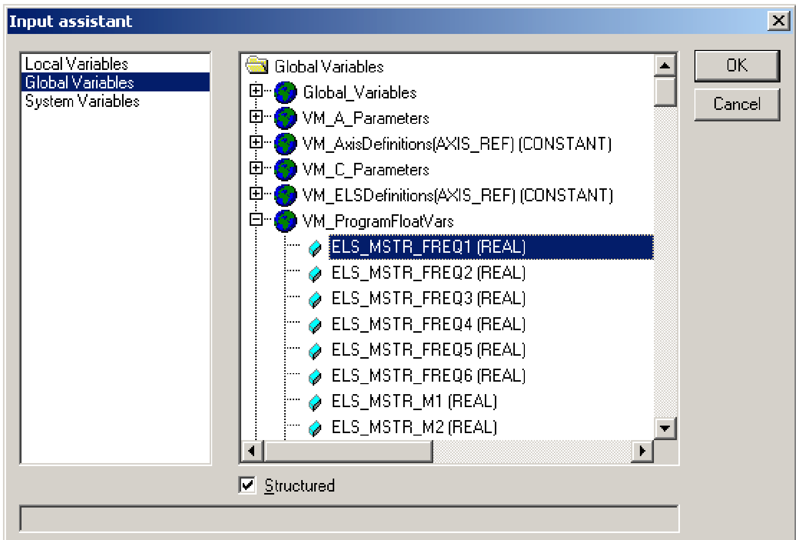
In IndraLogic, the selected labels are displayed as SysLibDirect variables in the **Resource** tab under Global Variables as VisualMotion Parameters (A,C,and T) and VisualMotion Variables.



DeclaredVMvariables.tif

Fig. 7-22: VisualMotion Data Types in IndraLogic

SysLibDirect variables can be accessed in a PLC program by selecting the location where the variable will be used, pressing the **F2** key (Input Assistant), and locating the desired variable under **Global Variables**.



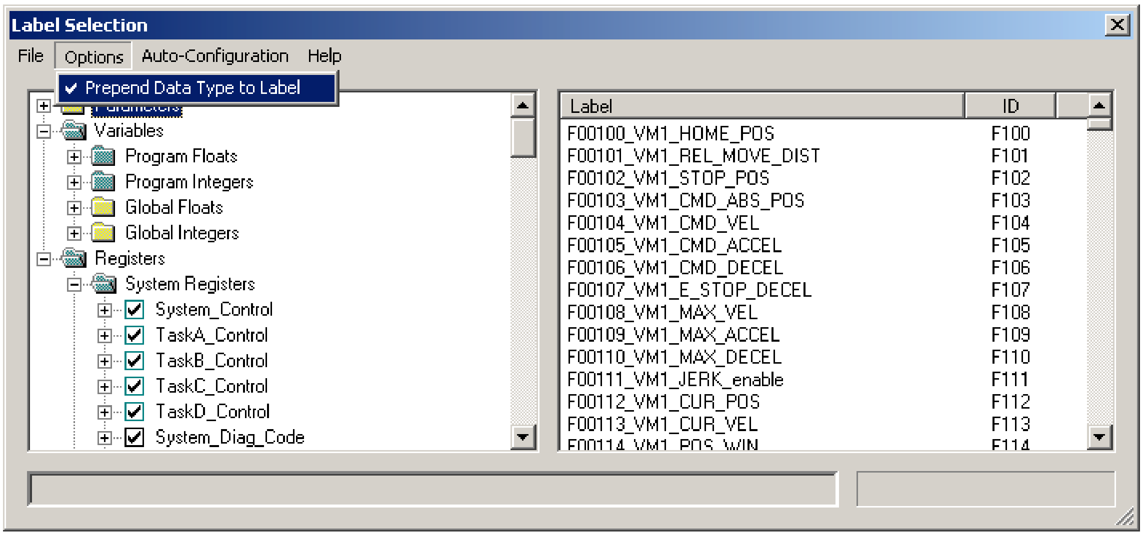
SysLibDirectGlobalVar.tif

Fig. 7-23: SysLibDirect Global Variables

Note: To view all global variables as one large list in alphabetical order, uncheck the **Structured** checkbox at the bottom of the Input Assistant window.

Prepend Data Type to Label

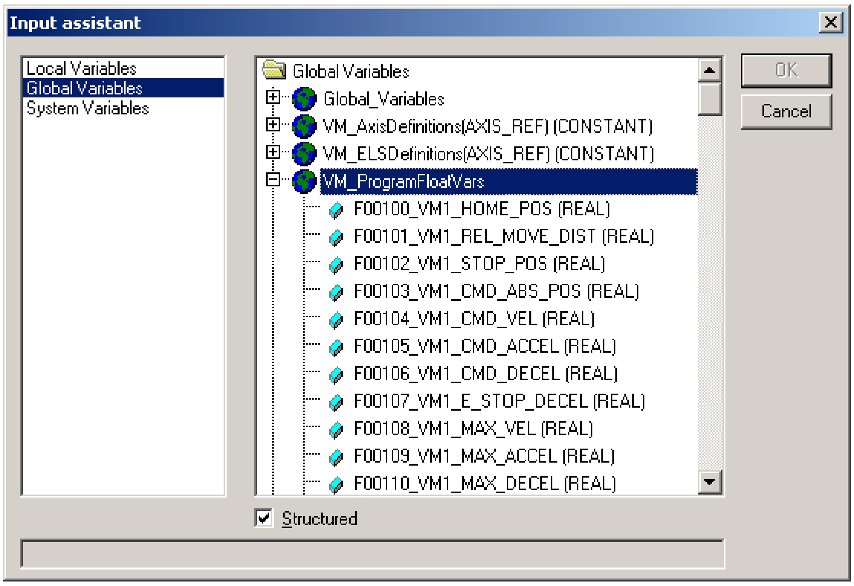
The data type ID (e.g. F100) of a variable label in VisualMotion can be added as a prefix to the label name before sharing it with IndraLogic. This is performed by selecting **Prepend Data Type to Label** under the *Options* menu. This option facilitates the identification of the different variable labels once they appear as SysLibDirect variables in IndraLogic.



PrependVMlabels.tif

Fig. 7-24: Prepend Data Type to Label

The following figure displays prepended label names as SysLibDirect variables in IndraLogic.



SysLibDirectGlobalVarPrepend.tif

Fig. 7-25: Prepended SysLibDirect Global Variables

MotionLogic Data Exchange in I/O Image

The MotionLogic Data Exchange uses input and output memory areas for reading and writing data. The data is updated at the beginning and end of every PLC cycle.

This memory area can be viewed in IndraLogic by selecting ***MotionLogic Data Exchange*** in the PLC Configuration. Data is subdivided into Cyclic Binary I/O and Cyclic Process Data types.

Cyclic Binary I/O

The cyclic binary I/O memory area contains the VisualMotion system, axes, Virtual Masters, ELS Masters, ELS Groups, Link Ring, and I/O registers.

Note: Cyclic binary I/O is structured into control and status register memory areas for the VisualMotion System and the ELS System. For the IO Registers, an input and output structure is used. These control and status register memory areas are predefined to access the correct VisualMotion registers.

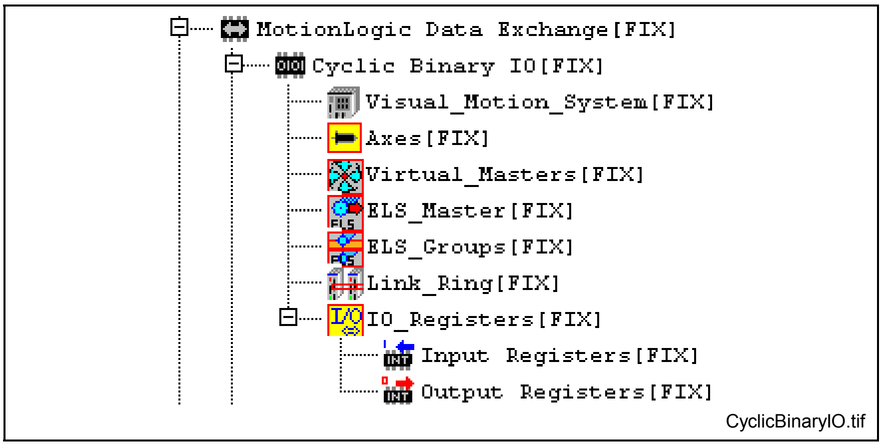


Fig. 7-26: Cyclic Binary I/O

Cyclic Process Data

The cyclic process data memory area contains VisualMotion program variables, global variables, card (control) parameters, axis parameters, and task parameters. All of the data in this memory area is of non-Boolean type.

Note: The declaration of process data is required in both an Input and Output memory area for reading and writing the same VisualMotion data from the I/O Image. Refer to Declaring Cyclic Process Data on page 7-24.

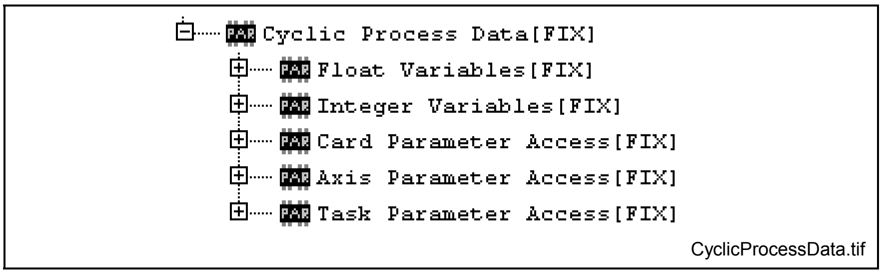


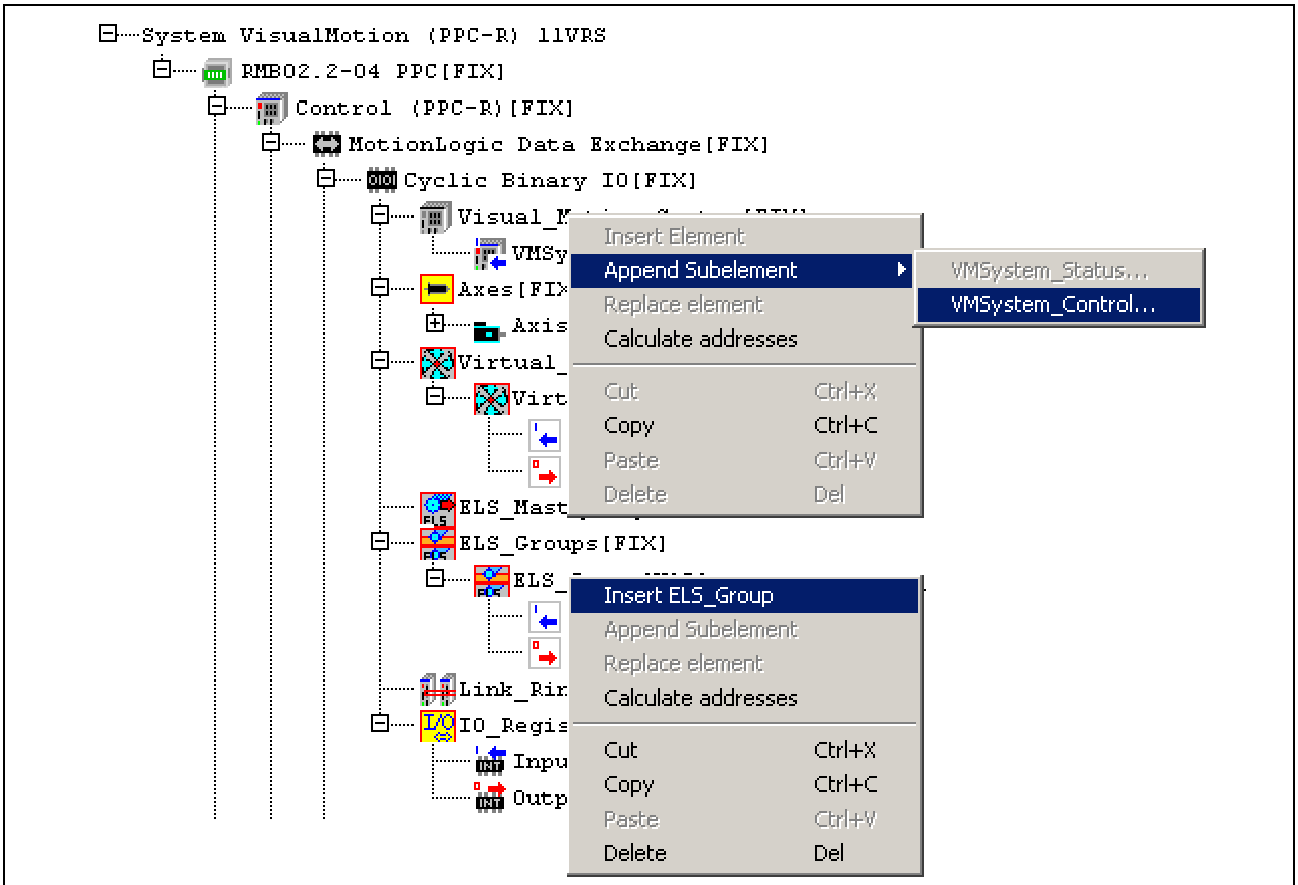
Fig. 7-27: Cyclic Process Data

Declaring the MotionLogic Data Exchange

The following section describes the basic declaration of data in the Cyclic Binary I/O and Cyclic Process Data memory locations.

Declaring Cyclic Binary I/O

For cyclic binary I/O, right clicking over one of the available data types allows the user to append subelements and specify data types such as Axes or Virtual_Masters.



AppendCyclicBinaryIO.tif

Fig. 7-28: Appending Cyclic Binary Elements

The following table lists the configurable variables in the Cyclic Binary IO:

Variable	Data Type	I/O Access	Description
Visual_Motion_System	STRUCT	Read/Write	System control and status
Axes	STRUCT	Read/Write	Axis control and status, up to 64 axes
Virtual_Master	STRUCT	Read/Write	Virtual Master control and status, up to 2 Virtual Masters
ELS_Master	STRUCT	Read/Write	ELS Master control and status for current master
ELS_Groups	STRUCT	Read/Write	ELS Group control and status, up to 8 ELS Groups
Link_Ring	STRUCT	Read only	Link Ring status and signal validity for up to 32 nodes
IO_Registers	WORD	Read/Write	VisualMotion registers, up to 1024 registers Each register expands to 16 bits of type BOOL

Table 7-7: Configurable Cyclic Binary I/O

Rexroth Bosch Group

Declaring Cyclic Process Data

When declaring cyclic process data, e.g., a float variable, two declarations are required for reading and writing data. The input declaration is used to read the data from VisualMotion at the beginning of the PLC cycle and the output declaration is used to write the data back to VisualMotion at the end of the PLC cycle. The same VisualMotion data is accessed, however, the I/O Image requires two separate memory locations for reading and writing data.

For example, to declare a global float as input memory:

1. Expand the *MotionLogic Data Exchange [FIX]* tree structure to display *Cyclic Process Data [FIX].*
2. Expand the *Global Floats* tree structure and right click over *Inputs Global Floats*.
3. Select *Append Input Single Global Float*.
4. In the *Module Parameters* tab to the right of the I/O Image, set the *Value* field to the number assigned to the VisualMotion global float that will be accessed.
5. Left click over "AT %ID16:", for example, and assign a label that will be used by the PLC program to access the specific input memory location.

Note: No two labels can have the same exact name in the I/O Image. When assigning labels to access (read/write) the same VisualMotion data, select names that will be recognized as accessing the same data between the input and output memory locations. Refer to the labels "GF100_IN" and "GF100_OUT" in the following figure.

To declare an output, repeat steps 2 – 5 for *Outputs Global Floats*.

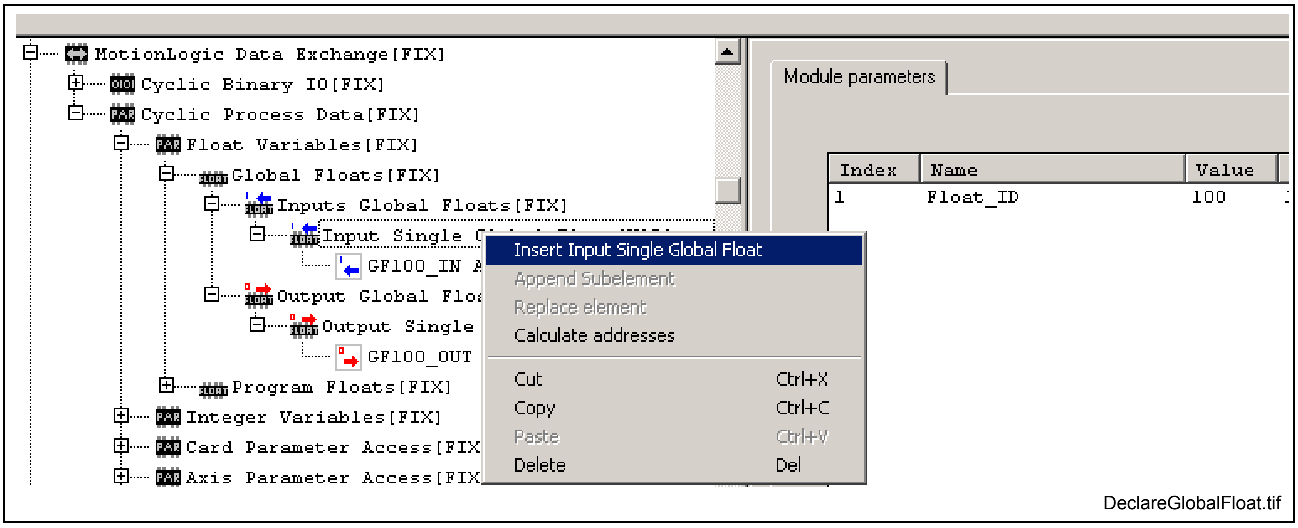


Fig. 7-29: Declare Global Float in Cyclic Process Data

The process for declaring parameters (Card, Axis, and Task) is similar to that of global floats. The difference is observed when the user right clicks over a parameter access type, selects *Append Subelements* and then needs to select a parameter based on the data type (e.g. REAL, WORD, INT, etc.). Once a data type is selected the *Module Parameters* tab displays a dropdown list of the available parameters for the selected data type along with the Axis_ID to specify the axis number.

For a complete listing of available parameters based on PLC data types, refer to Cyclic Process Data Types on page 7-32.

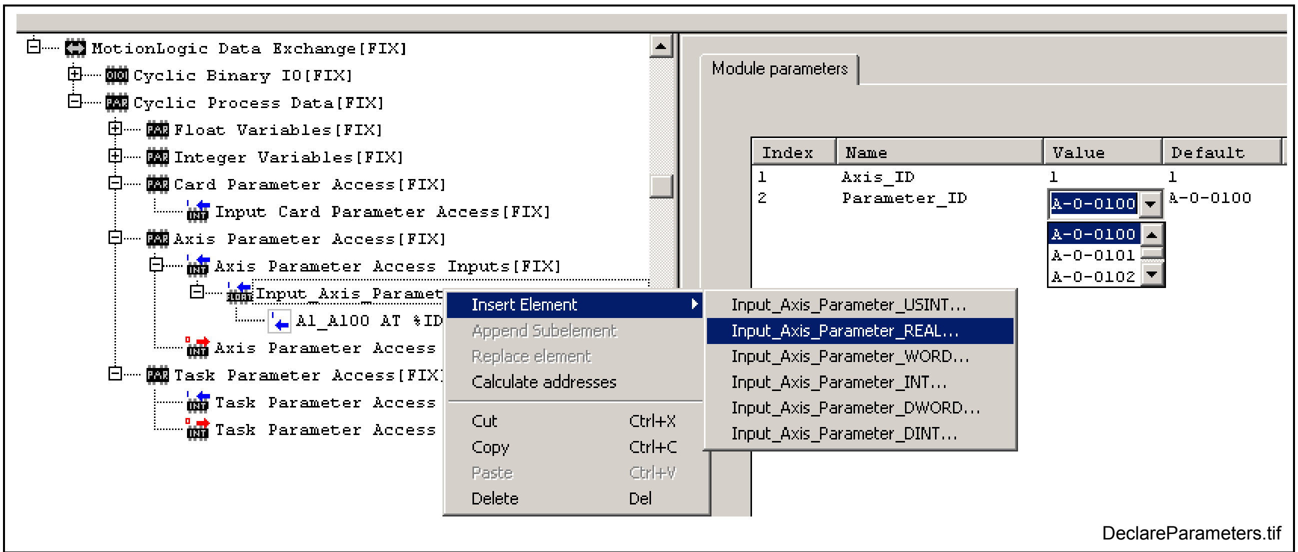


Fig. 7-30: Declare Parameters in Cyclic Process Data

Accessing the MotionLogic Data Exchange

Data in the MotionLogic Data Exchange memory area of the PLC Configuration can be accessed from a POU by highlighting the location where the data will be placed and pressing the F2 key to launch the *Input Assistant.* Select the relevant data type under Global Variables or System Variables.

Note: Only declared data labeled with a valid name will appear in the Input Assistant under Global Variables or System Variables.

Specific elements from a Cyclic Binary I/O structure can be accessed by using simple dot notation. Variables and parameters from the Cyclic Process Data area can be accessed from the Input Assistant, if labeled.

Accessing Cyclic Binary I/O

The elements of a configured variable of type STRUCT in the Cyclic Binary I/O Image can be accessed in a POU object by using the Input Assistant followed by standard dot notation.

Note: Online monitoring of I/O data is not possible unless the data is used in a PLC program.

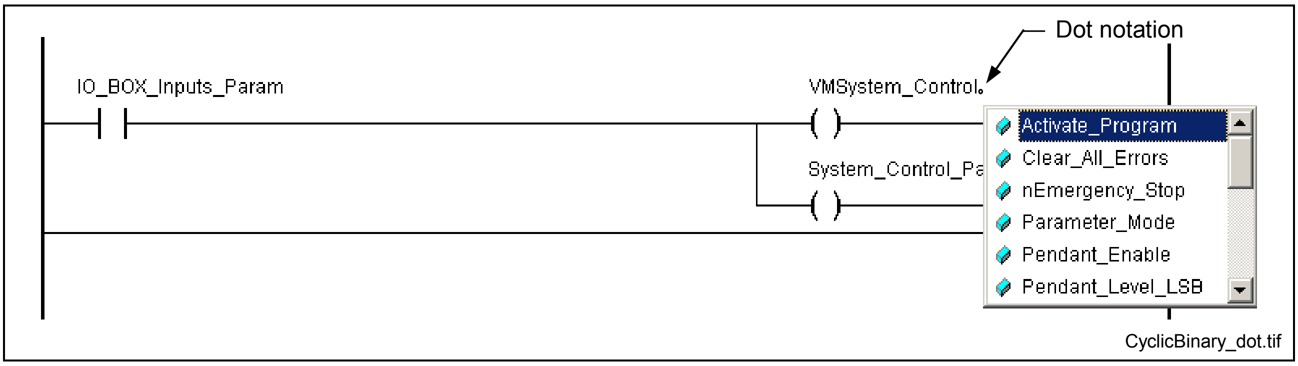


Fig. 7-31: Accessing Cyclic Binary I/O using Dot Notation

Note: The name of any subelement can be enter manually. To ensure that the correct subelement is used in a POU, use IndraLogic's Input Assistant (F2 key) to search for *System Variables* or enter the name of the subelement as assigned in the Cyclic Binary I/O.

The following tables list the available STRUCT elements for each variable type in the Cyclic Binary I/O:

Accessing Cyclic Process Data

Variables and parameters (Card, Axis, and Task) in the Cyclic Process Data area can be selected for use in a POU by using the Input Assistant (F2 key) and selecting the relevant data from Global Variables or System Variables.

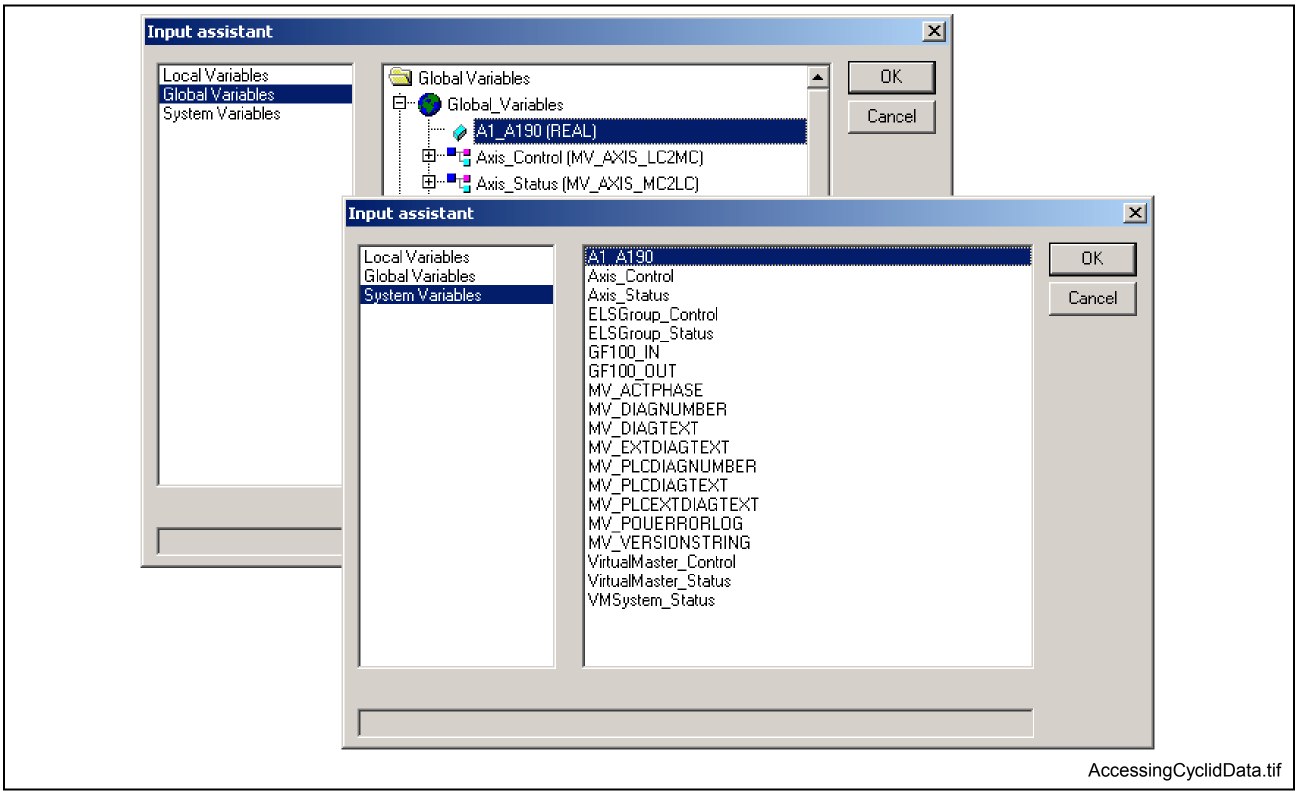


Fig. 7-32: Accessing Cyclic Process Data

Cyclic Binary I/O Data Structures

The following tables list the different cyclic binary I/O data structures supported in the I/O Image.

Visual_Motion_System Variable

This variable is used for the VisualMotion System control and status words. They're provided as two subelements of type STRUCT, one for the control word (MV_SYSTEM_LC2MC) and one for status word (MV_SYSTEM_MC2LC).

Note: These structure data types are part of the MV_CyclicIO.lib target library.

Subelement Name	STRUCT Element	Equivalent VM Register Bit
VMSystem_Control	Parameter_Mode	1
(Similar to Register 001)	nEmergency_Stop	3
	Clear_All_Erorrs	5
	Pendant_Live_Man	6
	Rebuild_Double_Ring	7
	Activate_Program	8
	Program_Select_LSB	9
	Program_Select_Bit_2	10
	Program_Select_Bit_3	11
	Program_Select_MSB	12
	Pendant_Enable	14
	Pendant_Level_LSB	15
	Pendant_Level_MSB	16

Table 7-8: Visual_Motion_System Control Subelement (Output)

Subelement Name	STRUCT Element	Equivalent VM Register Bit
VMSystem_Status	Parameter_Mode	1
(Similar to Register 021)	Service_Channel_Ready	4
	Error	5
	Error_Active	6
	Warning_Active	7
	PLC_Error_Active	8
	Active_Program_LSB	9
	Active_Program_Bit_2	10
	Active_Program_Bit_3	11
	Active_Program_MSB	12
	TP_Password_Active	13
	Teach_Pendent	14
	Flash_Back_Vaoid	15
	AutoStore_Copy_Valid	16

Table 7-9: Visual_Motion_System Status Subelement (Input)

Axis Variable

This variable is used to configure up to 64 instances of system axes. Each axes instance includes the structure for the axis control register (MV_AXIS_LC2MC) and a structure for the axis status register (MV_AXIS_MC2LC). The ID number of the axis corresponds to its Sercos address.

Note: These structure data types are part of the MV_CyclicIO.lib target library.

Subelement Name	STRUCT Element	Equivalent VM Register Bit
Axis_Control	Disable_Axis	1
(Similar to Register 11-18,	Jog_Forward	2
209-240 and 441-464)	Jog_Reverse	3
	Synchronized_Jog	4

Table 7-10: Axis Control Subelement (Output)

Subelement Name	STRUCT Element	Equivalent VM Register Bit
Axis_Status	Multiplex_Ch_Enabled	1
(Similar to Register 31-38,	Jogging_Forward	2
309-340 and 541-564)	Jogging_Reverse	3
	Phase_Adjusted	4
	ELS_Enabled	5
	ELS_Secondary_Mode	6
	Axis_in_Position	7
	Axis_Aligned	8
	Axis_Stopped	10
	Axis_Halted	11
	Class_3_Status	12
	Class_2_Warning	13
	Shutdown_Error	14
	Drive_Ready_LSB	15
	Drive_Ready_MSB	16

Table 7-11: Axis Status Subelement (Input)

Virtual_Masters Variable

This variable is used to configure up to 2 instances of Virtual Masters. Each instance includes a structure for the control register (MV_VIRT_MASTER_LC2MC) and a structure for the status (MV_VIRT_MASTER_MC2LC). The ID number of the Virtual Master corresponds to the Virtual Master number. Refer to section 6.1, *Electronic Line Shafting*, for descriptions of element names.

Note: These structure data types are part of the MV_CyclicIO.lib target library.

Subelement Name	STRUCT Element	Equivalent VM Register Bit
VirtualMaster_Control	VM_CT_FSTOP	1
(Similar to Default	VM_CT_HOME	2
Reg. 150 for Virtual	VM_CT_GO	3
Master 1 and Reg. 151	VM_CT_VMODE	4
for Virtual Master 2)	VM_CT_RELMODE	5
	VM_CT_RELTRIG	6

Table 7-12: Virtual Master Control Subelement (Output)

Subelement Name	STRUCT Element	Equivalent VM Register Bit
VirtualMaster_Status	VM_ST_FSTOP	1
(Similar to Default	VM_ST_HOME	2
Reg. 241 for Virtual	VM_ST_VMODE	4
Master 1 and Reg. 242	VM_ST_RELMODE	5
for Virtual Master 2)	VM_ST_ZEROVEL	7
	VM_ST_INPOS	8

Table 7-13: Virtual Master Status Subelement (Input)

ELS_Master Variable

This variable is used to configure a single instance of the current ELS Master. This instance includes a structure for the control register (MV_ELS_MASTER_LC2MC) and a structure for the status register (MV_ELS_MASTER_MC2LC). Refer to section 6.1, *Electronic Line Shafting*, for descriptions of element names.

Subelement Name	STRUCT Element	Equivalent VM Register Bit
ELSMaster_Control	ELS_M_CT_SET_REF1	7
(Similar to Default	ELS_M_CT_SET_REF2	8
Reg. 140 for ELS	ELS_M_CT_SET_REF3	9
Master Control)	ELS_M_CT_SET_REF4	10
	ELS_M_CT_SET_REF5	11
	ELS_M_CT_SET_REF6	12
	ELS_M_CT_SLIP_CAPT1	15
	ELS_M_CT_SLIP_EN	16

Table 7-14: ELS Master Control Subelement (Output)

Subelement Name	STRUCT Element	Equivalent VM Register Bit
ELSMaster_Status	ELS_M_ST_STOPPED1	1
(Similar to Default	ELS_M_ST_STOPPED2	2
Reg. 141 for ELS	ELS_M_ST_STOPPED3	3
Master Status)	ELS_M_ST_STOPPED4	4
	ELS_M_ST_STOPPED5	5
	ELS_M_ST_STOPPED6	6
	ELS_M_ST_REF1	7
	ELS_M_ST_REF2	8
	ELS_M_ST_REF3	9
	ELS_M_ST_REF4	10
	ELS_M_ST_REF5	11
	ELS_M_ST_REF6	12
	ELS_M_ST_SLIP_ERROR	14
	ELS_M_ST_SLIP_ENC	15
	ELS_M_ST_SLIP_ENA	16

Table 7-15: ELS Master Status Subelement (Input)

ELS_Groups Variable

This variable is used to configure up to 8 instances of ELS Groups. Each instance includes a structure for the control register (MV_ELS_GROUP_LC2MC) and a structure for the status register (MV_ELS_GROUP_MC2LC). The ELS Group value in *Module parameters* tab corresponds to the ELS Group's number. Refer to section 6.1, *Electronic Line Shafting*, for descriptions of element names.

Subelement Name	STRUCT Element	Equivalent VM Register Bit
ELSGroup_Control	G_CT_LOCK_OFF	1
(Similar to Default	G_CT_M_REL_PH	2
Reg. 152-159 for ELS	G_CT_S_REL_PH	3
Groups 1-8 Control)	G_CT_MSTR_SEL	4
	G_CT_VAR_CLK	5
	G_CT_LOCAL	6
	G_CT_LM_FSTOP	7
	C_CT_LM_HOME	8
	G_CT_LM_GO	9
	G_CT_LM_VMODE	10
	G_CT_M_ABS_PH	11
	G_CT_S_ABS_PH	12
	G_CT_LM_RELMODE	13
	G_CT_LM_RELTRIG	14
	G_CT_MSTR_FOL_PH	15
	G_CT_MSTR_ABORT_PH	16

Table 7-16: ELS Group Control Subelement (Output)

Subelement Name	STRUCT Element	Equivalent VM Register Bit
ELSGroup_Status	G_ST_LOCK_ON	1
(Similar to Default	G_ST_M_REL_PH	2
Reg. 243-250 for ELS	G_ST_S_REL_PH	3
Groups 1-8 Status)	G_ST_MSTR_SEL	4
	G_ST_VAR_ACK	5
	G_ST_LOCAL	6
	G_ST_LM_FSTOP	7
	G_ST_LM_HOME	8
	G_ST_MOTION	9
	G_ST_LM_VMODE	10
	G_ST_M_ABS_PH	11
	G_ST_S_ABS_PH	12
	G_ST_LM_RELMODE	13
	G_ST_LM_ZEROVEL	14
	G_ST_LM_INPOS	15

Table 7-17: ELS Group Status Subelement (Input)

Link_Ring Variable

This variable is used to configure a single instance of the Link Ring cross communication (MV_LINK_RING_MC2LC). This instance includes a structure for the input used for monitoring the cross communication. Refer to section 6.1, *Electronic Line Shafting*, for descriptions of element names.

Subelement Name	STRUCT Element	Equivalent VM Register Bit
Link_Ring_Status	LINK_ERROR	Reg. 40, Bit 4
(Similar to Default	ERROR_PRIM_OPTIC_RING	Reg. 40, Bit 5
Reg. 40-42 for Link	ERROR_SEC_OPTIC_RING	Reg. 40, Bit 6
Status and Data 1)	REDUNDACY_LOSS	Reg. 40, Bit 7
	NODE_1_DATA_VALID through NODE_16_DATA_VALID	Reg. 41, Bits 1-16
	NODE_17_DATA_VALID through NODE_32_DATA_VALID	Reg. 42, Bits 1-16

Table 7-18: Link Ring Status Subelement

IO_Registers Variable

The IO-Registers variable can be configured to access up to 1024 VisualMotion registers. Each register is of type WORD and can be expanded to display 16 bits of type BOOL. Symbolic names can be assigned at the WORD and/or bit level.

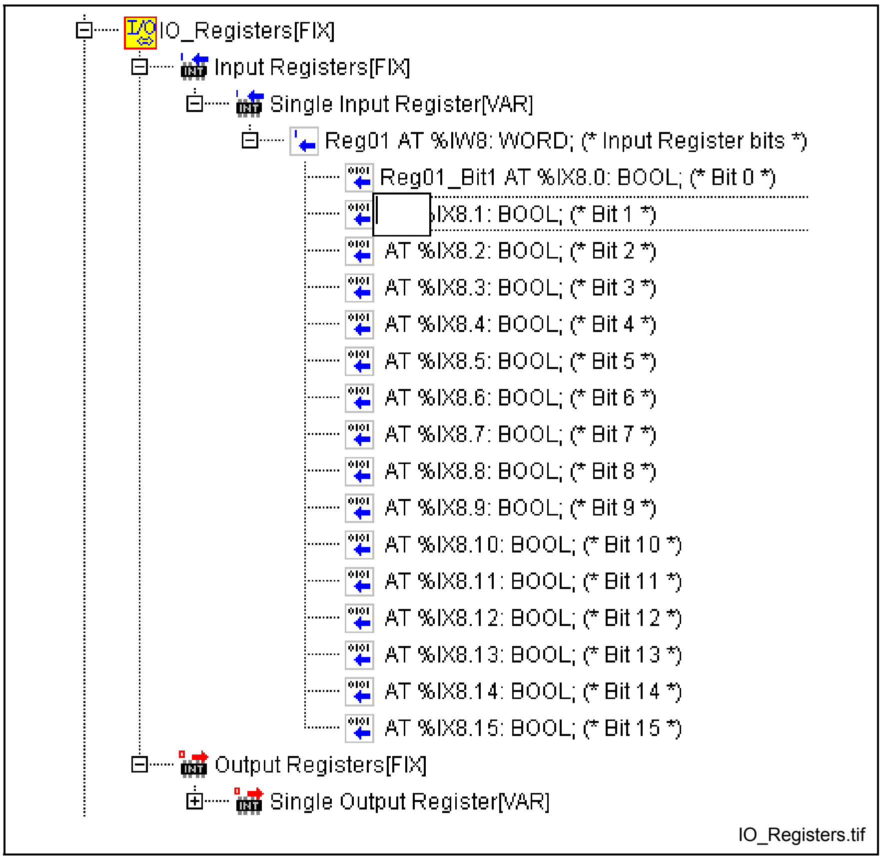


Fig. 7-33: IO_Registers Input and Output Subelements

Note: IndraLogic register bits are accessed from 0-15. VisualMotion register bits are accessed from 1-16.

Cyclic Process Data Types

The following tables list the different cyclic process data types supported in the I/O Image.

Float and Integer Variables

The following table list the VisualMotion variables supported in the Cyclic Process Data:

Type	Data Type	Description
Global Floats	REAL	Valid Global Float addresses are 1 through 32767(default max is 256). Refer to control parameter C-0-0081. The address is checked at runtime since the size of the global variables is configurable.
Program Floats	REAL	Valid Program Float addresses start at 1. The address is checked at runtime since the size of the program variables is configurable.
Global Integers	DINT	Valid Global Integer addresses are 1 through 32767(default max is 512). Refer to control parameter C-0-0080. The address is checked at runtime since the size of the global variables is configurable.
Program Integers	DINT	Valid Program Integer addresses start at 1. The address is checked at runtime since the size of the program variables is configurable.

Table 7-19: Configurable Variables

Card Parameters Input (Read) Access

The following table lists the supported card (control) parameters supported in the cyclic process data area.

Card Parameter (Input)	VM Data Type	PLC Data Type
C-0-0120 Operating Mode	char	SINT
C-0-0125 System Timer Value	ulong	UDINT
C-0-0127 Current PPC-R Temperature	short	INT

Table 7-20: Configurable Card Parameters

Task Parameters Input (Read) Access

The following table lists the supported task parameters supported in the cyclic process data area.

Task Parameter (Input)	VM Data Type	PLC Data Type
T-0-0100 Target Point Number	short	INT
T-0-0101 Segment Status	enum	DINT
T-0-0111 Current X Position	float	REAL
T-0-0112 Current Y Position	float	REAL
T-0-0113 Current Z Position	float	REAL

Table 7-21: Configurable Task Parameters

Axis Parameters Input (Read) Access

The following table lists the supported axis parameters supported in the cyclic process data input memory area.

Axis Parameter (Input)	VM Data Type	PLC Data Type
A-0-0034 Control Cam Currently Active	uchar	USINT
A-0-0100 Target Position	float	REAL
A-0-0101 Commanded Position	float	REAL
A-0-0102 Feedback Position	float	REAL
A-0-0110 Programmed Velocity	float	REAL
A-0-0111 Commanded Velocity	float	REAL
A-0-0112 Feedback Velocity	float	REAL
A-0-0120 Programmed Acceleration	float	REAL
A-0-0121 Programmed Deceleration	float	REAL
A-0-0131 Sercos Control Word	float	WORD
A-0-0132 Sercos Status Word	float	WORD
A-0-0141 Torque Mode Commanded Torque	float	REAL
A-0-0142 Torque Feedback (cyclic)	float	REAL
A-0-0150 Programmed Ratio Adjust	float	REAL
A-0-0151 Programmed Phase Offset	float	REAL
A-0-0156 Phase Offset Velocity Feedback	float	REAL
A-0-0157 Current Phase/ Control CAM Master Offset	float	REAL
A-0-0158 Relative Phase Offset Distance Remaining	float	REAL
A-0-0160 Commanded Ratio Adjust	float	REAL

Axis Parameter (Input)	VM Data Type	PLC Data Type
A-0-0190 Command Data #1	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0191 Command Data #2	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0192 Command Data #3	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0193 Command Data #4	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0194 Command Data #5	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0195 Feedback Data #1	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0196 Feedback Data #2	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0197 Feedback Data #3	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0198 Feedback Data #4	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0199 Feedback Data #5	list of various types	WORD, REAL, INT, DWORD, DINT

Table 7-22: Configurable Input Axis Parameters

Axis Parameters Output (Write) Access

The following table lists the supported axis parameters supported in the cyclic process data input memory area. These parameters must be predefined using A-0-0180 through A-0-0184.

Axis Parameter (Output)	VM Data Type	PLC Data Type
A-0-0190 Command Data #1	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0191 Command Data #2	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0192 Command Data #3	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0193 Command Data #4	list of various types	WORD, REAL, INT, DWORD, DINT
A-0-0194 Command Data #5	list of various types	WORD, REAL, INT, DWORD, DINT

Table 7-23: Configurable Output Axis Parameters

Shared Memory

The shared memory area is dynamically updated and can be accessed by both the motion and logic sides of the firmware. It provides mostly diagnostic information about the system. It resides on the motion side but can be accessed by IndraLogic as read-only text.

The available shared memory items are dependent on the current PLC target. All shared memory information is pre-labeled starting with the letters "MV_" (e.g. "MV_ACTPHASE"). The shared memory items for VisualMotion 11 include:

Variable	Description
MV_ACTPHASE	STRING (81) (*Control Firmware Version, C-0-0100*)
MV_DIAGNUMBER	UDINT (*Diagnostic Code, C-0-0123*)
MV_DIAGTEXT	STRING (81) (*Diagnostic Message, C-0-0122*)
MV_EXTDIAGTEXT	STRING (81) (*Extended Diagnostic, C-0-0124*)
MV_PLCDIAGNUMBER	UDINT (*PLC Diagnostic Code, C-0-1611*)
MV_PLCDIAGTEXT	STRING (81) (*PLC Diagnostic Message, C-0-1612*)
MV_PLCEXTDIAGTEXT	STRING (81) (*PLC Extended Diagnostic, C-0-1613*)
MV_POUERRORLOG	ARRAY[0..29] OF UDINT (* PLC Function Block Error Log, C-0-1620 *)
MV_VERSIONSTRING	INT (*Sercos Communication Phase, C-0-0121*)

Table 7-24: Shared Memory Variables

The data can be accessed from IndraLogic in the *System Variables* or *Global Variables* areas. To access shared memory, press the F2 key while on a variable name location and select the relevant shared memory variable from Global Variables and/or System Variables.

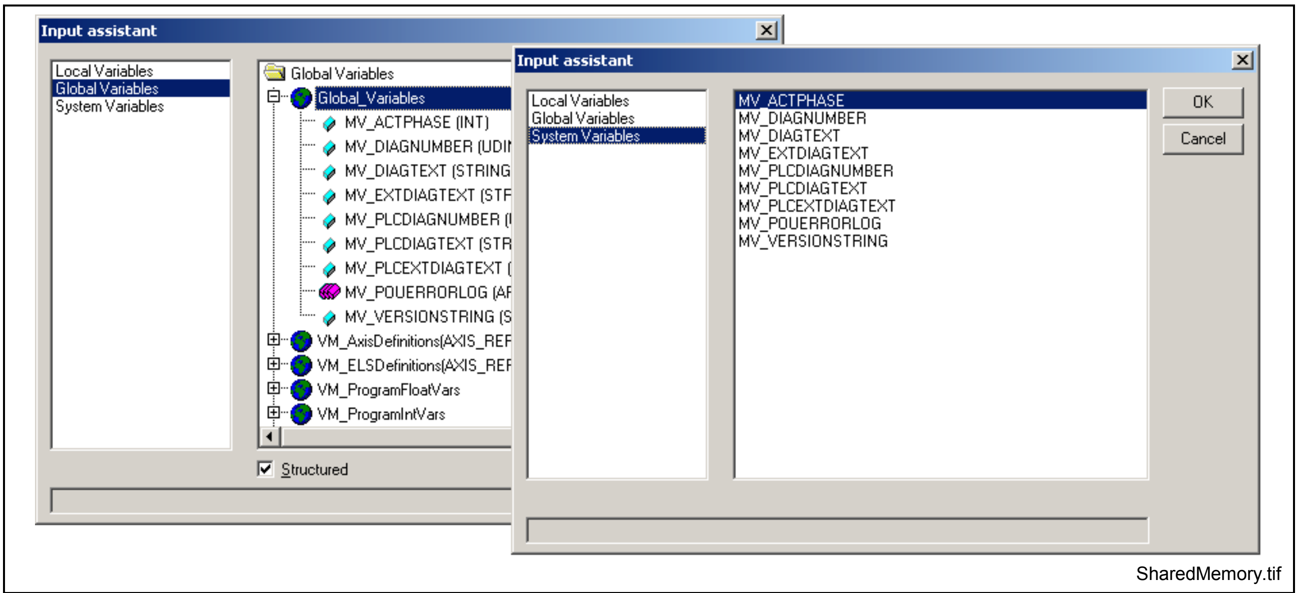


Fig. 7-34: Shared Memory Variables

Configuring a Profibus Fieldbus Master

VisualMotion 11 includes a number of Bosch Rexroth fieldbus slave device configuration files (GSD) as part of the firmware targets. To include a GSD file that is not part of the current target, refer to

Install a GSD File on page 7-37 for details.

Append a Profibus Master

Step 1: Profibus Configuration

1. From the *Resources* tab, double-click on ***PLC Configuration***.
2. Expand the configuration tree structure down to the control level.
3. Right-click over Control and select ***Append Subelement*** ⇒ ***DPM01_PC104…***

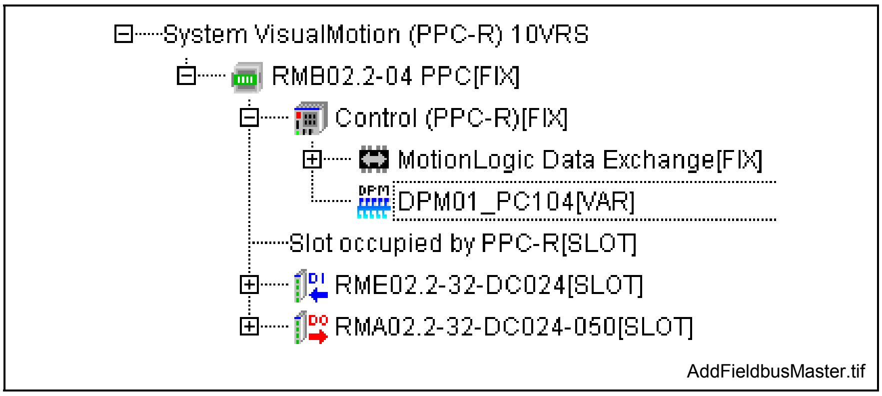


Fig. 7-35: Add Profibus Fieldbus Master

4. Select the *DP Parameters* tab and set the **Station address** and **Highest station address**. The Highest station address refers to the device address of the last slave device.
5. Switch to the *Bus Parameters* tab and set the **Baud rate** for the fieldbus master to match to system. Profibus baud rate is commonly set to 120000,00 kBits/s (12 Mbits/s).

Note: Refer to the IndraLogic *Online Help System* for additional fieldbus master parameter settings.

Append a Profibus Slave

Step 2: Profibus Configuration

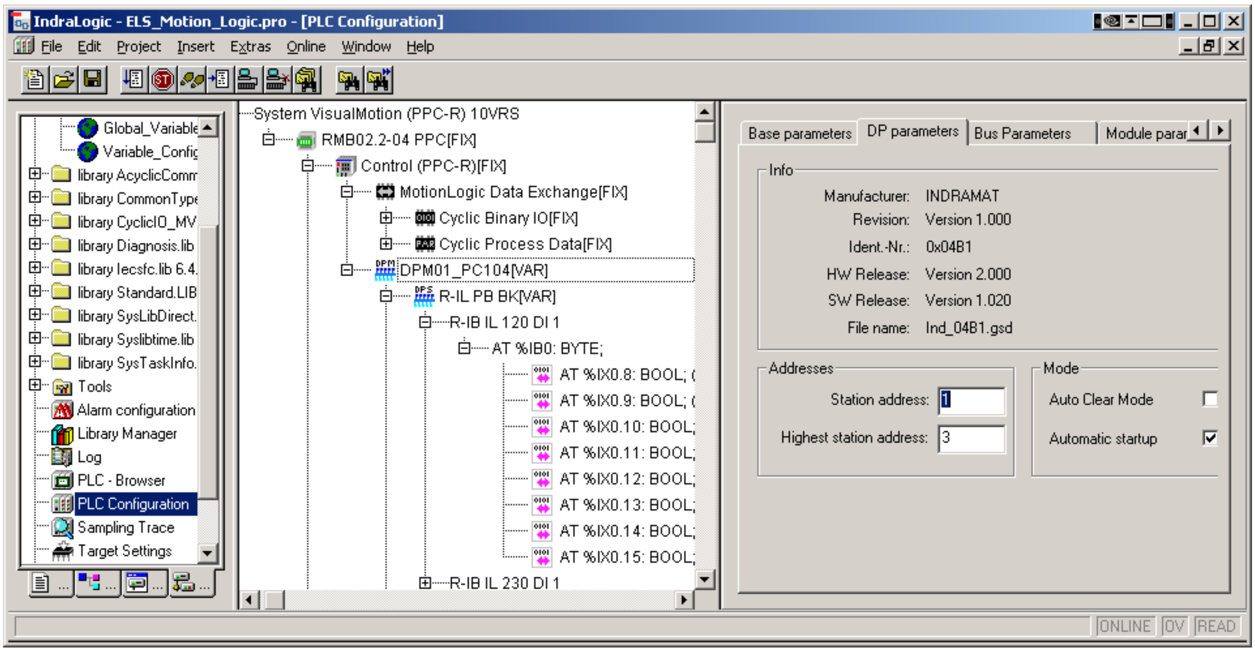
1. Right-click on *DPM01_PC104*, select ***Append Subelement*** and then the relevant fieldbus slave type.

Note: The available slave types are dependent on the pre-configured Profibus master added in step 1.

The Fieldbus slave device is added to the tree structure below the Fieldbus Master.

2. Select the DP Parameters tab and configure the slave devices.

Note: The *Station address* for each slave device in IndraLogic must match the *Device Address* configured for the slave device.



IND_FB_Config.tif

Fig. 7-36: Fieldbus Configuration

3. Configure the remaining options for the selected fieldbus slave device.

Note: The status of fieldbus I/O can only be monitored online if the data is used in the PLC program.

Note: Refer to the IndraLogic *Online Help System* and *Configuring Profibus Modules* in the *PLC Programming with Rexroth IndraLogic 1.0 manual, DOK-CONTRL-IL**PRO*V01-AW01-EN-P* for details.

Install a GSD File

1. In IndraLogic, close the PLC Configuration window if it is currently opened.
2. From the *Resources* tab, expand the **Tools** folder and double click ***Install / Deinstall GSD-File***.
3. From the *IndraLogic GSD Installer* window, click on the **Select** button.
4. Locate and open the relevant fieldbus slave GSD file.
5. From the top half of the *IndraLogic GSD Installer* window, highlight the relevant slave device, then highlight the VisualMotion control hardware and click the **Add** button.

Note: The *.gsd file contains configuration properties for the specific device and is required for registering the slave device with the fieldbus master.

Configuring a DeviceNet Fieldbus Master

VisualMotion 11 includes a number of Bosch Rexroth fieldbus slave device configuration files (EDS) as part of the firmware targets. To include an EDS file that is not part of the current target, refer to Install a DeviceNet EDS Configuration Files on page 7-40 for details.

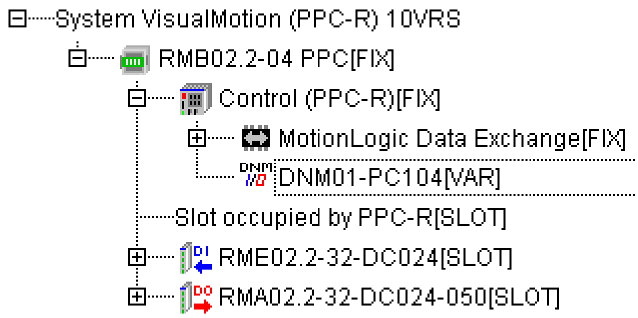
The following steps provide a summary for configuring a DeviceNet network in IndraLogic: The detailed steps are described in the sections that follow.

1. Add DeviceNet *.eds configuration files to IndraLogic.
2. Configure DeviceNet master parameters in the MotionLogic Data Exchange area of the I/O Image.
3. Append and configure DeviceNet slaves and download complete DeviceNet configuration to the control.

Append a DeviceNet Master

Step 2: Configure DeviceNet Master

1. From the *Resources* tab, double-click on ***PLC Configuration***.
2. Expand the configuration tree structure down to the Control level.
3. Right-click over Control and select ***Append Subelement*** ⇒ ***DNM03_PC104…***



AddFieldbusMaster.tif

Fig. 7-37: Add DeviceNet Fieldbus Master

Base Parameters

4. Select the *Base Parameters* tab and set the **Node id** for the master. The Node id identifies the DeviceNet master within the IndraLogic environment.

Device Net Parameters

5. Select to the *Device Net Parameters* tab and set the **baudrate** for the fieldbus master to match to system. DeviceNet baudrate is commonly set to 500 Kbits.
6. Set the **Address** for the master. The address is used for identification within the DeviceNet network.

Note: I/O handling for DeviceNet is performed using data consistency. The same data is used from a single I/O image instance. This ensures data consistency across multiple tasks accessing the same data.

Append a DeviceNet Slave

1. Right-click on *DNM01-PC104* and select ***Append Subelement***.
2. Select the relevant DeviceNet slave. Select **LK104DNS.EDS...** for the Rexroth DeviceNet slave card.

Note: The list of available DeviceNet slaves that are displayed comes from the system default and user installed *eds configuration files. Refer to Install a DeviceNet EDS Configuration Files on page 7-40 for details.

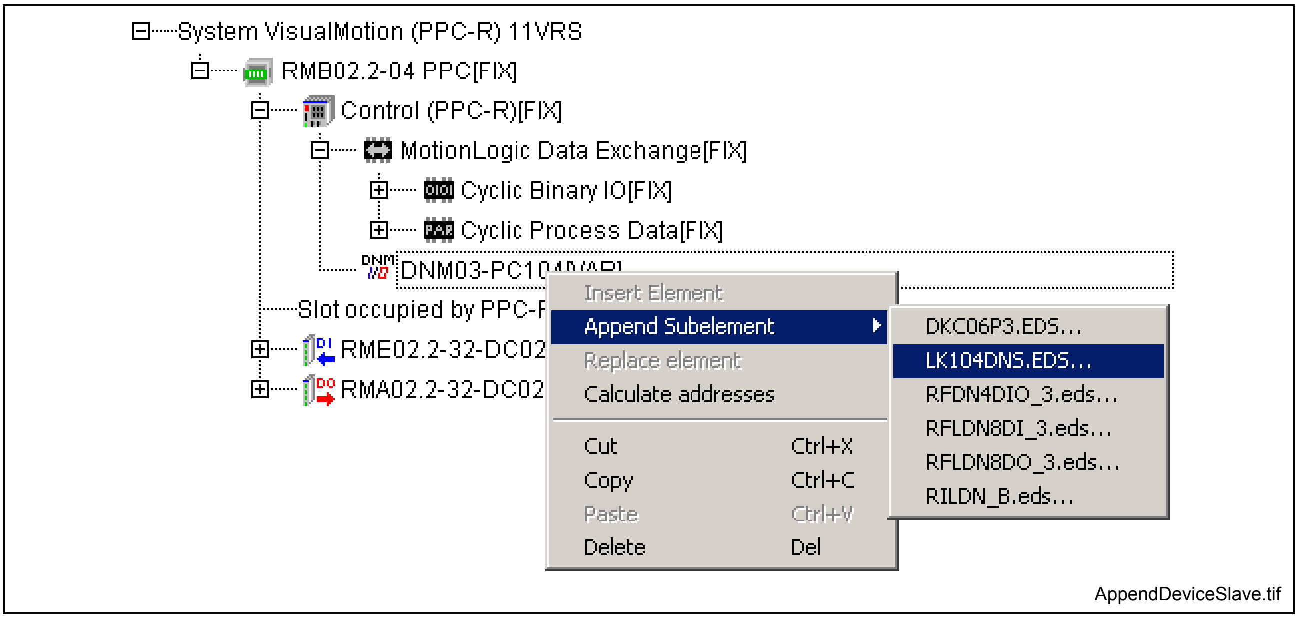


Fig. 7-38: Appending DeviceNet Slave Interface

3. Select the newly appended DeviceNet slave device to display the corresponding property sheets to the right of the I/O Image tree structure.

Base Parameters

4. The *Base Parameters* tab displays the starting input and output memory locations. By default, these values are automatically calculated and set by IndraLogic every time a new element is added before the current one.

Device Net Parameters

5. Select the *Device Net Parameters* tab and set the address for the slave device. (**Note: Must match master settings**)

I/O Connection Configuration

6. Select the *I/O Connection Configuration* tab and setup the size for the DeviceNet slave inputs and outputs.

Parameters

7. The *Parameters* tab is available for selected DeviceNet slave types. Unsupported types do not display any settings or selections.

Module Parameters

8. Select the *Module Parameters* tab and enable or disable DeviceNet slave diagnostic communication.

Note: The status of fieldbus I/O can only be monitored online if the data is used in the PLC program.

Note: Symbolic names can be assigned to DeviceNet IO. Refer to Applying Names to Reco 02 I/O Modules on page 7-12 as an example.

Install a DeviceNet EDS Configuration Files

Use the following steps to add DeviceNet *.eds files to IndraLogic:

Step 1: DeviceNet *.eds Files

1. From IndraLogic select ***Extras ⇒ Add configuration files…*** while not logged on.
2. Locate the DeviceNet *eds file and select **Open**.

Note: The *.eds file contains configuration properties for the specific device and is required for registering the slave devices with the fieldbus master.

Default system *.eds configuration files are stored in the "*config*" folder within the target folder. User installed *.eds configuration files are stored in the "*userconfig*" folder of the same target folder.

Folder Structure	Description
\Program Files\Rexroth\IndraLogic\Targets	Default IndraLogic target folder location
\VM_ML_11VRS_PPC_R\ config	folder location of system default eds files for PPC-R
\VM_ML_11VRS_PPC_R\ userconfig	folder location of user installed eds files for PPC-R
\VM_ML_11VRS_PPC_P\ config	folder location of system default eds files for PPC-P
\VM_ML_11VRS_PPC_P\ userconfig	folder location of user installed eds files for PPC-P

Table 7-25: EDS Configuration File Default Folder Locations

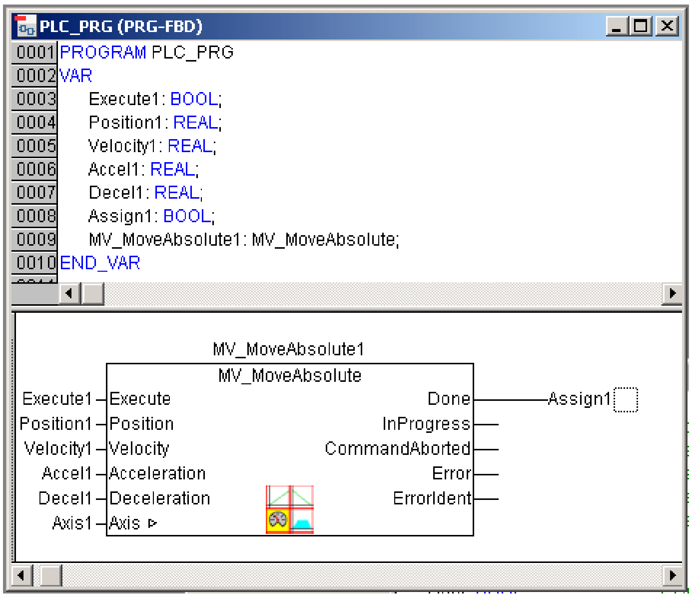
Functions and Function Blocks

In addition to common PLC functionality such as counters and timers, function blocks can access either direct memory in the motion control or call sub-functions from the motion control. Pre-configured function blocks are provided for a variety of tasks, including:

- Simple data function access (VisualMotion registers, variables, control parameters, etc.)
- Complex data access (including VisualMotion list and Sercos parameters)
- Motion commands (e.g., single axis and ELS)
- Math calculations

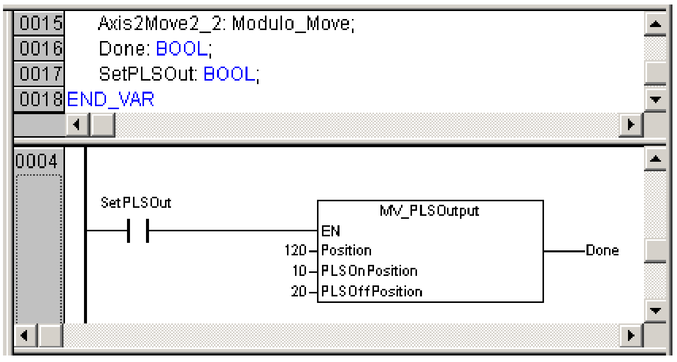
They are available in the IEC libraries, which can be viewed by selecting ***Library Manager*** from the *Resource* tab.

Function blocks offer the most flexibility in motion/logic data exchange because they call control code functions that can access any data or functionality in the motion system. Because each instance of a function block has allocated memory, it is possible to perform operations that take many PLC scans to perform. Functions, on the other hand, do not require an instance and can be called inline.



FB_Example.tif

Fig. 7-39: Motion Function Block Example



FN_Example.tif

Fig. 7-40: Function Example

Function Block Programming Considerations

The following general issues should be considered when using function block programming:

1. Function blocks that access the Sercos service channel require a minimum of 40-50 ms to complete.

   Some examples include:

   a. MSV_ReadParameter when accessing S and P drive parameters.

   b. MV_MoveAbsolute when the velocity, acceleration, and deceleration inputs are not in the cyclic channel (MDT).

2. The evaluation of the "Done" and "Error" outputs is highly recommended for all function blocks. This ensures a complete and error free state of the current function block before proceeding to the next command.

3. MSV_AcyclicComm function blocks:

   a. These function blocks should not be considered for use in a Sercos synchronous task due to their long completion times.

   b. Use the "Done" output to make sequential calls when using multiple function blocks to access drive parameters over the Sercos service channel.

4. MV_Motion function blocks:

   a. The cyclic channel is used to transmit relevant data if the items that are being transmitted are in the MDT. For example, if the velocity, acceleration and deceleration parameters are in the MDT, then the values are written to the relevant drives over the cyclic channel.

   b. The only variables that are double-buffered are Position and Phase Offset. These are the only variables that will be transmitted in the same Sercos cycle for all defined axes.

   c. The following function blocks simultaneously write to all defined axes within the same Sercos synchronous task:

      - MV_Abort
      - MV_Stop
      - MV_MoveRelativeOnly
      - MV_MoveAbsoluteOnly
      - MV_MoveAdditiveOnly

   d. The following function blocks can also be used to simultaneously write to all defined axes within the same Sercos synchronous task if their Velocity, Acceleration, and Deceleration input variables are setup using the MV_MoveSetup function block:

      - MV_MoveRelative
      - MV_MoveAbsolute
      - MV_MoveAdditive

   e. To ensure simultaneous move commands within the same Sercos cycle (Sercos synchronous task), the relevant drives must already be in single-axis mode and enabled (AF). If not, the function block will command a single-axis mode change but the moves will no longer begin within the same Sercos cycle.

IEC Libraries

A library is a list of objects that can be used in different projects. A default library called standard.lib is available in IndraLogic, but it is also possible to create custom libraries with IndraLogic and encrypt the libraries to protect proprietary information. In addition, it is possible to call motion control functions as library objects. The following table lists the libraries that are supported by VisualMotion and IndraLogic:

Library Name	Description	Loaded By Default
IecSfc.lib	Function block and data types for executing standard IEC Sequential Function Chart (SFC) actions	X
MSV_AcyclicComm.lib	Function blocks for reading parameter data	X
MSV_CheckRtv.lib	Used for monitoring proper operations on the data access mechanisms such as Division, Access to an indicated array element, Value assignment to a variable with limited value range, and Pointer access	
MV_CyclicIO.lib	Defined data types for binary portion of I/O Image configuration	X
MV_ErrorLog	This library provides the PLC programmer with the ability to log PLC errors and make them available to the control via the Shared Memory system variable MV_PouErrorLog.	
MV_Motion.lib	Function blocks and data types for executing single axis and ELS based motion from within the PLC task	X
MV_UserMotion	Allows the user to configure motion control profiles from a high priority Sercos event task	
NetVarUDP_Lib_v23.lib	Network global variables access between PPC and Ethernet via UDP. (PPC-R22.1 only)	
RIL_CommonTypes.lib	Defined data types for common use functions, such as error structure, axis_ref, etc.	X
RIL_DeviceNet.lib	Function blocks for managing the DeviceNet master interface	
RIL_Diagnosis.lib	This function is used to ensure compatibility of firmware	X
RIL_ProfibusDP.lib	Function blocks for managing the Profibus master interface	
RIL_Utilities.lib	Utility library containing functions and function blocks for various tasks, such as type conversions for AcyclicComm_MSV01.lib, time conversions, etc.	
RIL_VexUtil.lib	The library enables the user to transmit keystrokes on Rexroth VEH/VEP HMIs in a safe way to the control.	
RMC_PLCopen.lib	Standard PLC Open function blocks used for administrative and single axis motion	
Standard.lib	Function blocks for executing standard PLC capabilities such as timers, counters, latch registers, string manipulation, etc.	X
SysLibCallback.lib	Management of PLC Events	
SysLibCom.lib	Send and receive serial port messages via X16 on the PPC. X16 must be configured for "SysLibCom" under *Tools ⇒ Control Settings*.	
SysLibDirect.lib	Hidden library, not for user access	X
SysLibFile.lib	Functions can be used to create, open, close, delete, rename, write to or read from files on the PLC	
SysLibFileAsync	This library is the same as SysLibFile.lib, however function blocks instead of functions are used and the processing is asynchronous	
SysLibPlcCtrl.lib	Functions used to start, reset, stop, and shutdown a PLC program. Also, retain variables can be saved and restored	
SysLibSockets.lib	Used with network global variables (PPC-R22.1 only)	
SysLibSocketsAsync.lib	This library is the same as SysLibSocket.lib, however function blocks instead of functions are used and the processing is asynchronous	
SysLibStr.lib	An action control library for IEC 61131-3 SFC actions	
SysLibTime.lib	Function blocks used for reading the real-time clock of the local computer	X
SysTaskInfo.lib	Used to monitor task processing times	X
Util.lib	Utility library containing functions and function blocks for various task such as PID loops, Calculus functions, signal generators, data monitoring, and bit/byte management	

Table 7-26: Libraries Supported by VisualMotion and IndraLogic

To view the libraries in the current IndraLogic project, select ***Window ⇒ Library Manager*** or double-click on **Library Manager** under the *Resource* tab.

To add a library to the current IndraLogic project, right-click in the window displaying the installed libraries and select **Additional Library …**

Note: The default folder locations for IEC libraries are as follows:

Program Files/Rexroth/IndraLogic/Targets/<firmware target>

Any library starting with "SysLib" is stored in the **/internal** folder. All others are stored in the **/lib** folder.

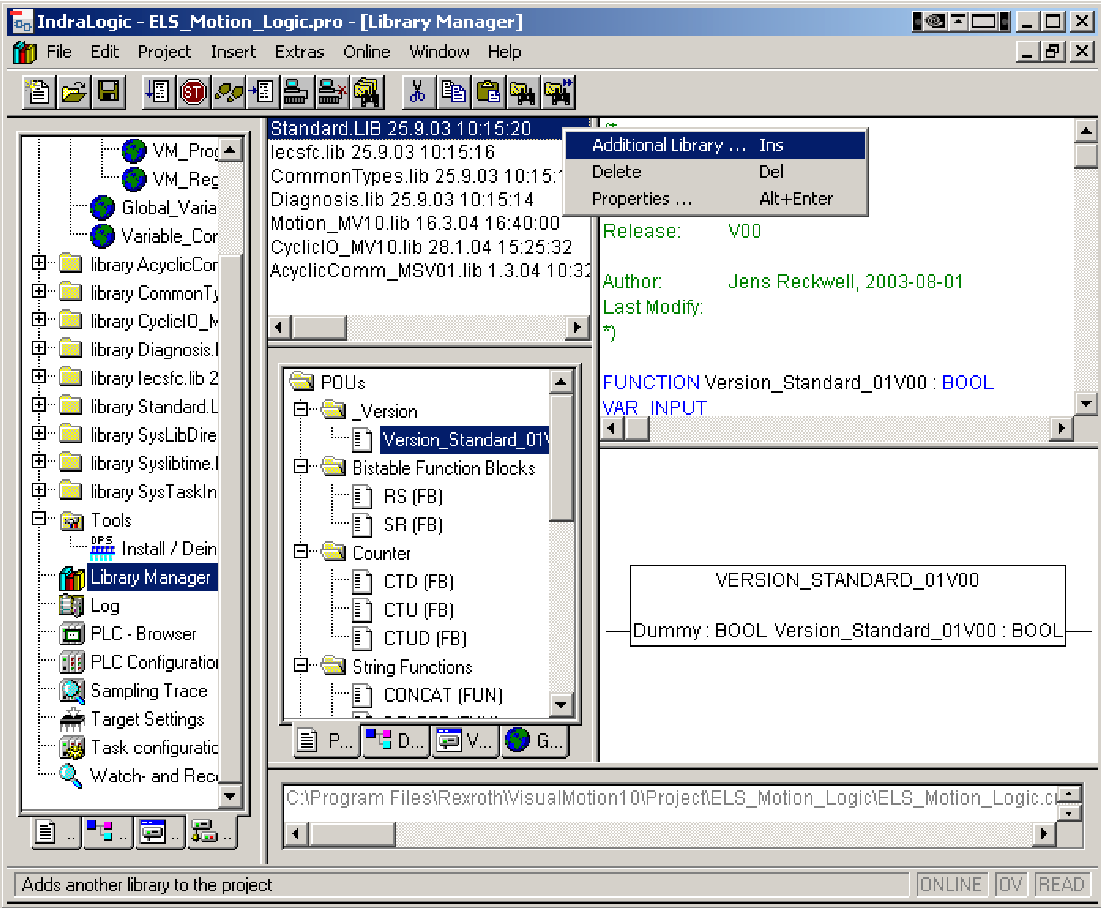
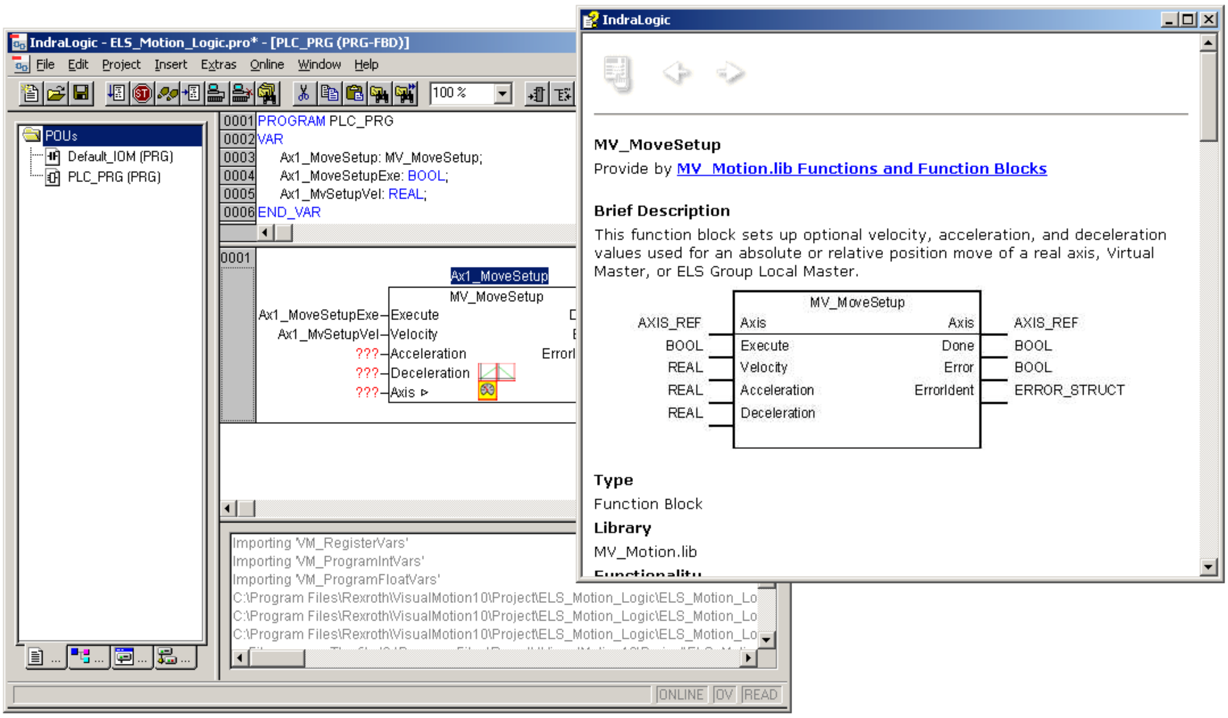


Fig. 7-41: IndraLogic Library Manager

For descriptions of the supported libraries, refer to the IndraLogic Help System.

Context Sensitive Help

Context sensitive help can be accessed for any Data Type, Function, and Function Block by highlighting the name and pressing the F1 key.



FbContextHelp.tif

Fig. 7-42: Context Sensitive Help

Rexroth
Bosch Group

IndraLogic Communication Channels

IndraLogic software can communicate with a PPC control via a serial interface (RS232), EtherNet or a PCI Bus (PPC-P11.1) connection. The communication channels used between IndraLogic and the control can be configured in IndraLogic or VisualMotion Toolkit.

Note: In VisualMotion Toolkit, the **IndraLogic** tab under menu selection ***Tools* ⇒ *Options…*** can be used to manually configure IndraLogic communication channels. Refer to section 13.10, Tools Menu, in volume 2 of the *VisualMotion 11 Functional Description* for details.

The following table lists the IndraLogic communication channels and the maximum connections to one control at any given time:

Communication Channel	Description	Maximum Connections
SCP VM	IndraLogic SCP/SIS Gateway driver for VisualMotion (SIS Tunneling)	Only one (1) IndraLogic connection can be configured via serial or EtherNet
RS232 PPC	PPC over RS232 connection (No SIS)	1 serial connection on X16 when configured as IndraLogic (C-0-0012)
TCP/IP Level 2 Route (PPC) (PPC-R22.1 only)	PPC over TCP/IP EtherNet connection (No SIS)	Up to 8 connections
User Configured	Last user selected communication channel will be used.	Depends on communication channel selected in IndraLogic

Table 7-27: IndraLogic Communication Channels

The following figures illustrates the different communication channels described in Table 7-27.

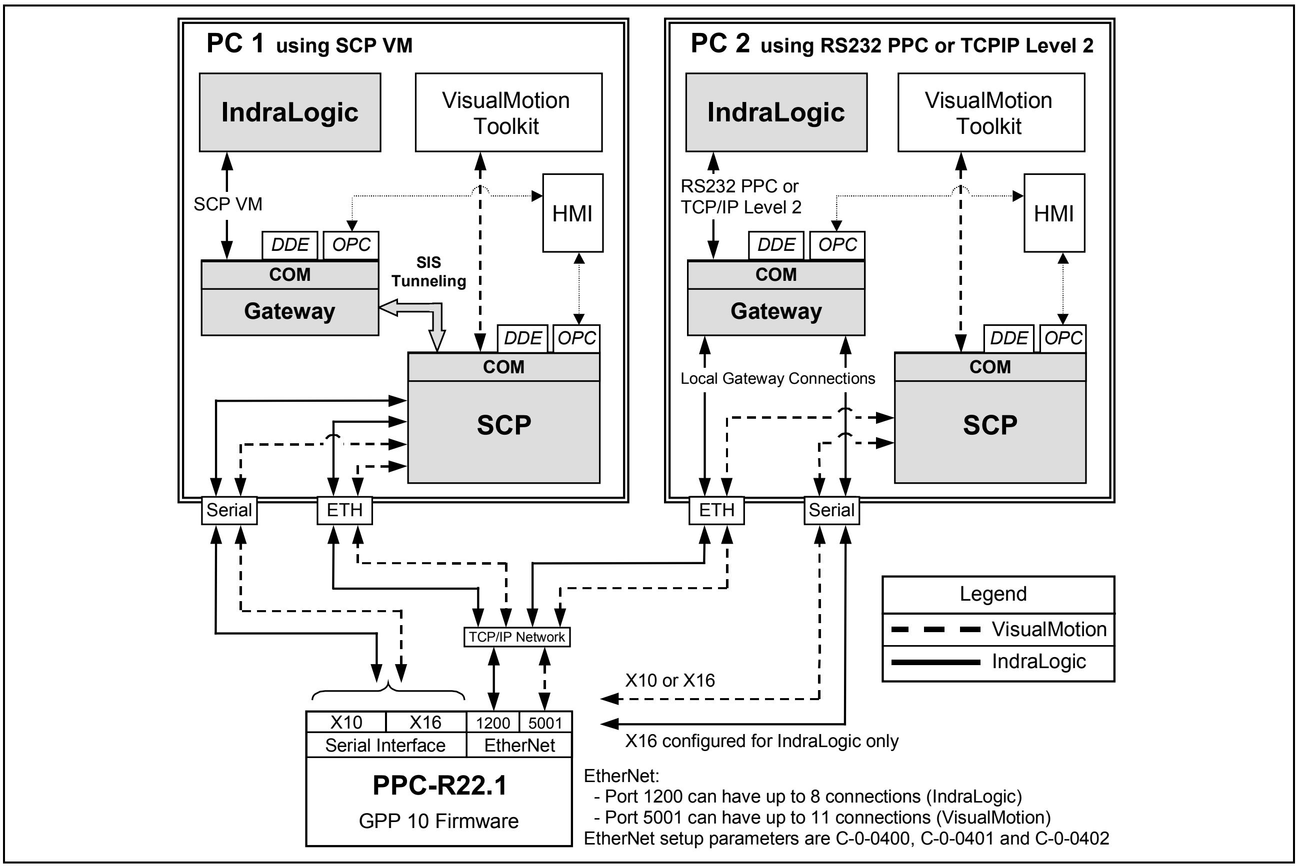


Fig. 7-43: Communication Access for the PPC-R22.1

Note: When a PPC-P11.1 control is installed in a host PC, a client PC can only access the motion functionality using SCP to SCP remote access.

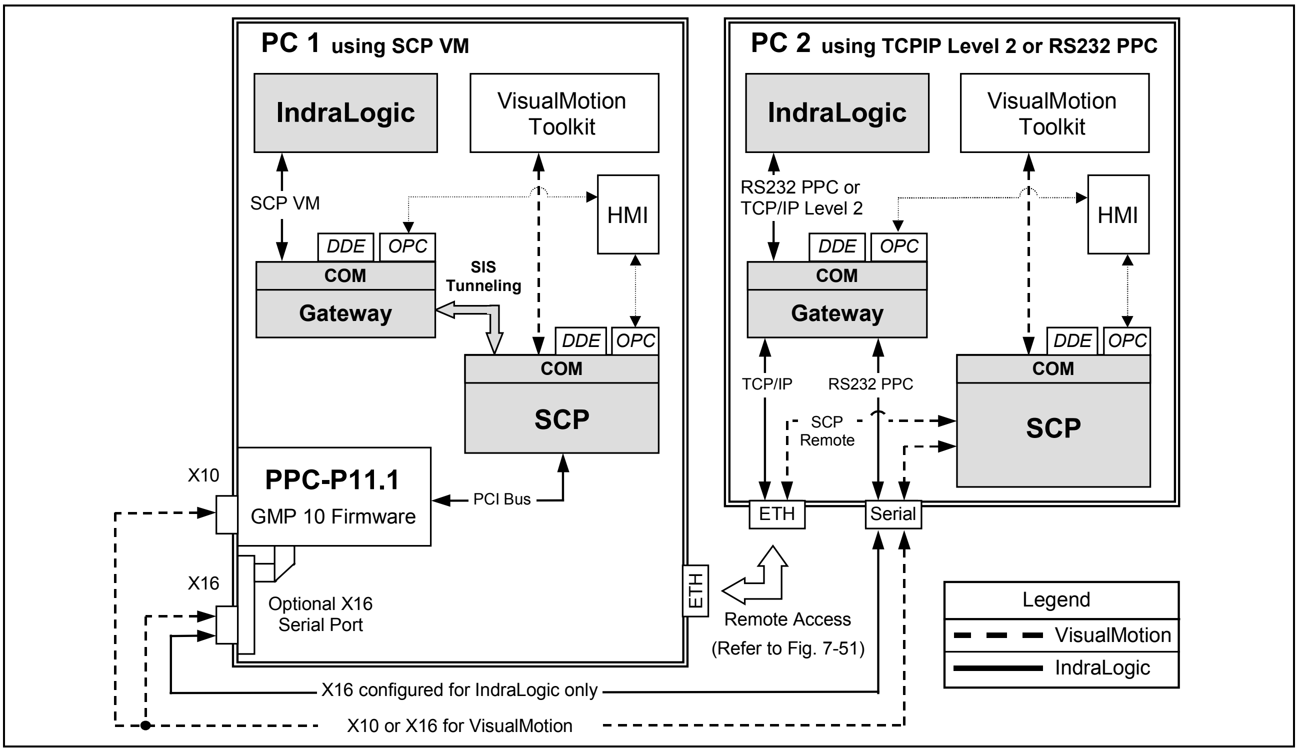


Fig. 7-44: Communication Access for the PPC-P11.1

SCP VM (SIS Tunneling)

SCP VM is a communication channel that enables communication between IndraLogic and a VisualMotion control via the SCP server. The Gateway interface provides the connection between IndraLogic and the SCP. When using SCP VM, the communication between the Gateway and the SCP is referred to as SIS Tunneling. SIS Tunneling is the process of embedding IndraLogic messages into SIS messages through the SCP to the control.

Note: Only one IndraLogic connection can be configured to use the SCP VM. Additional client IndraLogic connections must use either an RS232 PPC or a TCPIP Level 2 (EtherNet) channel.

Use the following steps to manually configure the SCP VM communication channel:

Note: Ensure that the IndraLogic Gateway server is not currently running. Refer to step 3 on page 7-53 for details.

1. From the IndraLogic main menu, select ***Online ⇒ Communication Parameters***.
2. To add a communication channel, click the **New** button, name the channel, and select **SCP VM**.

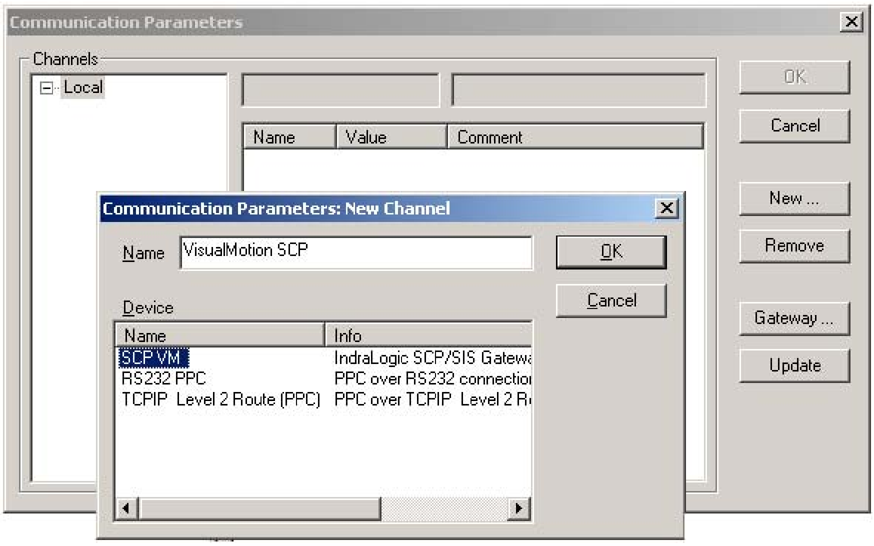


Fig. 7-45: Communication Channel Configuration for SCP VM

3. Set the SCP VM Type by double-clicking directly in the *Value* field or using the arrow keys to scroll between the previously configured devices in the SCP Systemconfigurator.

Note: Refer to section 18.3, *Configuring the SCP Server*, in volume 2 of the *VisualMotion Functional Description*, for details.

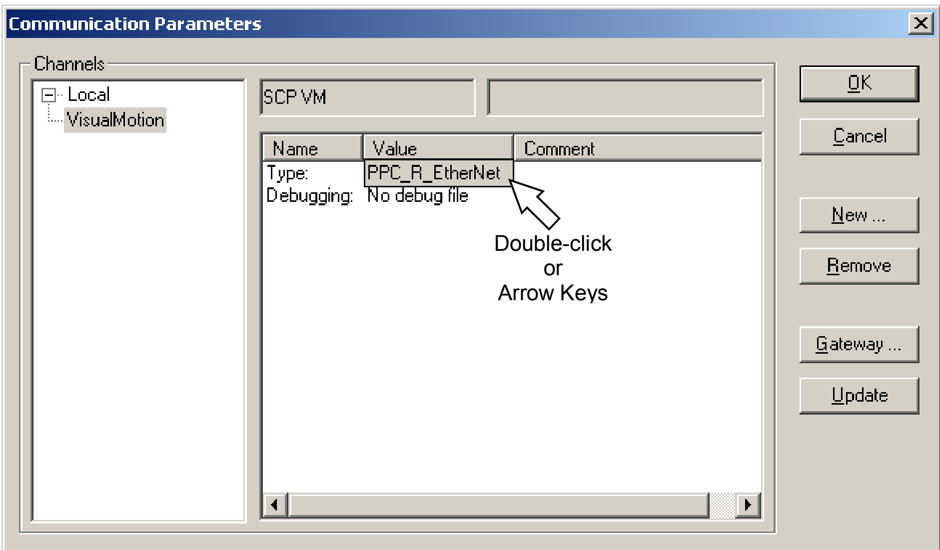


Fig. 7-46: Setting Communication Type for SCP VM

RS232 PPC

The RS232 PPC communication channel is a direct serial connection to the control's X16 port. The Gateway bypasses the SCP server and connects directly to the control. In VisualMotion, the X16 communication port parameter C-0-0012 must be configured for IndraLogic (online mode required).

Note: RS232 PPC provides a faster transmission rate between IndraLogic and the control in comparison to the SCP VM.

Use the following steps to manually configure the RS232 PPC communication channel:

1. From the IndraLogic main menu, select ***Online ⇒ Communication Parameters***.
2. To add a communication channel, click the **New** button, name the channel, and select **RS232 PPC**.

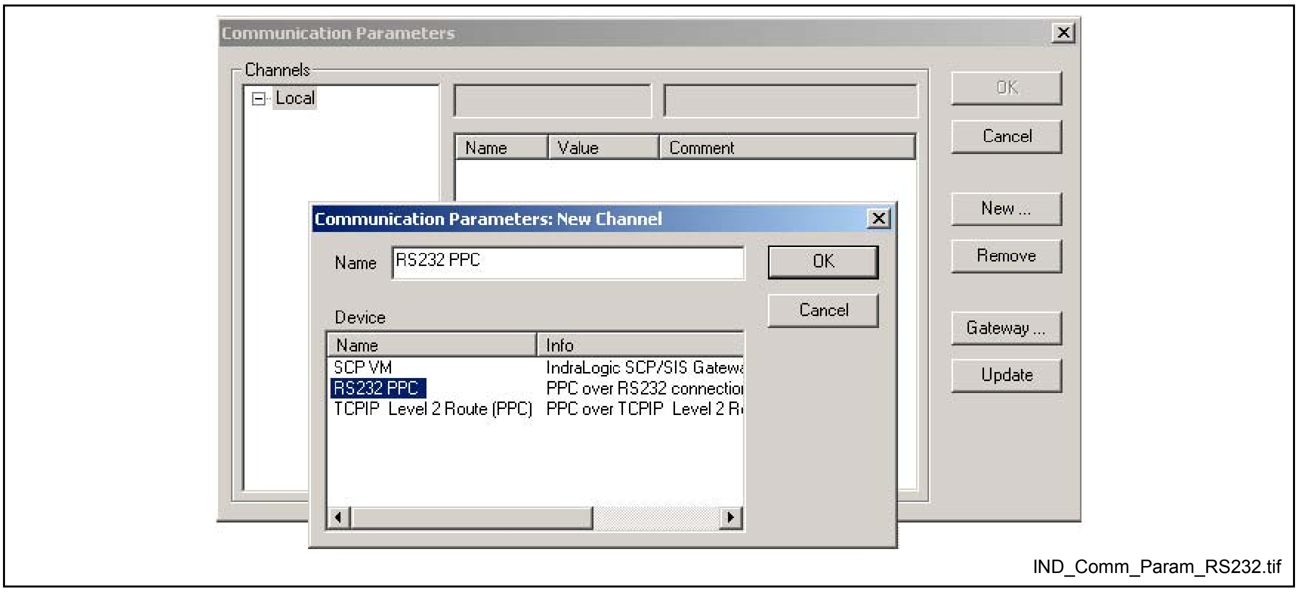


Fig. 7-47: Communication Channel Configuration for RS232 PPC

3. Set the **Port** to match the PC port and the **Baudrate** to match the current settings of the X16 communication port (C-0-0004) on the control. Selections are made by double-clicking or using the arrow keys in the *Value* field.

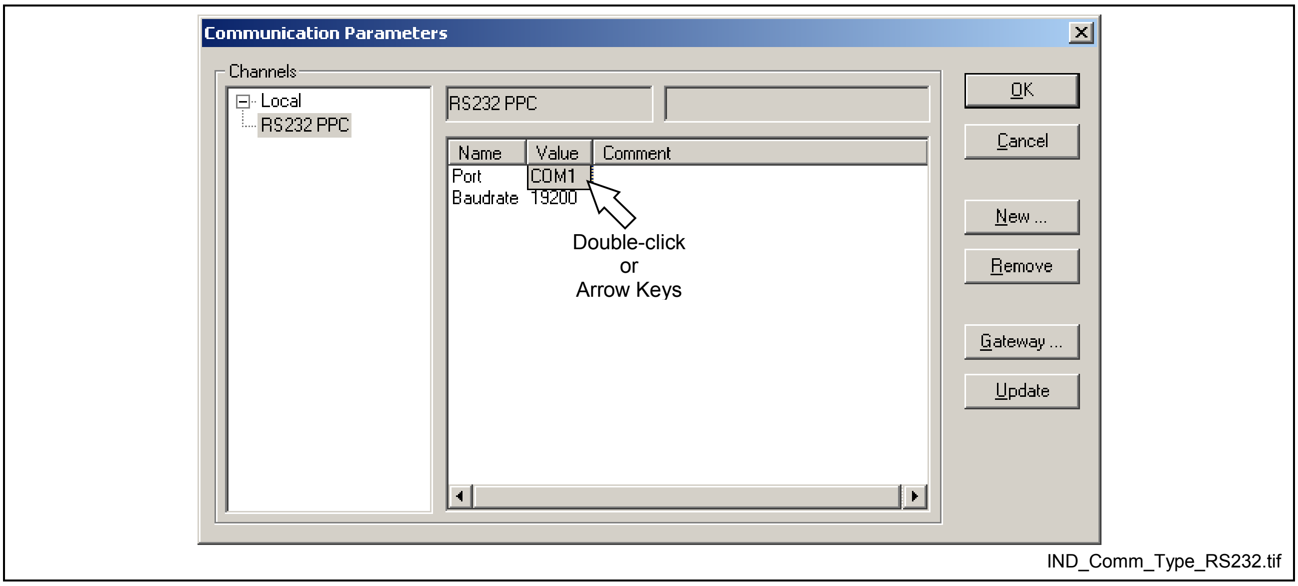


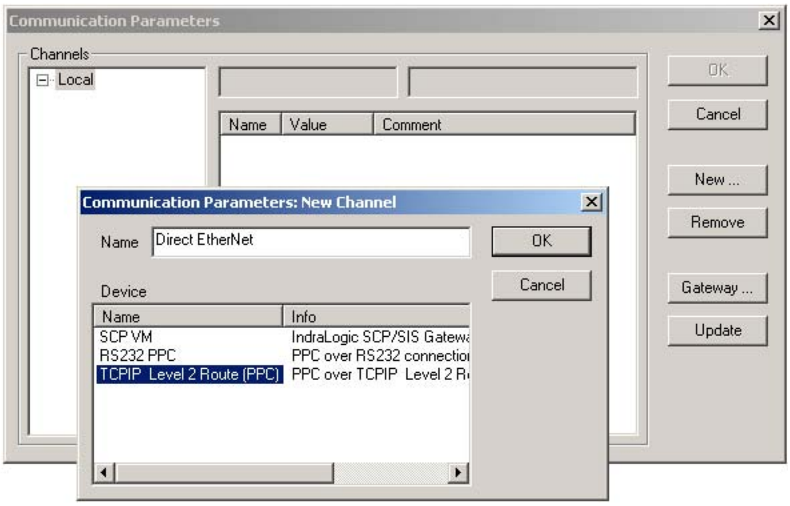
Fig. 7-48: Setting Communication Type for RS232 PPC

TCPIP Level 2 Route (PPC)

The TCPIP Level 2 communication channel is a direct EtherNet connection used only for the PPC-R22.1 control. This communication channel provides the fastest transmission rate and can have up to 8 simultaneous connections via the EtherNet port 1200.

Use the following steps to manually configure the TCPIP Level 2 communication channel:

1. From the IndraLogic main menu, select ***Online*** ⇒ ***Communication Parameters***.
2. To add a communication channel, click the **New** button, name the channel, and select **TCPIP Level 2 Route (PPC)**.

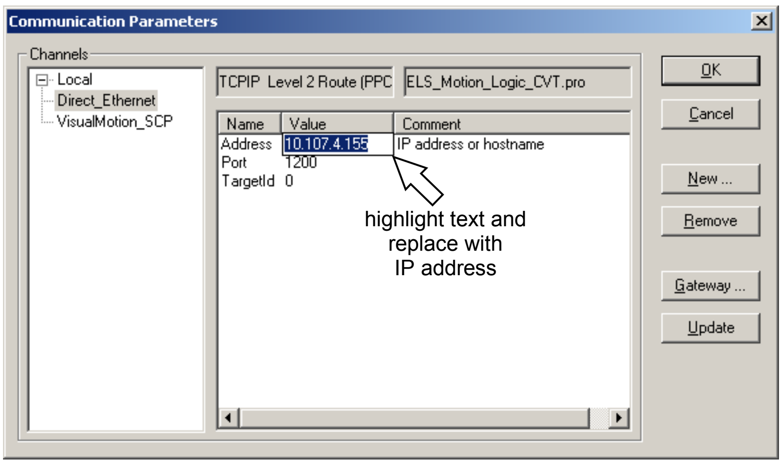


IND_Comm_Param_TCPIP.tif

Fig. 7-49: Communication Channel Configuration for TCPIP Level 2 Route

3. Set the IP address to match the IP address set in control parameter C-0-0400 for the relevant PPC-R22.1 control. Highlight the current text "localhost" and enter an IP address using dot notation, i.e., 010.152.45.125. (Do not use leading zeros. e.g., 010.152.**0**45.125)

Note: Contact your local IT department for the assignment of a unique IP address.



The Port setting must be set to 1200.
The TargetId is set to 0 for the contol.

IND_Comm_Type_TCPIP.tif

Fig. 7-50: Setting Communication Type for TCPIP Level 2 Route

Remote Access to a VisualMotion Control

Remote access is the ability for one PC to gain access to another PC via DCOM (Distributed Component Object Model). DCOM is a protocol that enables software components to communicate directly over a network. The configuration of DCOM is independent of VisualMotion and is performed in the Windows environment.

IndraLogic, on one PC, can be configured to remotely access a second PC hosting a VisualMotion control. Both the PC hosting the control and the client PC must have DCOM configured with appropriate security settings in order to communicate. Refer to DCOM Configuration on page 7-54 for details.

The following two remote access levels can be used for IndraLogic:

- Gateway to Gateway remote access
- Gateway to SCP remote access

Gateway to Gateway Remote Access

This remote access level allows the Gateway server on an client PC to remotely connect to the Gateway server of the PC hosting a VisualMotion control.

Note: The Gateway server of the host PC must be online and running in order to communicate. Only the communication channels configured on the host Gateway can be viewed and selected to access the connected control.

This following figure shows a Gateway to Gateway remote access:

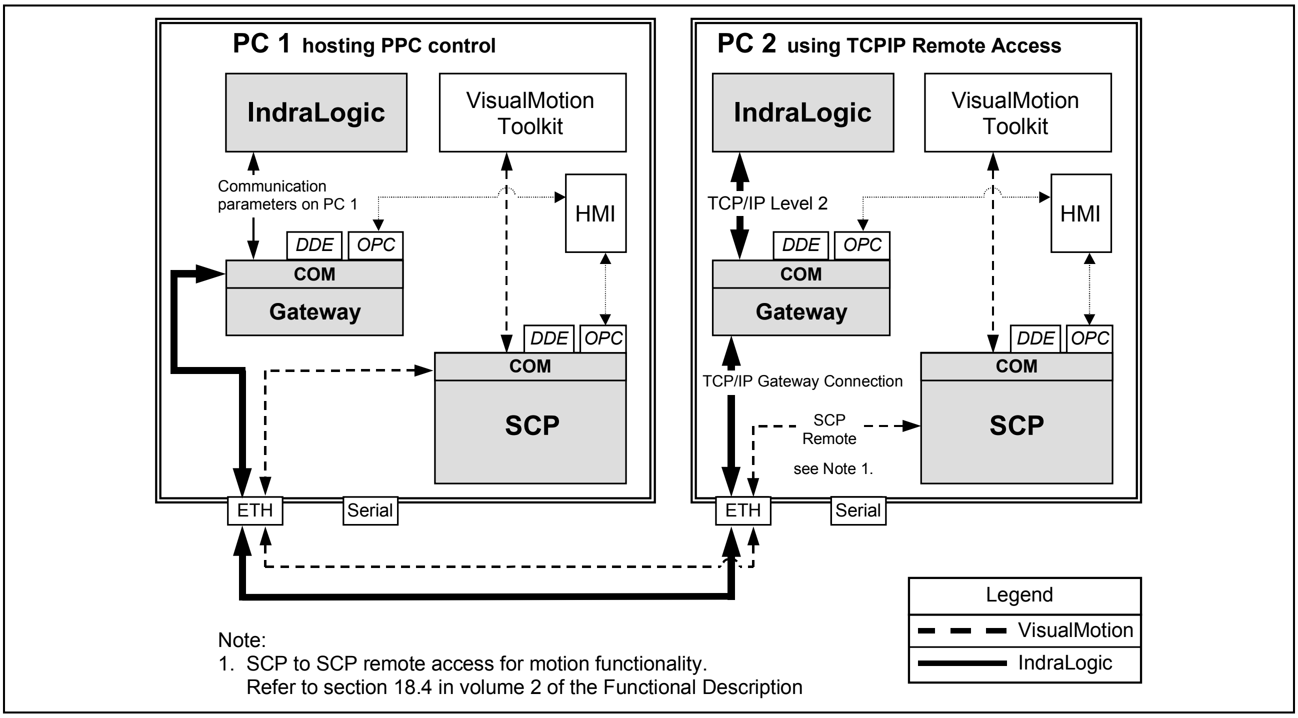
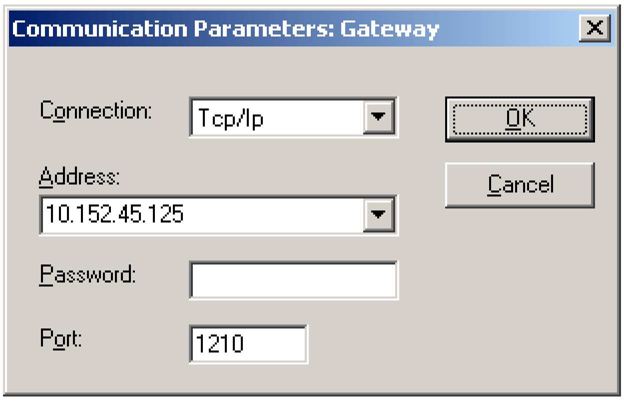


Fig. 7-51: Remote Access (Gateway to Gateway Connection)

Use the following steps to configure IndraLogic's Gateway for remote access:

1. Launch IndraLogic and select ***Online*** ⇒ ***Communication Parameters…***
2. Click on the **Gateway…** button.

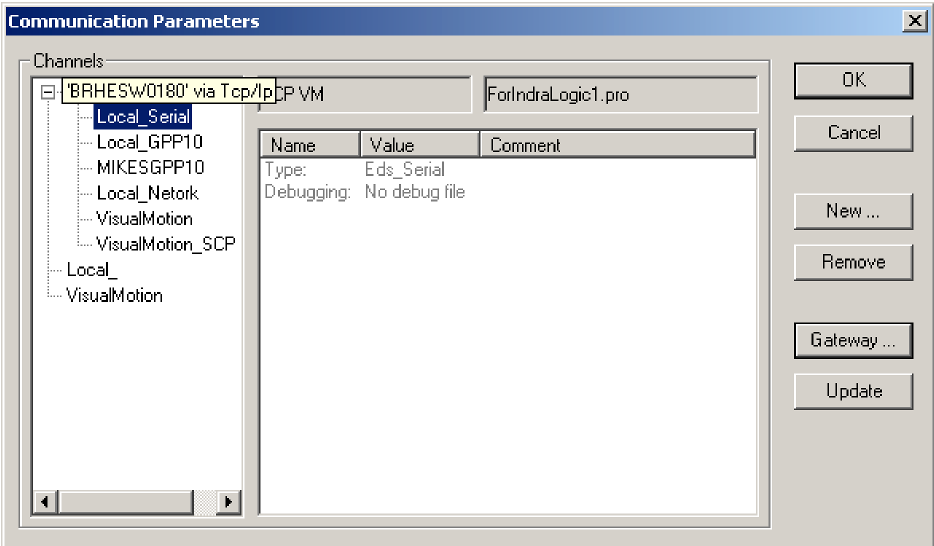
3. Select the *Connection* type as "**Tcp/Ip**", set the host PC's IP address, using dot notation, or enter the computer name of the host PC to access and use the default port 1210. The password field is not used.



Gateway_Tcp_Ip.tif

Fig. 7-52: Gateway using TCP/IP

Once a connection is established, the communication parameters configured on the other PC will be displayed under the address name, as illustrated in the following figure.



Gateway_comm_para.tif

Fig. 7-53: Remote Gateway Communication Parameters

Gateway to SCP Remote Access

This remote access level allows the Gateway server on an client PC to remotely connect to the SCP server of the PC hosting a VisualMotion control. The Gateway server on the client PC is configured for local connections.

Note: SCP VM (SIS Tunneling) is the only communication channel that can configured in IndraLogic to remotely access the SCP server. RS232 PPC or TCPIP Level 2 Route can only access a local Gateway connection. Local Gateway connections access the control directly.

This following figure shows a Gateway to SCP remote access:

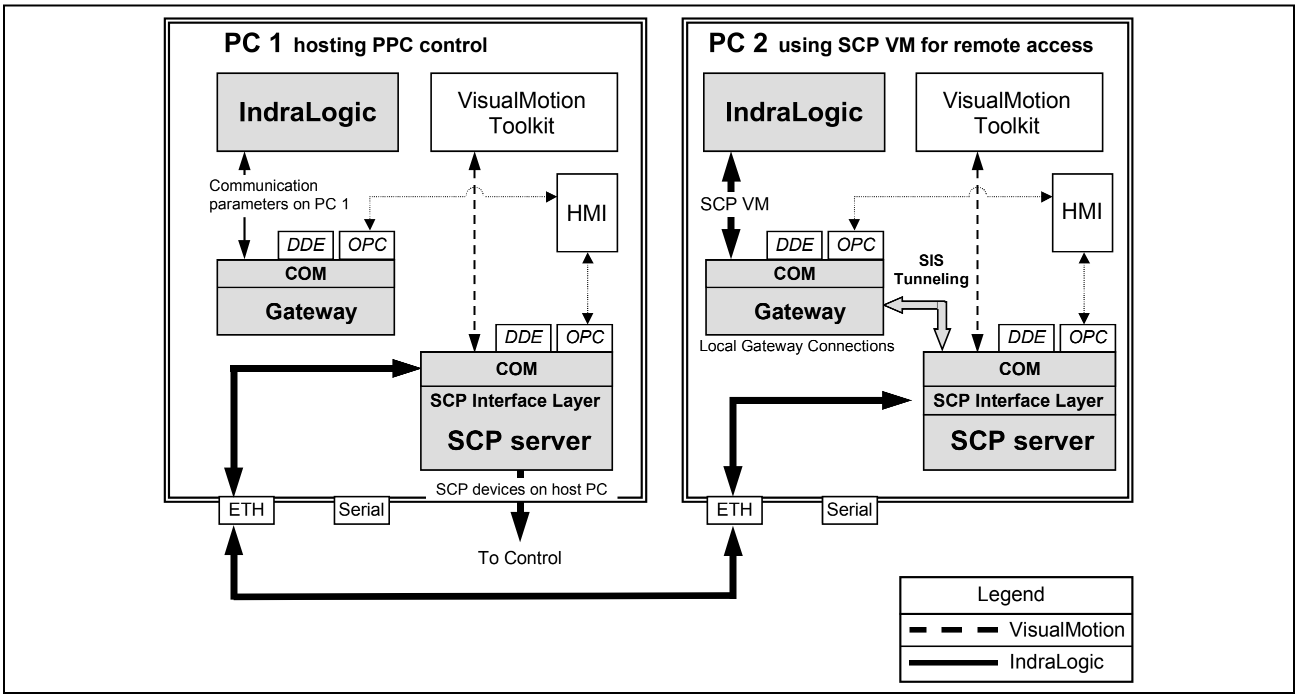


Fig. 7-54: Remote Access (Gateway to SCP Server)

Use the following steps to configure SCP Server for remote access:

1. From the Windows taskbar, select Start ⇒ Programs ⇒ Rexroth ⇒ SCP ⇒ SCP Remote Selection.
2. Select the computer name of a **Remote Host** or enter the **Remote IP Address** and click the **Apply** button. The window can then be minimized or closed.

Note: Once a connection is established, the SCP server on the host PC is automatically launched, if not currently running.

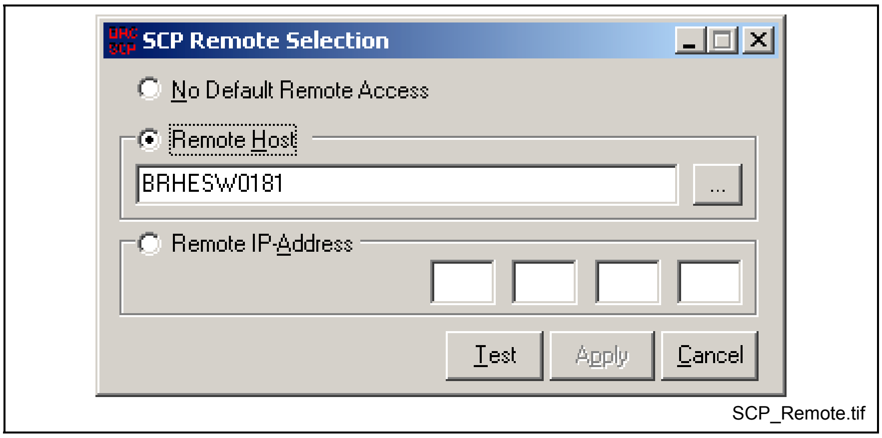


Fig. 7-55: SCP Remote Selection

3. Before launching IndraLogic, make sure that the Gateway server is not currently running on the client PC. If the Gateway server is running, right-click on the Gateway server icon in the Windows system tray and select **Exit**.

Rexroth Bosch Group

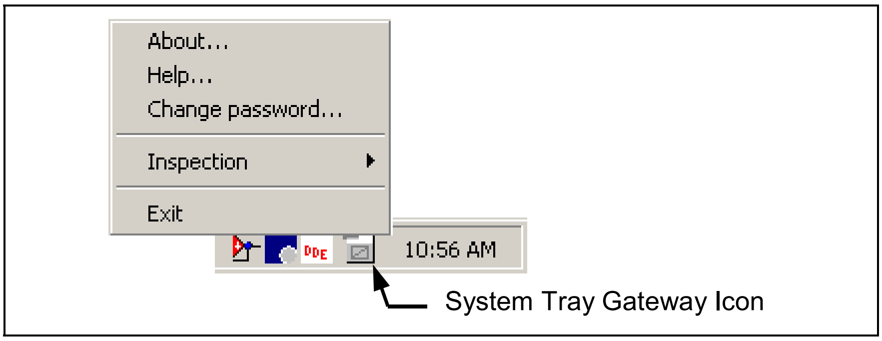


Fig. 7-56: Windows System Tray showing opened Gateway Server

4. Launch IndraLogic from VisualMotion Toolkit and create a new PLC project.

Target settings and communication parameters are passed to IndraLogic by VisualMotion. Refer to VisualMotion Variables on page 7-11 for details.

Note: The communication parameters configured on the host PC will be displayed, as illustrated in the following figure. The client PC can remove or create communication channels on the host PC.

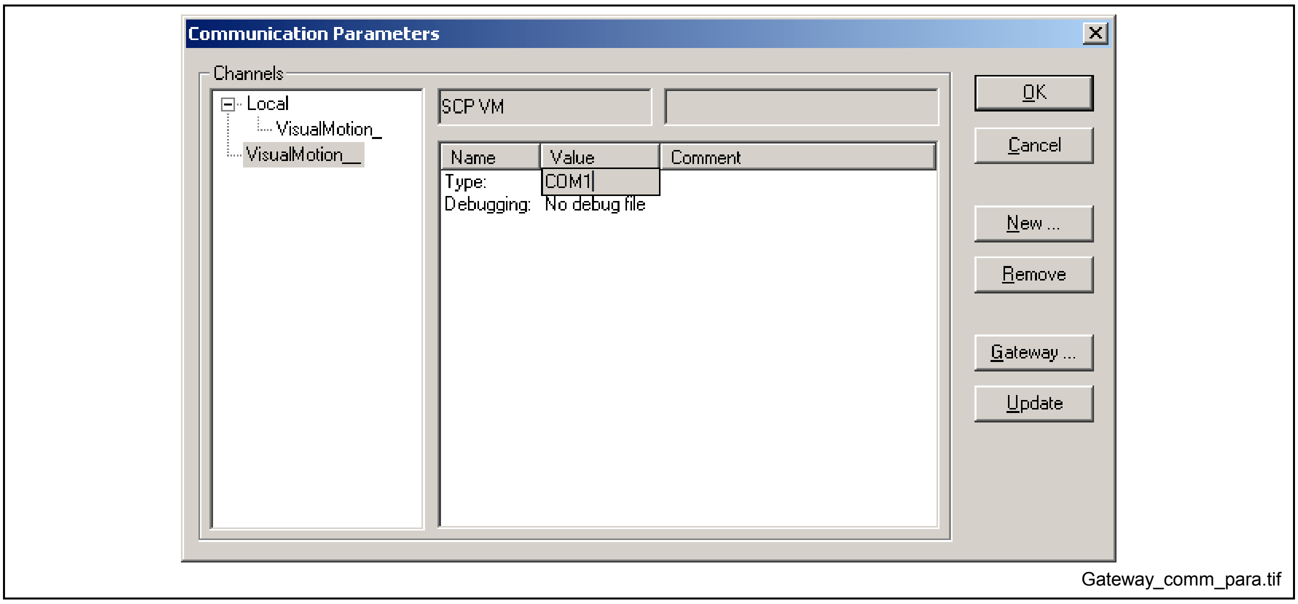


Fig. 7-57: Remote Gateway Communication Parameters

DCOM Configuration

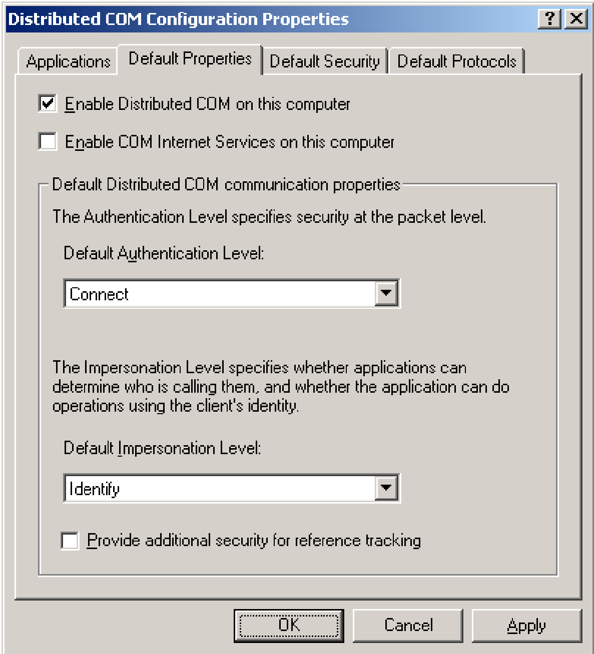
The configuration of DCOM is required in both the host and client PCs for remote access. Client PCs only need to enable DCOM, while the host PC enables DCOM and configures the security settings to allow a client PC access.

Note: This section describes the DCOM setup for Windows 2000. For Windows XP, refer to Windows XP help for details.

Note: The security settings for DCOM against unwanted access is the sole responsibility of the customer's Information Technology department.

Use the following steps to configure DCOM for remote access:

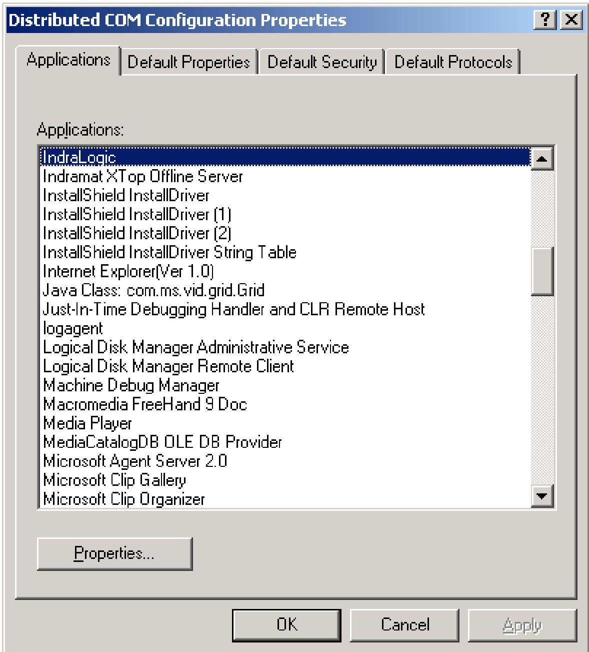
1. Select ***Start*** ⇒ ***Run*** from the Windows toolbar and enter the command, "**dcomcnfg.exe**".
2. From the *Default* tab, enable DCOM and set the *Default Authentication Level* to **Connect**.



DCOM_Def_Prop.tif

Fig. 7-58: DCOM Configuration Properties – Default Properties

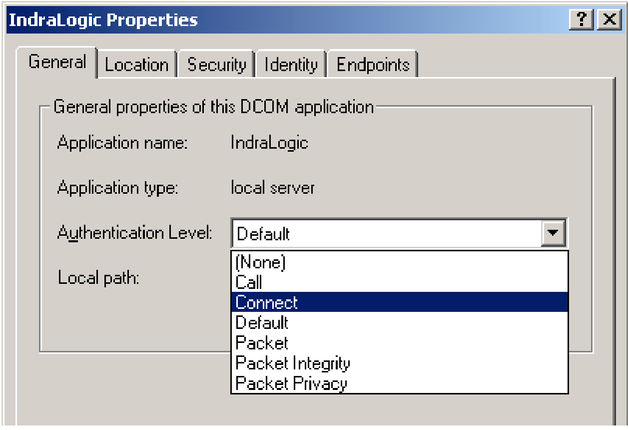
3. From the *Application* tab, scroll down and highlight **IndraLogic** and click on the **Properties…** button.



DCOM_Apps.tif

Fig. 7-59: DCOM Configuration Properties - Applications

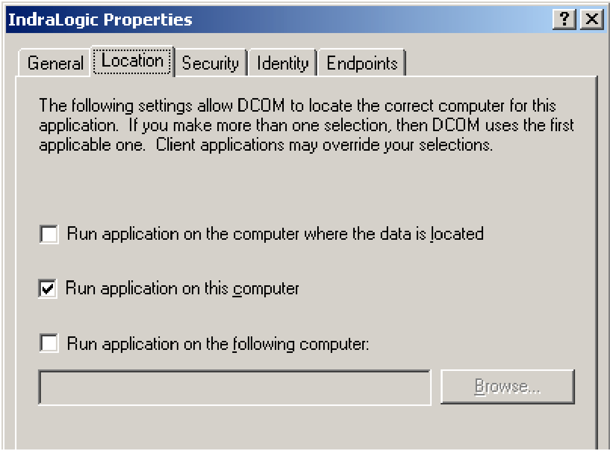
4. From the *General* tab, select **Connect** as the *Authentication Level*.



DCOM_IndraLogic_Genenral.tif

Fig. 7-60: IndraLogic Properties - General

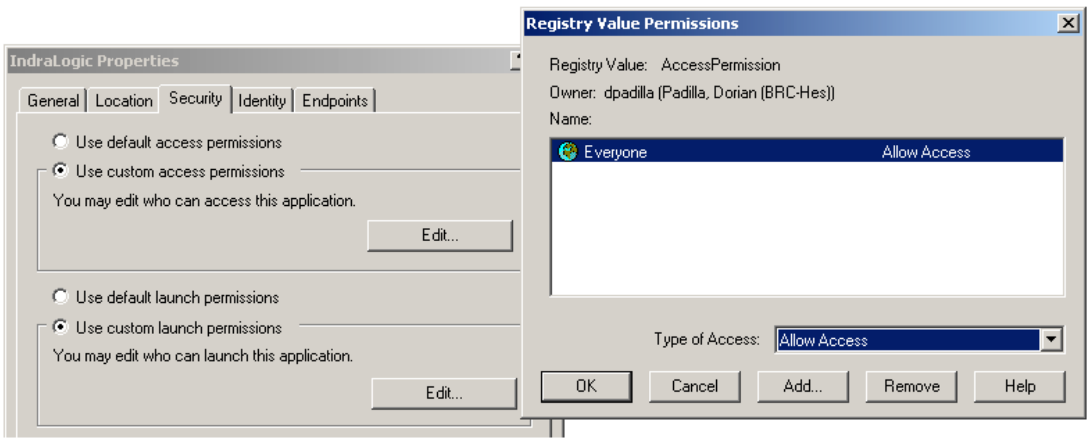
5. From the *Location* tab, set **Run application on this computer**.



DCOM_IndraLogic_Location.tif

Fig. 7-61: IndraLogic Properties - Location

6. From the *Security* tab, set the appropriate security setting for **access permissions** and **launch permissions**. Custom access permissions allows a client PC to access the host PC. Custom launch permissions is required to allow client PCs to start the SCP server for Gateway to SCP and SCP to SCP remote connections.



DCOM_IndraLogic_Security.tif

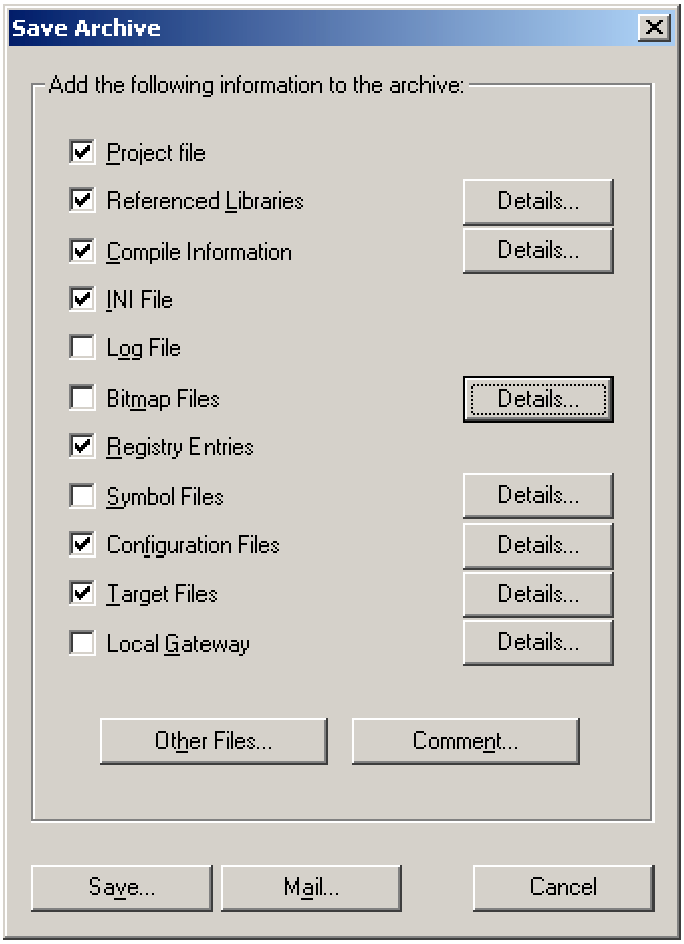
Fig. 7-62: IndraLogic Properties - Security

Managing IndraLogic Files

Although VisualMotion 11 combines motion and logic in an integrated system, the saving and archiving of motion and logic project files are handled separately by their respective programming environments. This section describes the recommended procedures for the saving and archiving of IndraLogic files. Refer to chapter 10, *Project Management Tools*, for details on how to mange VisualMotion project files.

Archive IndraLogic Files Offline

All files that are referenced by and used by the currently opened IndraLogic project can be saved as a zip file. To archive the current IndraLogic project while offline, select ***File ⇒ Save/Mail Archive…*** This opens the *Save Archive* window in IndraLogic. Refer to "Save Mail/Archive" in the *IndraLogic Help System* for more details.



SaveMail.tif

Fig. 7-63: Save Archive Window

Clicking the **Details …** button, allows for selectable entries within any given file type. The zip file can be saved to the hard drive or emailed directly from the *Save Archive* window.

Rexroth
Bosch Group

Full Online Archive of IndraLogic Project

Online archiving of an IndraLogic project is performed by reading the file contents in the PLC and saving the individual files to a folder location on the hard drive.

Note: This procedure describes the archiving of an IndraLogic project. To archive a VisualMotion project, refer to section 10.3, *Archiving Project Data*, in chapter 10.

Before archiving IndraLogic files, save any retain or persistent variables that exists in the PLC project.

Retain variables remain unchanged after an uncontrolled shutdown of the runtime system (off/on) or an 'Online' 'Reset' in IndraLogic. Persistent variables remain unchanged after a controlled shutdown of the runtime system (stop, start) or an 'Online' 'Cold reset' or a download.

Refer to "Retain variable" in the *IndraLogic Help System* for more details.

Use the following steps to perform an online archive of IndraLogic files:

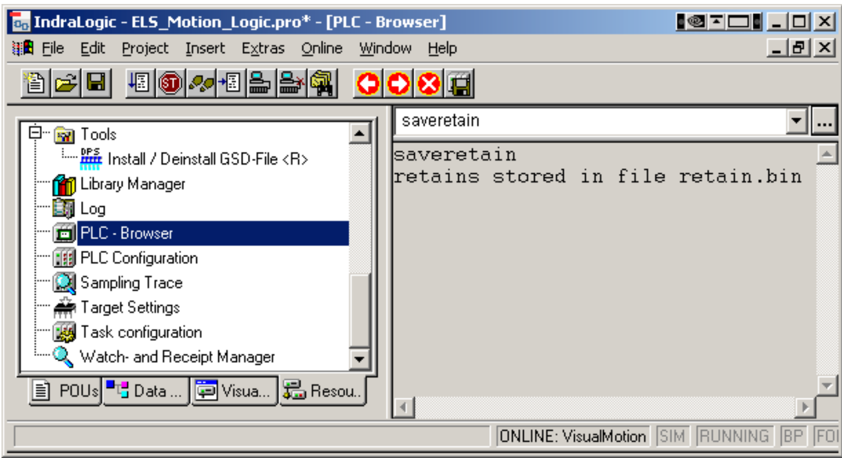
Note: Stop the PLC program before performing the save retain procedure below.

---

1. Launch IndraLogic from VisualMotion Toolkit with the relevant motion project opened in VisualMotion.
2. Go online by selecting ***Online ⇒ Login***.
3. In IndraLogic, double-click on **PLC Browser** under the *Resource* tab.

Save Retain Variables

4. Enter "saveretain" in the PLC Browser command line.



saveretain.tif

Fig. 7-64: PLC Browser – Save Retain

Note: Remanent variables can also be saved using the **SysSaveRetain** function available in the *SysLibPlcCtrl.lib* library or VisualMotion control parameter **C-0-1603, bit 4**.

Save PLC Files

5. Enter "filedir" in the PLC Browser command line. This command will list all the pertinent file on the PLC.
6. While still online with the PLC project, select ***Online ⇒ Read file from PLC***.

7. Locate a folder location on your hard drive and enter the file name of the first PLC file as it appears in the list, including the extension. **The file names are case sensitive**.

Note:	It is recommended to locate or create a folder under your current VisualMotion project folder structure.

8. Repeat this process until all the files have been saved.

Refer to the following figure as en example:

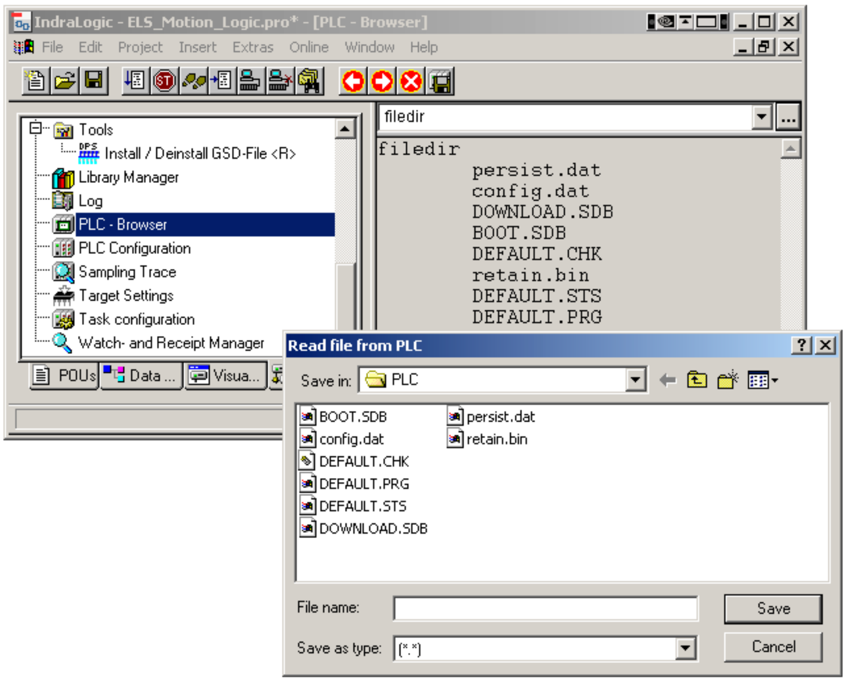


Fig. 7-65: Read file from PLC

Full Online Restore of IndraLogic Project

Online restoring of an IndraLogic project is performed by writing the previously arched IndraLogic files to the PLC.

Note:	This procedure describes the restoring of an IndraLogic project. To restore a VisualMotion project, refer to section 10.3, *Archiving Project Data*, in chapter 10.

Use the following steps to perform an online restore of IndraLogic files:

1. Go online by selecting ***Online*** ⇒ ***Login***.
2. If running, stop the PLC project by selecting ***Online*** ⇒ ***Stop*** (Shift+F8).
3. Select ***Online*** ⇒ ***Reset (original)*** to clear all files on the system.
4. Double-click on **PLC Browser** under the *Resource* tab and enter "ifsdir" in the PLC Browser command line.
5. Delete any remaining files using the "filedelete" command in the PLC Browser followed by the file name.

Note:	Remember that file names are case sensitive.

6. Using ***Online ⇒ Write file to PLC***, download all PLC project files from a previous archive.
7. Repeat step 6 until all the files have been downloaded to the PLC.
8. Rename the boot project file using a PLC Browser command as follows: "filerename DEFAULT.BAK DEFAULT.PRG"
9. Reload boot project using the PLC Browser command "reload DEFAULT.PRG".
10. Restore any existing retain data using the "restoreretain" command.
11. Start the PLC project by selecting ***Online ⇒ Run (F5)***.

Store IndraLogic Source Code on PLC

Storing an IndraLogic project source code to the PLC's memory requires that all the necessary files are saved. Use the following steps to perform an online download of the IndraLogic source code:

1. With the relevant PLC project opened, go online by selecting ***Online ⇒ Login***.
2. Select ***Project ⇒ Options…***
3. From the *Options* window, select the **Source download** category and set the *Extent* section to **All files**.
4. Select ***Online ⇒ Sourcecode download*** to download all PLC project files to the PLC.

Create a New Project from Source Code on PLC

The creation of a new PLC project from a previously downloaded source code on the PLC requires that an equivalent project in VisualMotion be created from a previously archived *.str file downloaded to the control. This procedure requires steps to be performed in both VisualMotion Toolkit and IndraLogic.

Use the following steps to created a new PLC project from the source code:

1. In VisualMotion Toolkit, create a new project from program and data on the control by selecting ***File ⇒ New…***.

Note: Step 1 can only be performed if an icon program was previously archived on the control. Refer to section 10.7, *Synchronize Project Components*, for details.

---

2. Select ***Commission ⇒ IndraLogic*** to open an empty IndraLogic project with the same name used in step 1.
3. In IndraLogic, select ***File ⇒ Open*** and select the **PLC** button.
4. In the Target Settings window, select the configuration that match the source hardware.
5. Select or create the **Communication Parameters** to connect to the control.

Note: During the upload process a message will be displayed that requires existing configuration files to be deleted in order to proceed. Press the **OK** button.

---

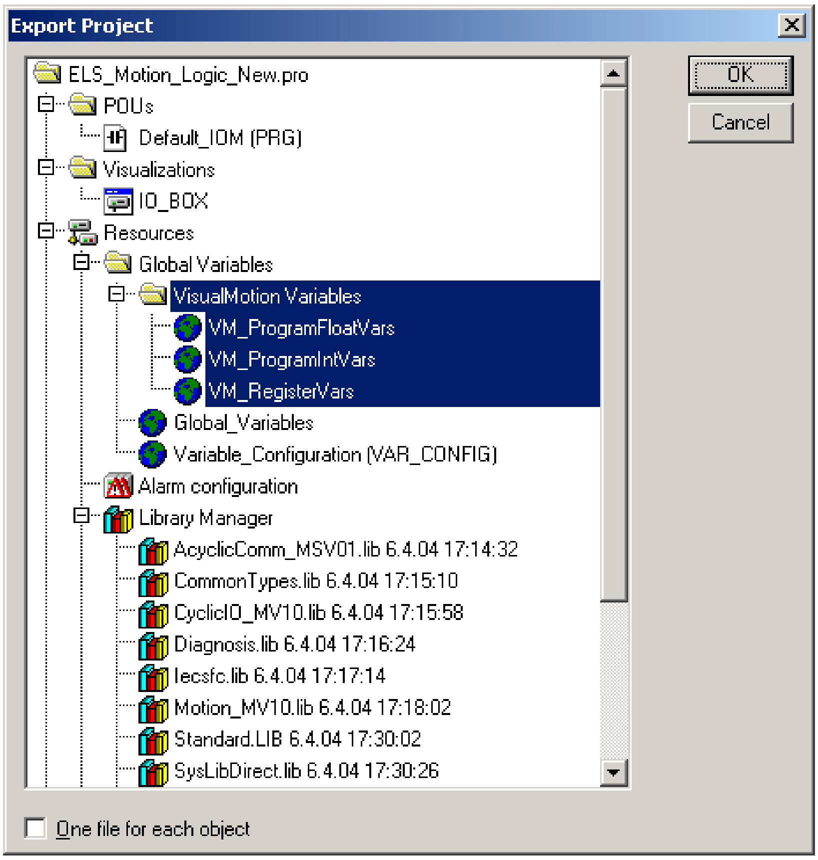
6. In the *Load project from PLC* window, select the **Yes, all** button to upload all existing configuration files on the PLC.
7. Save the IndraLogic project using the ***Save as*** command under the File menu.

8. Locate the new folder location, created in step 1, select the existing *.pro file and click on the **Save** button.

Export SysLibDirect Labels

9. To export the SysLibDirect labels to be used by VisualMotion Toolkit, select ***Project*** ⇒ ***Export***, expand the Global Variables folder and click on VisualMotion Variables.

Note: Make sure that "*One file for each object*" is not checked.



ExportVMlabels.tif

Fig. 7-66: Export VisualMotion Variables

10. Locate the new folder location created in step 1, name the file "VMLabels.exp".
11. Rebuild the IndraLogic project by selecting ***Project*** ⇒ ***Rebuild all***.
12. Go online with both VisualMotion Toolkit and IndraLogic and run the PLC project.

7.2 I/O Setup Tool

The I/O Setup Tool is used to configure all Sercos based I/O devices connected to the Sercos ring. It also serves to monitor and document all Sercos I/O devices in a VisualMotion system. This tool is available in project or service mode. Online editing of an I/O configuration is supported and recommended before configuring Sercos I/O devices.

Note: Local Reco 02 I/O modules are configured using IndraLogic. Refer to Configuring Local Reco 02 I/O Modules in IndraLogic on page 7-11 for details.

The I/O Setup Tool is launched by selecting ***Commission⇒ I/O Setup***.

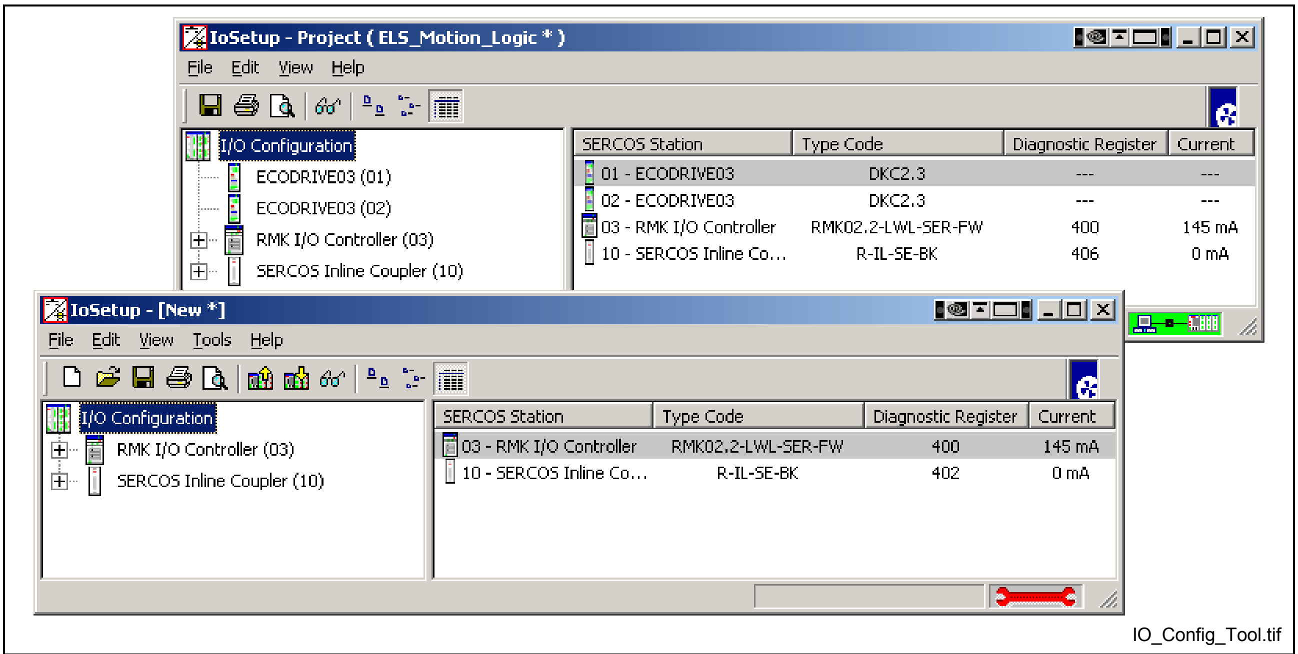


Fig. 7-67: I/O Setup Tool

VisualMotion's I/O system supports the following I/O devices:

I/O Type	Supported Programming Mode	Supported Control Firmware
Sercos Inline	Online only (Automatically configured)	GPP 11 / GMP 11
Sercos Reco 02	Online / Offline	GPP 11 / GMP 11
IndraDrive with MD1	Online / Offline	GPP 11 / GMP 11
EcoDrive 03 EcoX Bus System	Online / Offline	GPP 11 / GMP 11
Diax 03/04 Digital Drive I/O Modules	Online / Offline	GPP 11 / GMP 11

Table 7-28: VisualMotion I/O Devices

These I/O devices communicate with VisualMotion through user assigned registers. VisualMotion GPP 11 and GMP 11 firmwares provide 1,024 (16-bit) registers.

Refer to the *VisualMotion 11 Project Planning* manual for I/O device hardware specifications.

I/O Configuration in Project Mode

Rexroth digital drives and Sercos Reco 02 I/O modules can be configured for a project in offline or online mode. Sercos Inline I/O devices cannot be manually added to the I/O configuration and must be initially scanned in online mode before configuring them in offline mode.

Note: In online mode, digital drives connected to the Sercos ring will be uploaded to the I/O Setup Tool whether or not they contain I/O devices. Drive-based I/O devices cannot be detected and must be manually added to the I/O configuration.

Offline Programming

The programming of Sercos I/O devices in offline mode is supported when no communication with the control is possible.

Note: Sercos I/O devices cannot be detected in offline mode. In offline mode, any drives and/or I/O devices that have been previously configured and save to the project will appear. The user should know the types of drives and I/O devices that will be part of the project.

Use the following steps to create an I/O configuration in offline mode:

1. Select ***Commission ⇒ I/O Setup*** to open the I/O Setup Tool.
2. Right-click on I/O Configuration and select **Add Sercos Device** to add one of the following Sercos devices:
   – RMK I/O Controller
   – Diax Drive with I/O Module
   – EcoDrive with Eco-X I/O Module
   – IndraDrive with MD1 I/O Module
3. To add a remote Reco 02 I/O rack, follow step 2 and select **RMK I/O Controller** as the Sercos device. Select a Sercos address that will not be used by any other Sercos device or drive. Set the Sercos address on the RMK to match your selection.
4. To add Sercos Reco 02 I/O modules to the RMK, refer to Add I/O Module on page 7-69.
5. To add a digital drive (Diax 04/EcoDrive/IndraDrive), follow step 2 and select one of the following digital drives:
   – Diax Drive with I/O Module
   – EcoDrive with Eco-X I/O Module
   – IndraDrive with MD1 I/O Module

   Select a Sercos address that will not be used by any other Sercos device or drive. Set the Sercos address on the drive to match your selection. To add I/O modules to the drive, refer to Add I/O Module on page 7-69.
6. Save your I/O configuration to the offline project file.

Switching an I/O Configuration to Online Mode

To synchronize an I/O configuration file commissioned offline, open the project containing the I/O configuration and switch the control to online mode (***File ⇒ Online***).

VisualMotion detects any changes to the current project on the control (if downloaded previously) and informs the user of the components in the project that have changed. The user can accept the changes and download them to the control or establish a connection to the control without downloading any data by selecting the **Go online Unsynchronized** button.

Note: The system must be in parameter mode before downloading an I/O configuration to the control.

Online Programming (Editing)

In online mode, the I/O Setup Tool automatically detects and uploads digital drives, Sercos Reco 02 and Sercos Inline I/O modules.

Importing an I/O Configuration into a Project

An I/O configuration downloaded to the control can be imported into the project while in online mode, or from another project or file when in offline mode.

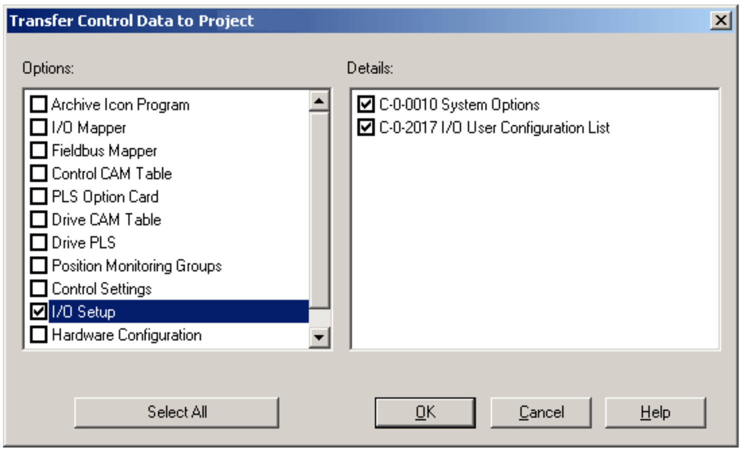
Use the following steps to import an I/O configuration from data stored on the control.

1. Start VisualMotion Toolkit and open the target project.
2. Switch VisualMotion to online mode.

Note: To import data from another offline project or file, switch VisualMotion to offline mode.

3. Select ***File ⇒ Import Project Component*** from VisualMotion Toolkit's main menu.
4. From the *Transfer Control Data to Project* window, select the **I/O Setup** checkbox. By default, both control parameters C-0-0010 and C-0-2017 are checked. To transfer only the I/O User Configuration List, uncheck control parameter C-0-0010.

Note: The system must be in parameter mode before transferring data from the control.



IOsetupImport.tif

Fig. 7-68: I/O Setup Transfer from Control

5. Press the **OK** button to transfer the I/O configuration to the current project. The I/O configuration data is now synchronized with the current project.

Note:	When importing data from another project or file in offline mode, the data is not synchronized with the project until VisualMotion is switched to online mode and the data is downloaded to the control.

I/O Configuration in Service Mode

Service mode allows the user to make modifications to an I/O configuration stored in the control's memory. Communication with the control is required before proceeding.

The following steps outline the procedure for uploading and downloading an existing I/O configuration from and to the control for modifications.

1. Open VisualMotion Toolkit in service mode.
2. Select ***Commission ⇒ I/O Setup*** to open the I/O Setup Tool.
3. Switch the control to parameter mode.
4. Upload the I/O configuration by selecting ***File ⇒ Upload Configuration*** or click the upload icon ( ).
5. Make the necessary modifications to the I/O configuration. Save the file and download the modifications to the control by clicking the download icon ( ).

Note:	I/O configurations downloaded to the control or saved to a file in service mode are not synchronized with a project's offline data. It is the responsibility of the project manager to ensure that I/O configurations modified in service mode are imported into the relevant VisualMotion project.

I/O Configuration of Sercos Inline Modules

Sercos Inline modules are only detected and uploaded to the I/O Setup Tool in online or service mode. The assignment of registers to the Sercos Inline coupler and Inline modules is automatically performed. The user cannot individually assign different register numbers to the modules.

Use the following steps to assigned registers to the Sercos Inline modules:

1. Open a VisualMotion project in online mode and launch the I/O Setup Tool.
2. Locate and expand the Sercos Inline Coupler to display the attached Inline modules.
3. Right-click on the Sercos Inline Coupler, select **Modify I/O Station** and assign a starting register for the entire Sercos Inline I/O configuration.

Note:	Make sure that the starting register number has enough available consecutive registers for the Inline modules attached to the Sercos Inline Coupler. If any new Inline modules are added to the hardware configuration, the configuration process would have to be repeat.

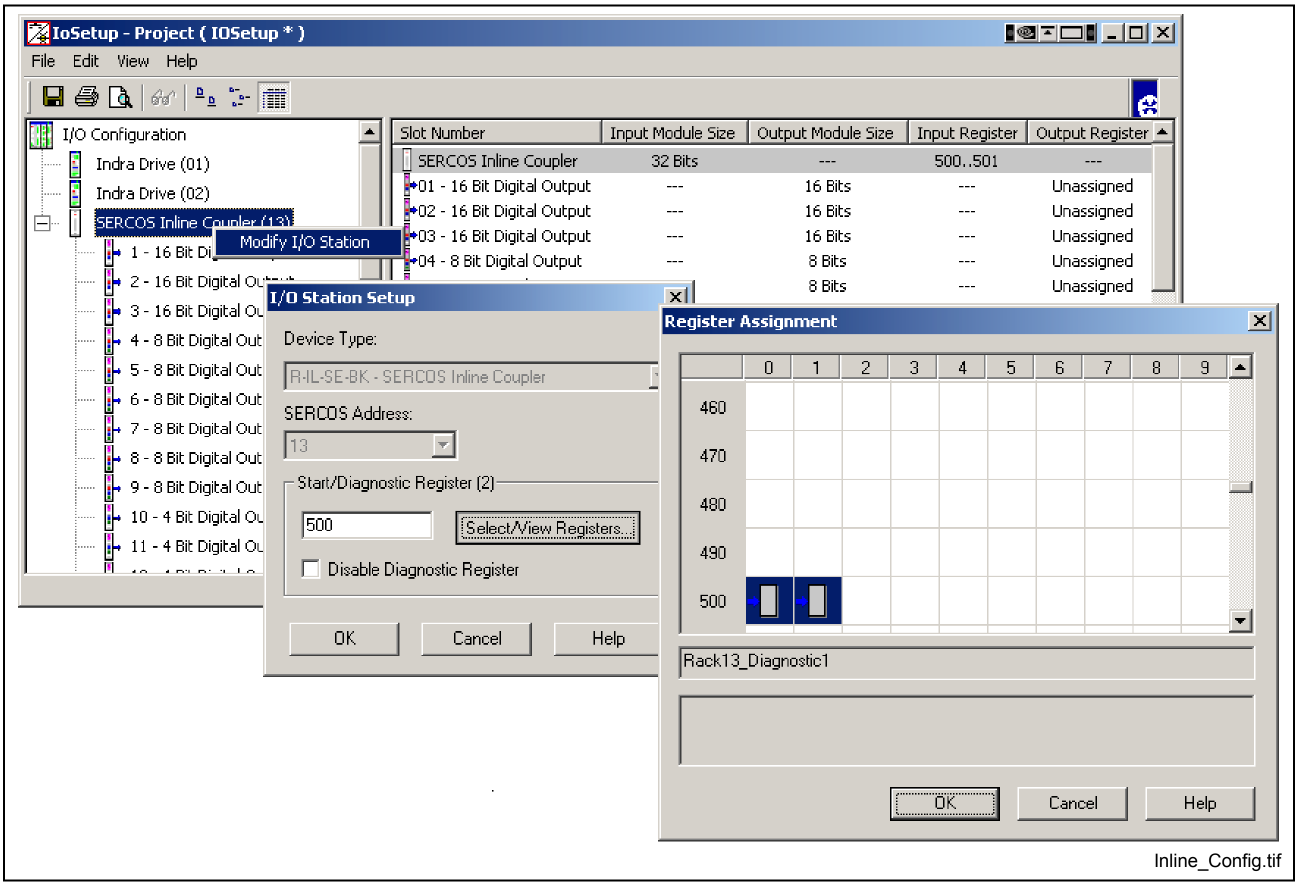


Fig. 7-69: Assigning Starting Register to Sercos Inline Coupler

4. After a starting register is assigned, clicking on the **OK** button in the *I/O Station Setup* window will allow the user to automatically assigned consecutive registers to all the attached Inline modules.

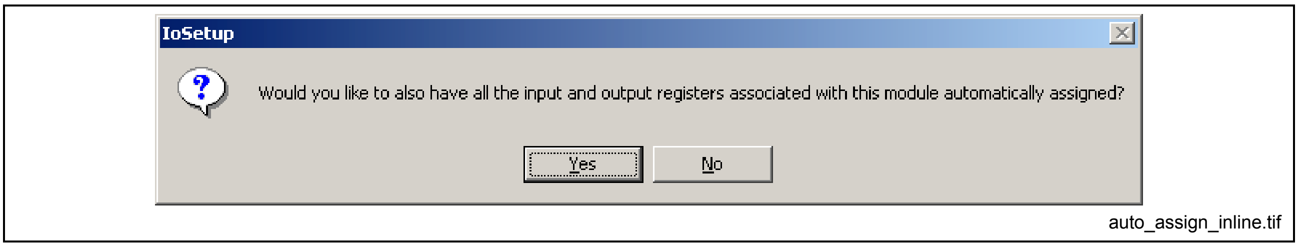


Fig. 7-70: Auto Assignment of Inline Modules

Register numbers can also be automatically assigned by selecting ***Edit ⇒ Auto Assign Registers***.

The block of register numbers that is assigned to the Inline module configuration is calculated based on the size and arrangement of each Inline module. Refer to section 5.2, *Supported I/O Devices*, in the *VisualMotion 11 Project Planning* manual for Sercos Inline specifications.

Note: If the Inline modules are not automatically assigned register numbers, they are defaulted to an inactive (disabled) state.

Enable and Disable Configured Sercos Inline Modules

To enable or disable an individual Sercos Inline module, right-click over the module icon in the tree structure and select **Modify I/O Module**. The project needs to be online and synchronized in order to perform this function.

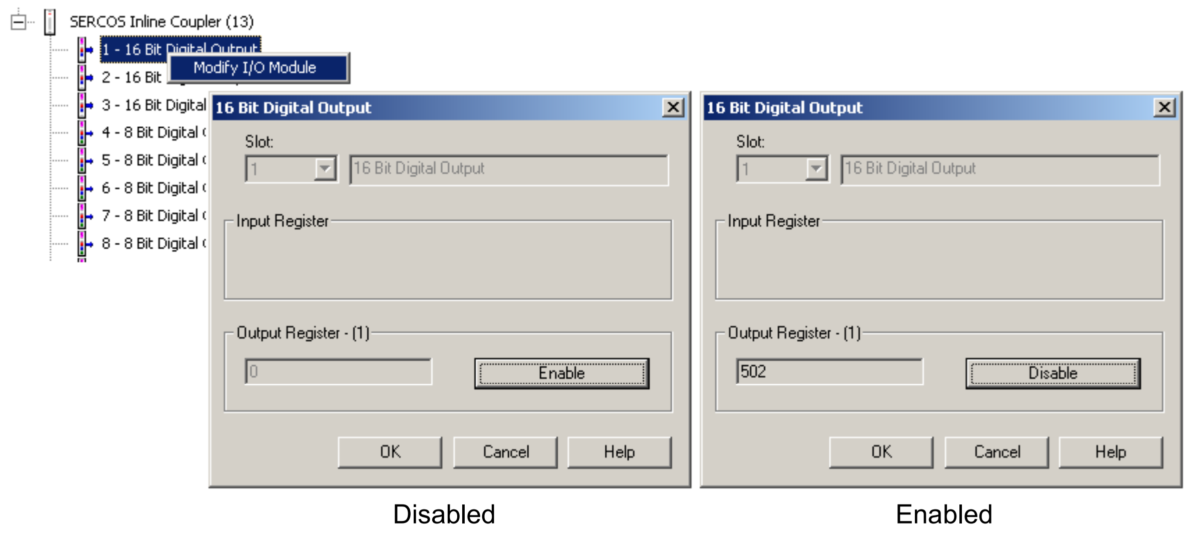


Fig. 7-71: Enable / Disable Sercos Inline Module

Determining Sercos Inline Bit Usage

The register and bit numbers assigned to each Sercos Inline module can only be determined by viewing the actual module using the I/O Setup Tool. With the I/O Setup Tool opened in online mode, expand the Sercos Inline Coupler to display the attached Sercos Inline modules. Click on the relevant module to display the configured information. The module's register bit usage can be read in the *Register Bit Range* column.

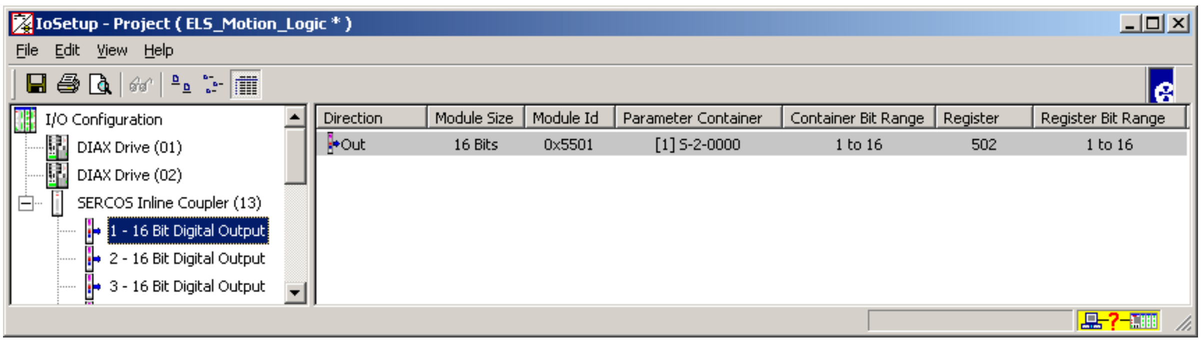


Fig. 7-72: Sercos Inline Bit Usage

The File Menu

The ***File*** menu contains standard Windows™ commands, such as Save and Print. The Upload and Download Configuration are specific I/O Setup functions and are only available in service mode.

New Ctrl+N

This selection is available in service mode and is used to create a blank I/O Configuration work area.

Open… Ctrl+O

This selection is available in service mode and is used to open existing I/O configuration (*.prm) file.

Upload Configuration

This selection is available in service mode and is used to upload the current I/O configuration from the control's memory to the I/O Setup Tool.

Download Configuration

This selection is available in service mode and is used to download any modifications made to the I/O configuration to the control's memory.

The Edit Menu

The Edit menu is used to make additions and modifications to the I/O configuration. All edit menu selections are available in both project and service mode. Selections are made available or gray-out based on the I/O device icon selected in the tree structure. The user can select either from the edit menu or by right-clicking over any I/O Device icon. The following table lists the edit selections available for each I/O device tree icon.

Tree Icon	Edit Menu Selection
I/O Configuration	Add Sercos Device
DIAX Drive (04) or Indra Drive (01) ECODRIVE03	Add I/O Module Remove Drive Modify Drive
RMK I/O Controller	Add I/O Module Remove I/O Station Modify I/O Station
Reco I/O Module 2 - 32 Inputs @ 24VDC IndraDrive and Diax I/O Module 1 - 15 Inputs/16 Outputs @ 24VDC EcoDrive I/O Module 1 - 16 Inputs/16 Outputs @ 24VDC	Remove I/O Module Modify I/O Module
Sercos Inline Coupler SERCOS Inline Coupler (13)	Modify I/O Station
Rexroth Inline I/O Module 1 - 8 Bit Digital Output 2 - 2 Bit Digital Output	Modify I/O Station

Table 7-29: Edit Menu Selections

Add Sercos Device

Select the appropriate Sercos device to add to the I/O Configuration. The following selections are available:

- RMK I/O Controller
- Diax Drive with I/O Module
- EcoDrive with Eco-X I/O Module
- IndraDrive with MD1 I/O Module

Note: The Reco I/O modules on the PPC controls rack are configured as part of the PLC Configuration in IndraLogic.

I/O Station Setup

RMK I/O Controllers, Sercos Inline couplers and digital drives use the selected Sercos address set on the front of the unit.

Note: Registers can not be assigned to digital drives for status and diagnostic purposes. Only I/O modules installed in digital drives can be assigned registers.

The **Diagnostic Register** section is used to assign a register for monitoring the status of the I/O station. Click on the ***Select/View Register*** button to add a register number to the I/O station.

The *Register Assignment* window displays the available registers in the current project. Select the desired register number and click the ***OK*** button. The number of required registers will be automatically highlighted.

Note: The assignment of I/O registers to Sercos devices and I/O modules can be performed when adding the devices or by selecting ***Edit*** ⇒ ***Auto-Assign Registers*** after all the devices are configured.

Add I/O Module

Select the type of I/O module to configure for a RMK I/O Controller or digital drive. The **Add I/O Module** selection is only available when a RMK I/O Controller or digital drive is selected in the tree structure. The number of assigned registers in a VisualMotion I/O system is based on the configured I/O modules.

Note: Sercos Inline I/O modules cannot be added to the I/O configuration. The Inline modules are uploaded only when the I/O Setup Tool is launched in online mode.

RMK I/O Controller I/O Modules

The following I/O modules are available for the RMK I/O Controllers.

Type (radio button)	Selection	Description	Registers Used
Analog	RMC02.2-2E-1A	Reco 02 Analog Module (2 Inputs / 1 output)	4
Digital Input	RME02.2-16-DC024	Reco 02 Digital 16 Input Module	1
	RME02.2-16-AC115	Reco 02 Digital 16 Input Module	1
	RME02.2-32-DC024	Reco 02 Digital 32 Input Module	2
Digital Output	RMA02.2-16-DC024-200	Reco 02 Digital 16 Output Module	1
	RMA02.2-16-DC024-100	Reco 02 Digital 16 Output Module	1
	RMA02.2-32-DC024-050	Reco 02 Digital 32 Output Module	2
	RMA02.2-16-RE230-200	Reco 02 Digital 16 Output Module	1
	RMA02.2-16-AC230-200	Reco 02 Digital 16 Output Module	1

Table 7-30: Local Reco and RMK I/O Controller I/O Modules

Diax03/04 Digital Drive I/O Modules

The following I/O modules are available for Diax03/04 digital drives.

Type (radio button)	Selection	Description	Registers Used
Analog Input	DAE02.1	Analog Input Module (2 inputs – 14 bit @ 10V)	2
	DRF02.1	Analog Input Module (2 inputs – 12 bit @ 10V)	2
Digital (Input/Output)	DEA4.1	Digital Drive 15 Inputs/16 Outputs Module	2
	DEA5.1	Digital Drive 15 Inputs/16 Outputs Module	2
	DEA6.1	Digital Drive 15 Inputs/16 Outputs Module	2
	DEA8.1	Digital Drive 32 Inputs/24 Outputs Module	4
	DEA9.1	Digital Drive 32 Inputs/24 Outputs Module	4
	DEA10.1	Digital Drive 32 Inputs/24 Outputs Module	4

Table 7-31: Diax03/04 Digital Drive I/O Modules

IndraDrive with MD1 I/O Module

The MD1 I/O module is only supported with the IndraDrive advanced interface. It provides 12 inputs and 8 outputs at 24VDC.

EcoDrive 03 Eco-X I/O Module

The EMD module is an expansion I/O module that interfaces with the DKC22.3 using the EcoX bus system. EcoX communication is designed for a DKC22.3 digital drive using SGP20 firmware. Refer to the *VisualMotion 11 Project Planning* manual for details.

Remove I/O Station (Remove Drive)

Select the I/O station (Local Reco, RMK I/O Controller or Digital Drive) to be removed from the I/O configuration. The *Remove I/O Station or Drive* selection is only available when a RMK I/O Controller or a digital drive is selected in the tree structure.

Note: Any I/O modules configured under the selected I/O station will also be removed.

Modify I/O Station (Modify Drive)

Select the I/O station (Digital Drive, RMK I/O Controller or Sercos Inline Coupler) to modify. The **Modify I/O Station** or **Drive** selection is only available when a digital drive, RMK I/O Controller or Sercos Inline Coupler is selected in the tree structure. Only the address and register assignment of an I/O station can be modified. Sercos Inline I/O Couplers have an additional checkbox for disabling the diagnostic register.

Remove I/O Module

Select the I/O module to be removed from an I/O station or digital drive. The **Remove I/O Module** selection is only available when a Reco or digital drive I/O module is selected in the tree structure.

Modify I/O Module

Select the I/O module to be modified in the I/O station, digital drive, or Sercos Inline Coupler. The **Modify I/O Module** selection is only available when a Reco, digital drive, or Rexroth Inline module is selected in the tree structure. Only the I/O module's slot (Reco only) and register assignment can be modified.

Auto-Assign Registers

This menu selection allows the user to add default starting registers to all configured I/O devices. Used registers are displayed with an icon within each register number when selecting *View ⇒ Display Register Usage*.

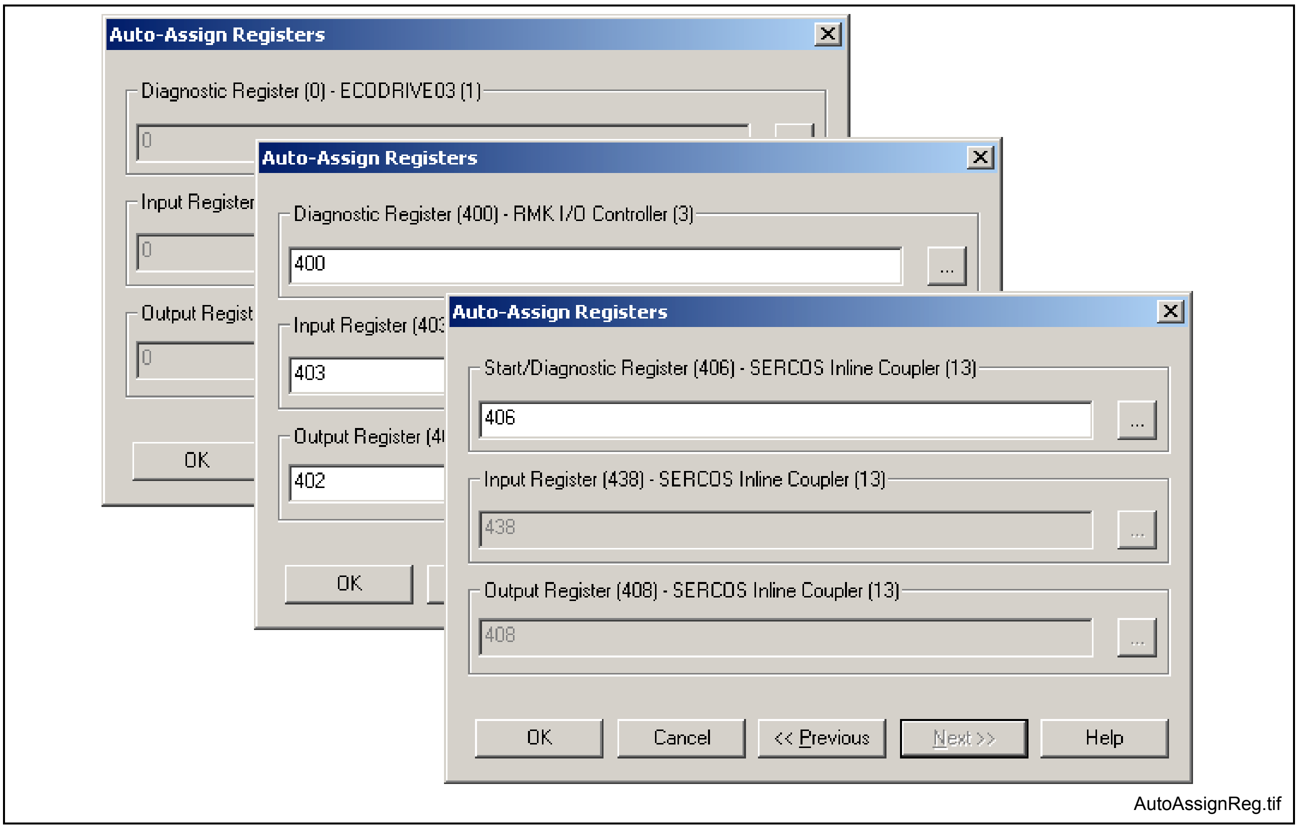


Fig. 7-73: Auto Assign Registers to I/O

Reco I/O Error Reaction

This menu selection allows the user to select one of the following error reactions for all Reco I/O modules in a system. The error codes displayed on the control dependent on the set Reco error reaction.

Error Reaction	Description
Ignore Errors	This setting will not stop motion to the system, but indicate a condition in the I/O Setup Tool. No errors are displayed on the control.
Generate a Warning	This setting will not stop motion, but error diagnostic "**215 Reco I/O Failure, see ext. diag**" is displayed on the control.
Generate Fatal Error (Default Setting)	If an error is detected on any Reco I/O module, all motion to the system is stopped and error code diagnostic "**544 Reco I/O Failure, see ext. diag**" is displayed. This code remains on the control until the error is corrected and the system is restarted.

Table 7-32: Reco I/O Error Reaction Settings

Reco 02 Error Detection

I/O Configurations consists of a Sercos device (RMK I/O Controller or digital drive) and individual I/O modules. Two diagnostic registers are assigned to each RMK I/O Controller for use as a 32-bit status word for monitoring the status of each configured module. Reco 02 I/O status words are scanned every Sercos cycle scan time (2 or 4 ms).

The following figure shows the bit structure for reporting errors and indicating the module at fault. The lower 16-bits specify the Reco rack slot number in which the I/O module error exists. Bit 0 refers to Reco rack slot 0 and bit 15 refers Reco rack slot 15. Refer to **Slot Addressing of the RMB02.2 Racks** in the *VisualMotion 11 Project Planning* manual for details.

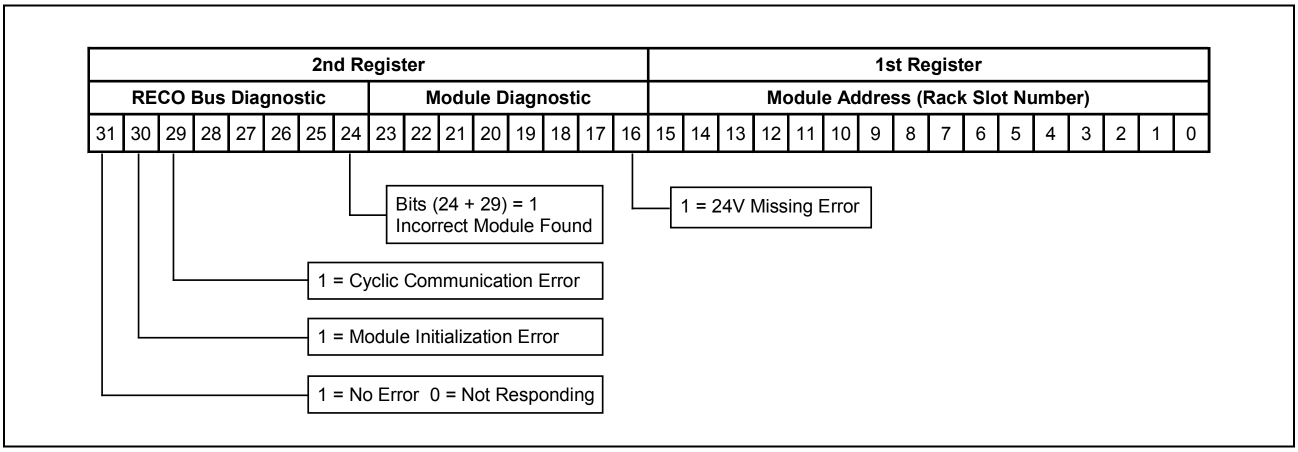


Fig. 7-74: 32-bit Reco Status Word

The following table describes all supported errors for Reco 02 modules. Reco 02 I/O errors, other than the *24V Missing* error, can only be cleared/reset by cycling power to the control after correcting the cause of the problem.

Reco 02 Error	Description
No Error	All configured Reco modules were found and working properly.
Not Responding	This error is normally generated at the Reco controller level.
24V Missing	Output module is missing 24V supply. Verify that all output module connectors are properly connected to 24V.
Incorrect Module Found	The I/O module identified does not match the original I/O Configuration.
Cyclic Communication	A cyclic communication error occurred in the I/O module identified in the *Module Address*.
Module Initialization	An error occurred during the initialization of the I/O module. Contact Bosch Rexroth Service.

Table 7-33: Reco 02 Error Description

When a Reco error is issued, the Sercos device and I/O module name are displayed as bold text in the left window and as red text in the right window.

Note: Digital drive I/O module errors are not monitored by VisualMotion. I/O module errors are reported directly to the digital drive. Refer to the appropriate *Digital Drive* documentation for parallel I/O module errors.

The View Menu

The View menu is used to increase the overall size of the work area by allowing the user to toggle the Toolbar and Status area on or off. A check to the right of the name indicates that the item is active.

Toolbar

Selecting ***View*** ⇒ ***Toolbar*** turns the icon toolbar on or off. The icon toolbar contains standard Windows™ commands. The icons available vary based on the mode of communication, as illustrated in Fig. 7-75.

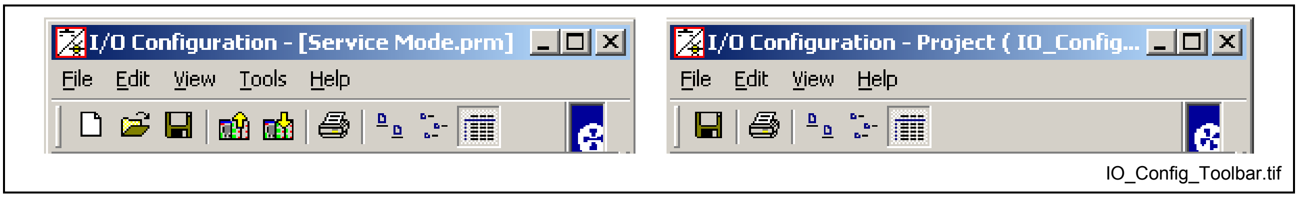


Fig. 7-75: I/O Setup Toolbar

Status Bar

Selecting ***View*** ⇒ ***Status Bar*** turns the status indicator at the bottom of the window on or off. The status bar contains icon description text to the left, downloading and uploading percentage (when in service mode) and communication mode to the right.

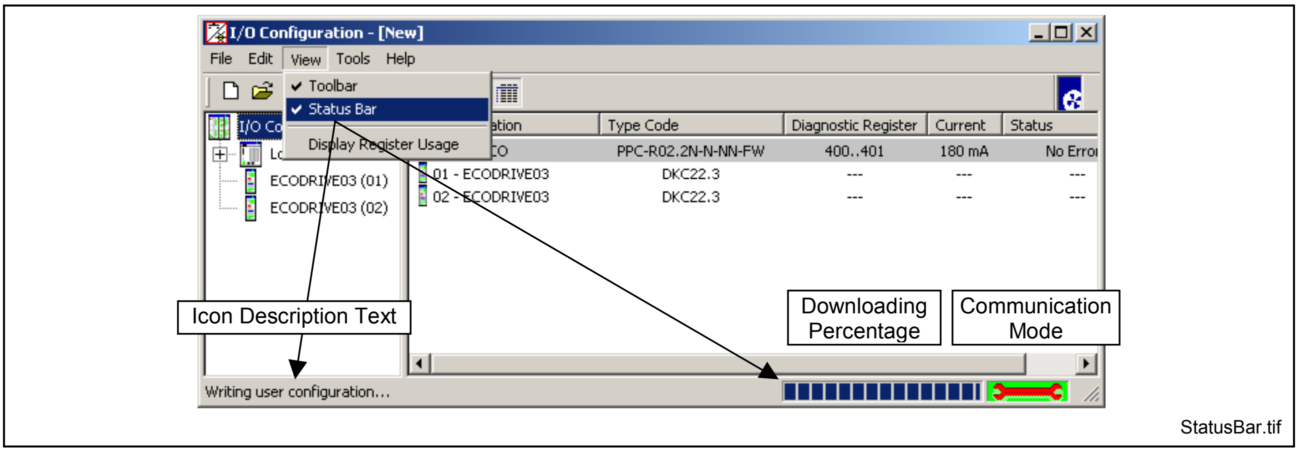
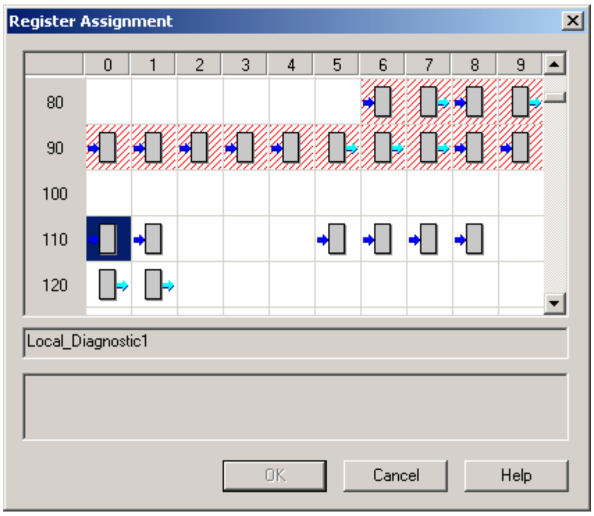


Fig. 7-76: I/O Configuration Status Bar

Display Register Usage

Each register assigned to an I/O device can have a maximum of 16 input or output points. Some registers are reserved for system functions, while others are recommended as defaults.

Selecting ***View*** ⇒ ***Display Register Usage*** opens a window indicating what system registers contain labels. Any registers that have been assigned to any Reco, Drive I/O or Sercos Inline will be displayed as input or output. System reserved registers display red crosshatches behind each I/O icon. Label names appear in the field just below the icons for the selected register, as shown in Fig. 7-77. Register labels can be assigned and edited by selecting ***Edit*** ⇒ ***VM Data*** from VisualMotion Toolkit's main menu.



Register_usage.tif

Fig. 7-77: I/O configuration Register Usage

Monitor Status F7

This menu selection enables the monitoring of I/O status. This feature can be also be enabled using the eye-glass icon .

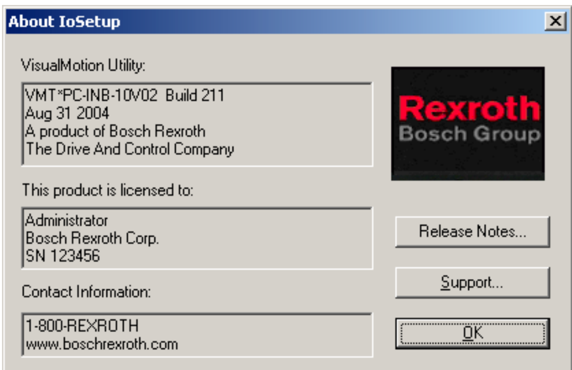
The Tools Menu

The ***Tools*** menu selection is only available for service mode. In project mode, the communication settings are handled by the project. Selecting Tools opens the *Control Selection* window. Refer to section 13.10 in volume 2 of the *VisualMotion Functional Description* for details.

The Help Menu

Selecting ***Help*** ⇒ ***Contents F1*** opens the I/O Configuration portion in the VisualMotion help system.

Selecting ***Help*** ⇒ ***About…*** displays VisualMotion Toolkit version, licensed and contact information. Clicking the **Release Notes…** button opens a PDF file of the current firmware version release notes. For a listing of Bosch Rexroth Service and Support locations throughout the World, click the **Support** button.



IO_Help.tif

Fig. 7-78: About I/O Configuration Window

7.3 Sercos Drive Telegram Utility

This utility provides the user with a convenient and comprehensive interface for viewing and configuring the AT (Drive Telegram) and MDT (Master Data Telegram).

The AT and MDT are comprised of parameter numbers (IDNs) from various drive parameters. Some IDNs displayed in the AT and MDT are automatically configured based on the control's primary and secondary modes of operation. User configurable IDNs can be added to each telegram. User configurable IDNs and data are placed within axis parameters. The user configurable IDNs are labeled as *Optional Feedback ID* for the AT and *Optional Command ID* for the MDT.

Refer to *Axis Parameters used by AT and MDT* on page *7-80* for details.

AT (Drive Telegram)

The AT is a block of drive data transmitted cyclically to the control every Sercos cycle. The configurable data block of the AT contains a 2-byte drive status word, variable length drive dependent parameters that are automatically configured based on the mode of operation and a user configurable data block.

Note: The AT's configurable data block size limitation is dependent upon the connected drive and drive firmware. Refer to the relevant *Digital Drive* documentation for the actual size limitation of the AT.

VisualMotion controls allow up to 5 variable length drive dependent parameters to be added to the user configurable data block of the AT.

EcoDrive 03 Drives:

DKC02.3 drives using SGP/SMT firmware have a 16-byte (8 word) limit for the configurable data block. If the length of the data that the user adds exceeds this limit, the cyclic multiplex channel is automatically activated. Refer to *Multiplex (MUX) Channel (DKC 2.3 only)* on page *7-80* for details.

Diax 04 Drives:

Diax 04 drives using SSE/ELS firmware have a 24-byte (12 word) limit and do not have a multiplex channel. Since Diax 04 drives have a much larger configurable data block for the AT, adding 5 drive parameters will not exceed the 24-byte limit.

IndraDrive Drives:

IndraDrive drives using MPH02 firmware have a 32-byte (16 word) limit and do not have a multiplex channel. Since IndraDrive drives have a much larger configurable data block for the AT, adding 5 drive parameters will not exceed the 32-byte limit.

Drive Status Word

The Drive Status Word (S-0-0135) is part of the AT's configurable data block and contains drive status information transmitted to the control. The following table describes the bits of the drive status word.

Bit	Description
2 – 0	Control Information for Sercos service channel
5	Bit change command
7 – 6	Real-time status bits 1 and 2

Bit	Description
9 – 8	actual type of operation `00:` primary mode of operation `01:` secondary mode of operation
11	Bit change class 3 diagnostic
12	Bit change class 2 diagnostic
13	Drive lock, error in class 1 diagnostic
15 – 14	Ready to operate: `00:` Drive not ready for power to be switched on because internal checks not positively connected `10:` Control and power supply ready for operation, torque free `01:` Ready to switch on power `11:` In Operation, under torque

Table 7-34: S-0-0135 Drive Status Word

Configurable AT Data Block

The contents of the AT telegram are automatically set by the control based on axis configuration options and the settings in the Sercos drive telegram utility.

MDT (Master Data Telegram)

The MDT is a block of control data transmitted cyclically to the drives every Sercos cycle. The configurable data block of the MDT contains a 2-byte master control word, variable length drive dependent parameters that are automatically configured based on the mode of operation and a user configurable data block.

Note: The MDT's configurable data block size limitation is dependent upon the connected drive and drive firmware. Refer to the relevant *Digital Drive* documentation for the actual size limitation of the MDT.

VisualMotion controls allow up to 5 variable length drive dependent parameters to be added to the user configurable data block of the MDT.

EcoDrive 03 Drives:

DKC02.3 drives using SGP/SMT firmware have a 16-byte (8 word) limit for the configurable data block. If the length of the data that the user adds exceeds this limit, the cyclic multiplex channel is automatically activated. Refer to *Multiplex (MUX) Channel (DKC 2.3 only)* on page *7-80* for details.

Diax 04 Drives:

Diax 04 drives using SSE/ELS firmware have a 32-byte (16 word) limit and do not have a multiplex channel. Since Diax 04 drives have a much larger configurable data block for the MDT, adding 5 drive parameters will not exceed the 32-byte limit.

IndraDrive Drives:

IndraDrive drives using MPH02 firmware have a 32-byte (16 word) limit and do not have a multiplex channel. Since IndraDrive drives have a much larger configurable data block for the MDT, adding 5 drive parameters will not exceed the 32-byte limit.

Master Control Word

The Master Control Word (S-0-0134) is part of the MDT's configurable data block and contains control data transmitted to the drives. The following table describes the bits in the master control word.

Bit	Description
5 – 0	Control Information for Service Channel
7 – 6	Real-time Control Bits 1 and 2
9 – 8	Command Mode `00`: primary mode `01`: auxiliary modes
10	IPOSYNC: interpolator pulse, toggles if new command values to be transmitted
13	Drive HALT, 1⇒ 0 transition, standstill of drive while maintaining max. acceleration(S-0-0138 (only possible if bits 14 and 15 set to 1)
14	Drive ENABLE, 1⇒ 0 transition, torque off without delay (independent of bit 15 or 13)
15	Drive ON, 1⇒ 0 transition, best possible standstill (only possible if bit 14 set to 1)

Table 7-35: S-0-0134 Master Control Word

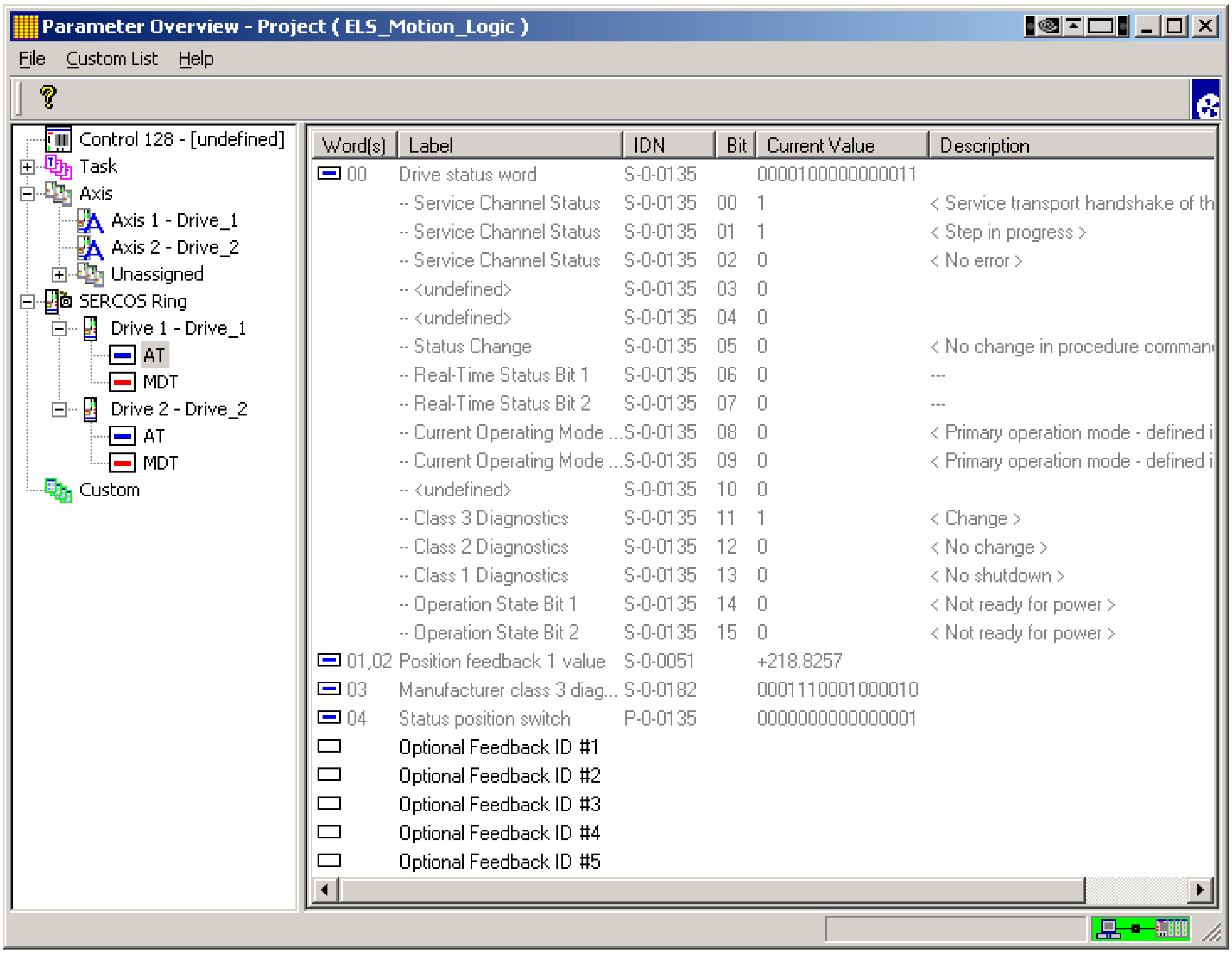
Configurable MDT Data Block

The contents of the MDT are automatically set by the control based on axis configuration options and the settings in the Sercos drive telegram utility.

Rexroth
Bosch Group

Displaying the Contents of the AT and MDT

To launch the Sercos Drive Telegram Utility, select ***Data ⇒ Parameters ⇒ Edit…*** Using the parameter tree structure, expand the Sercos Ring to display the active drives. Expanding each drive icon displays the AT and MDT selections. Parameters that are operational mode specific are displayed in gray and user configurable data blocks in black.



AT_MDT.tif

Fig. 7-79: Sercos Drive AT and MDT

Configuring the AT and MDT

Use the following steps to add parameter IDNs to the AT or MDT.

Note: The control must be switched to parameter mode to add or modify user configurable IDNs.

1. Open either the AT or MDT to display the contents of the telegram.
2. Double-click a user configurable IDN or right-click and select *Edit* to open the "List of configurable data" window of the selected telegram.

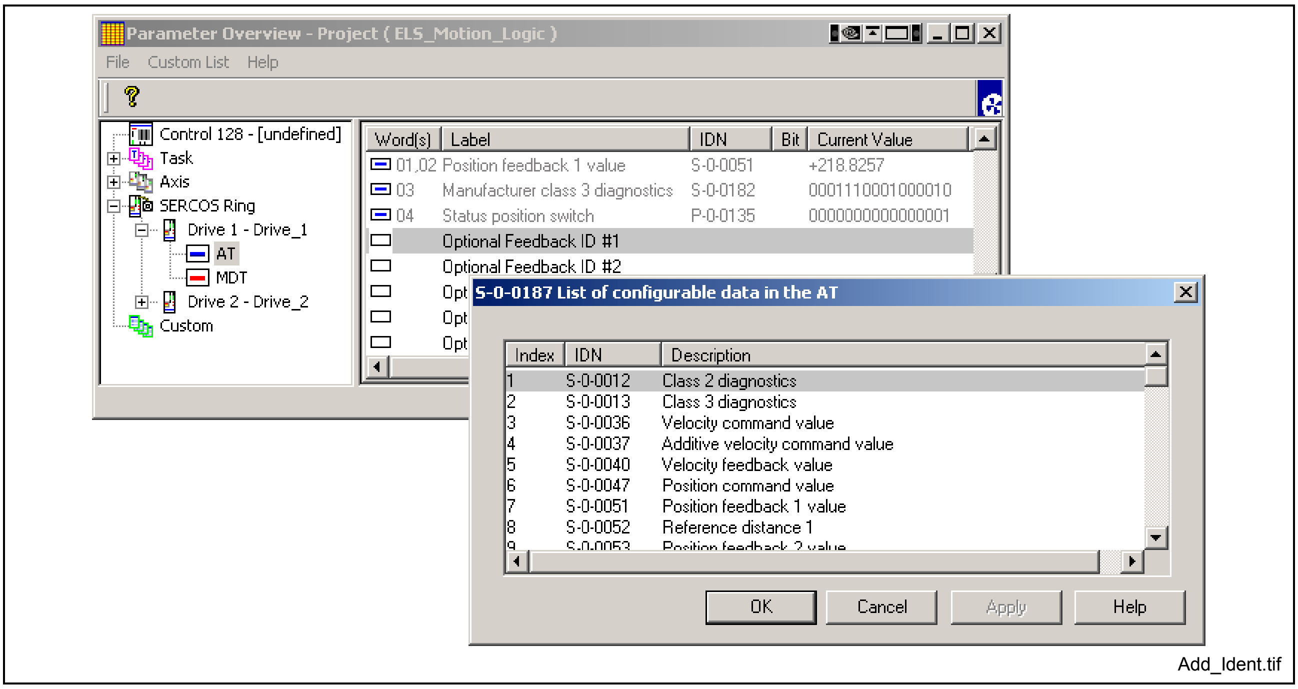


Fig. 7-80: Adding a Parameter IDN

Note: The list of configurable data for each telegram is provided in the following drive parameters:

- S-0-0187 for the AT
- S-0-0188 for the MDT

3. Select the desired parameter from the list and click the **OK** button. A message will appear notifying the user that changes take effect in phase 4.
4. Select the next user configurable IDN and repeat the process.
5. Switch the control to phase 4, right-click any parameter IDN in the telegram, select *Refresh* of press the *F5* key to display the configured telegram.

Fig. 7-81 displays the configured telegram with user configured IDNs in black.

Note: To remove a user configured IDN, switch the control to parameter mode, right-click over the IDN and select *Remove*. Refresh the window to display the optional default IDN.

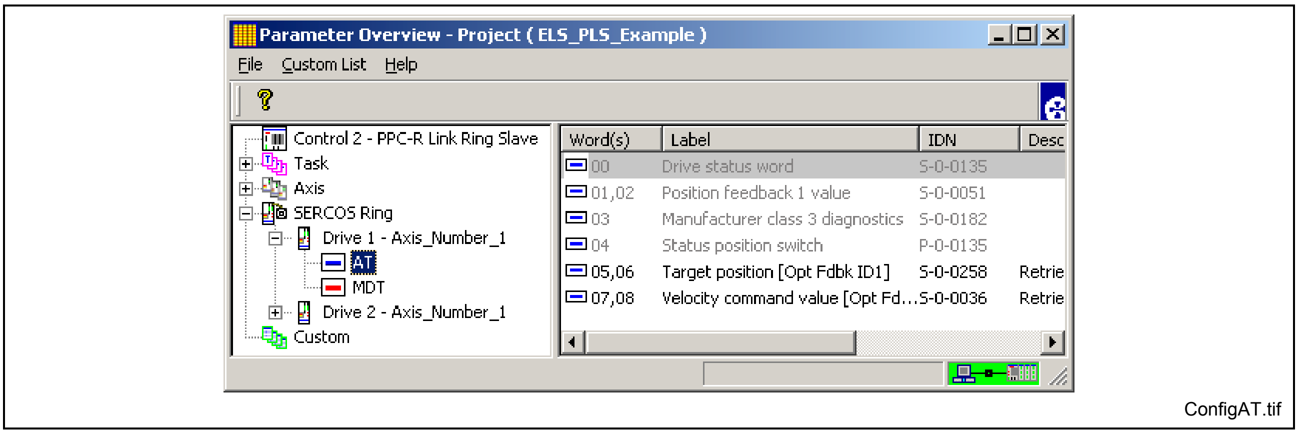


Fig. 7-81: Configured Telegram

Rexroth Bosch Group

Axis Parameters used by AT and MDT

A user configured parameter IDN is placed in an axis parameter. The data for the configured parameter IDN is accessed from a different axis parameter. The Sercos telegram displays the axis parameter from where the data is retrieved under the *Description* column.

User Configured AT

The user can place up to 5 additional parameters into the user configurable data block of the AT. The corresponding axis parameters containing the parameter's IDN and data are listed in the following table.

User Configured AT (Drive Telegram) Data Location		
User Configured Data	**Parameter Ident**	**Data Accessed**
Optional Feedback ID #1	A-0-0185	A-0-0195
Optional Feedback ID #2	A-0-0186	A-0-0196
Optional Feedback ID #3	A-0-0187	A-0-0197
Optional Feedback ID #4	A-0-0188	A-0-0198
Optional Feedback ID #5	A-0-0189	A-0-0199

Table 7-36: User Configured AT Data

User Configured MDT

The user can place up to 5 additional parameters into the user configurable data block of the MDT. The corresponding axis parameters containing the parameter's IDN and data are listed in the following table.

User Configured MDT (Drive Telegram) Data Location		
User Configured Data	**Parameter Ident**	**Data Accessed**
Optional Command ID #1	A-0-0180	A-0-0190
Optional Command ID #2	A-0-0181	A-0-0191
Optional Command ID #3	A-0-0182	A-0-0192
Optional Command ID #4	A-0-0183	A-0-0193
Optional Command ID #5	A-0-0184	A-0-0194

Table 7-37: User Configured AT Data

Multiplex (MUX) Channel (DKC 2.3 only)

The MUX channel is a feature of DKC02.3 drive firmware and was designed to overcome the limited 16-byte data container length of the AT and MDT. The MUX channel is either enabled by the selection of the Drive PLS Fast Write Feature or enabled automatically when the AT or MDT exceed the 16-byte limit. Once enabled, the MUX channel reconstructs the AT and MDT to place predetermined parameter IDNs into the user configurable portion of the telegram and the remaining IDNs into the MUX channel.

The MUX IDNs are placed in a circular queue. Every Sercos cycle, the index for each queue is incremented and one IDN and corresponding data from the MDT MUX queue is written to the drive and one IDN from the AT MUX queue is requested from the drive. The following AT read cycle updates the data location for that AT MUX IDN. The number of IDNs in the circular queue will determine how many Sercos cycles occur before the data is updated.

The following parameters and registers are associated with the MUX channel:

Parameter or Register	Description
A-0-0004, Bit 8 = 1	enables Drive PLS Fast Write
A-0-0200 A-0-0201	read-only lists of all IDNs that can be placed into the MUX for AT and MDT
A-0-0202 A-0-0203	holds the circular MUX queues for the AT and MDT
S-0-0360 S-0-0364	holds the data for the MUX channel parameter for the AT and MDT
S-0-0362 S-0-0366	holds the list of indexes for MUX channel list parameters
S-0-0368, Bits 0-5 S-0-0368, Bits 8-13	holds the index into A-0-0202 and A-0-0203 for the AT and MDT
S-0-0370 S-0-0371	holds copies for A-0-0200 and A-0-0201 for the AT and MDT
Registers 31-38, 309-340 and 441-464	bit 1 shows the status of the MUX channel enabled for axes 1-8, 9-40 and 41-64

Table 7-38: Multiplexing Parameters

Telegram Options

The following telegram options are available when right-clicking on any user configurable data:

The *Edit* and *Remove* selections are only visible when right-clicking on user configurable data while in parameter mode. The telegram option selections are only available in parameter mode and require a system P2 ⇒ P4 transition for the modifications to take effect.

Edit

This selection is used to open the telegram's list of configurable data for adding an allowable parameter IDN to the telegram's user configurable data block.

Remove

This selection is used to remove a user configured parameter IDN from the selected telegram.

Refresh (F5)

This selection refreshes the contents of the telegram window to display any recent changes.

Axis Options

The *Axis Options* window is used to add a specific parameter IDN ,regardless of the mode of operation, to the cyclic data as well as enable features such as the Multiplex channel. Refer to axis parameter A-0-0004 for details.

Axis Option	Function
Optional Cyclic Data	sets A-0-0004, Bit 6 to 1
Drive PLS Fast Write	sets A-0-0004, Bit 8 to 1
Position Using Secondary Encoder	sets A-0-0004, Bit 11 to 1
Configure Minimum Cyclic Data	sets A-0-0004, Bit 14 to 1

Table 7-39: Axis Options

ELS Options

The *ELS Options* window is used to set several options for the ELS or CAM motion. It is also used to remove specific parameter IDNs from the cyclic data. Refer to axis parameters A-0-0004 and A-0-0164 for details.

ELS Option	Function
Remove Dynamic Phase Offset	sets A-0-0164, Bit 2 to 1
Remove MFG Class 3 Diagnostics	sets A-0-0164, Bit 3 to 1
Remove Target Position	sets A-0-0164, Bit 4 to 1
ELS Secondary Mode	sets A-0-0164, Bit 9 to 1

Table 7-40: ELS Options

System Options

The *System Options* window is used to enable a 4Mbps Sercos transmission rate, set a Sercos refresh delay, and set the minimum Sercos cycle time. Refer to control parameters C-0-0010, C-0-0098 and C-0-0099 for details.

7.4 Drive Commissioning

VisualMotion 11 software, when used in combination with GPP 11 or GMP 11 firmware, uses DriveTop, a software application designed for commissioning and configuring Rexroth digital drives.

Note: Earlier versions of GPP firmware, such as GPP8 or GPP7 launch the Drive Parameter Editor tool. Refer to *VisualMotion 8 Application* manual for information on the Drive Parameter Editor.

DriveTop

VisualMotion 11 uses an integrated version of DriveTop (16V06) software. During VisualMotion installation, DriveTop is automatically installed.

Initial setup of the drive parameters must be performed in the control before complete drive parameterization can be performed.

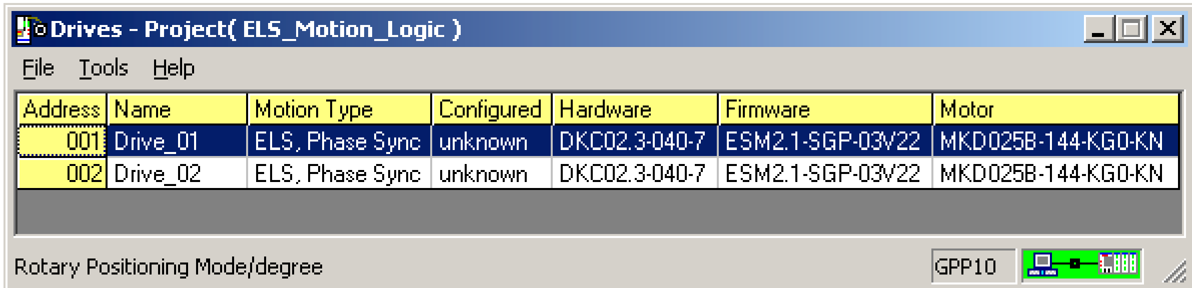
Note: It is recommended that no stand-alone version of DriveTop be installed on the PC.

Use the following steps to start DriveTop and begin drive commissioning:

1. Open VisualMotion Toolkit, with a valid project, and switch the project to online mode.
2. Clear any drive errors that may exist.
3. Transition the drives from parameter mode to operation mode.

Note: Transitioning the drive from parameter mode to phase 4 configures the drives with the correct operating modes.

4. Start the commissioning process by selecting ***Commission ⇒ Drive Overview***.



Drives_Project.tif

Fig. 7-82: Drives – Project Window

5. Double-click on a drive to open the *Drive Status* window in DriveTop.

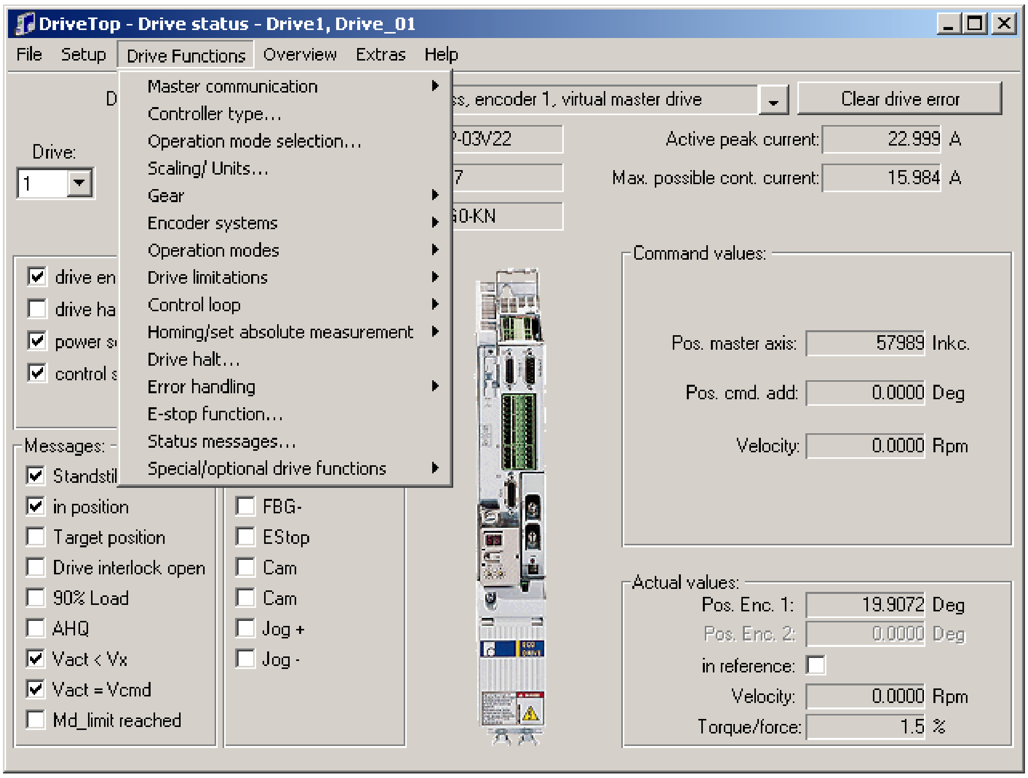


Fig. 7-83: Main DriveTop window

The default settings that appear in all of the DriveTop windows are dependent on the motor/drive combination that is currently being commissioned. Since all of the settings in DriveTop are related to actual drive parameter settings, the **Help** buttons in the different DriveTop windows are designed to launch help on the parameters that are used in the currently opened window.

DriveTop Wizard

When commissioning a drive for the first time, the DriveTop wizard can be used to guide the user through the necessary steps.

To launch the DriveTop wizard:

1. Pick a drive to commission from the **Drive:** drop-down list and select ***Setup ⇒ Initial start-up drive…***
2. Click on the **Start setup** button to start the process.
3. Click ***Next>*** in the *Warning* window indicating that standard parameter values in the drive will be overwritten if you continue.

Note: Clicking the **Next >** button in any window continues the wizard. At any point, clicking the **Exit** button accepts any modifications made and ends the wizard. All windows available in the wizard can be found under the ***Drive Functions*** menu selection.

7.5 Jogging

The jog tool is used to jog an axis in manual or automatic mode while the project is in online or service mode. Only axes assigned to a VisualMotion task and configured or switched to either *Velocity* or *Single Axis* operating modes can be jogged using this tool.

Note: An axis that is assigned to IndraLogic cannot be jogged using this tool. The PLC program must access the relevant jogging parameters in order to jog the axis. Refer to Axis Configuration on page 7-14 for details.

The jog tool is launched by selecting ***Tools ⇒ Jogging***… The *Jog* window displays the Axis-mode, the current feedback position for the active axis, the task assignment and the current system status messages and extended diagnostics.

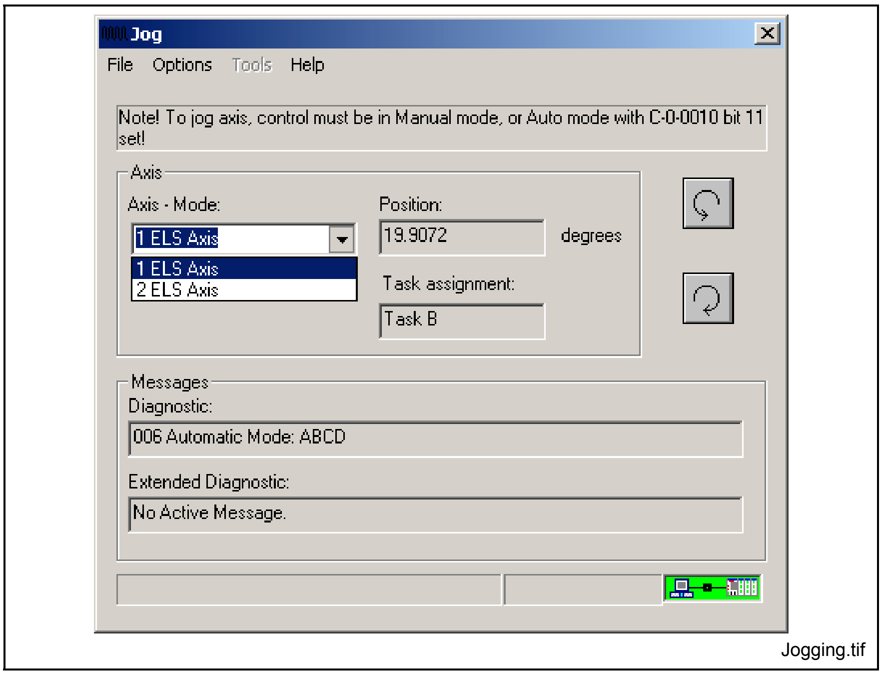


Fig. 7-84: Jog Window

Note: *Coordinated Motion* axes cannot be jog using this tool and must be jogged in manual mode using the Task Jog registers 007-010 for task A-D, respectively.

Any axis configured for velocity or single axis mode can be jogged in manual mode.

To jog an axis in automatic mode, control parameter C-0-0011, bit 11 must be set to 1. By default C-0-0011, bit 11 is set to 0 and can only be modified when the system is switched to parameter mode.

WARNING

Independent Axis Movement When Jogging in Automatic Mode

⇒ If an axis is jogged while the task is running and C-0-0011, bit 11 is set to 1, the axis can be manually jogged in either direction but will no longer be part of the running program. The program must be stopped and restarted to synchronize the axis to the program. It is the responsibility of the user to ensure that an axis running in the program is properly stopped before manually jogging the axis.

Before jogging an axis, set the desired axis jogging, system jogging and task options to the desired behavior.

Holding down the jog reverse or jog forward buttons, respectively, changes the state of bit 3 (Jog_Reverse) or bit 2 (Jog_Forward) in the Axis_Control Register.

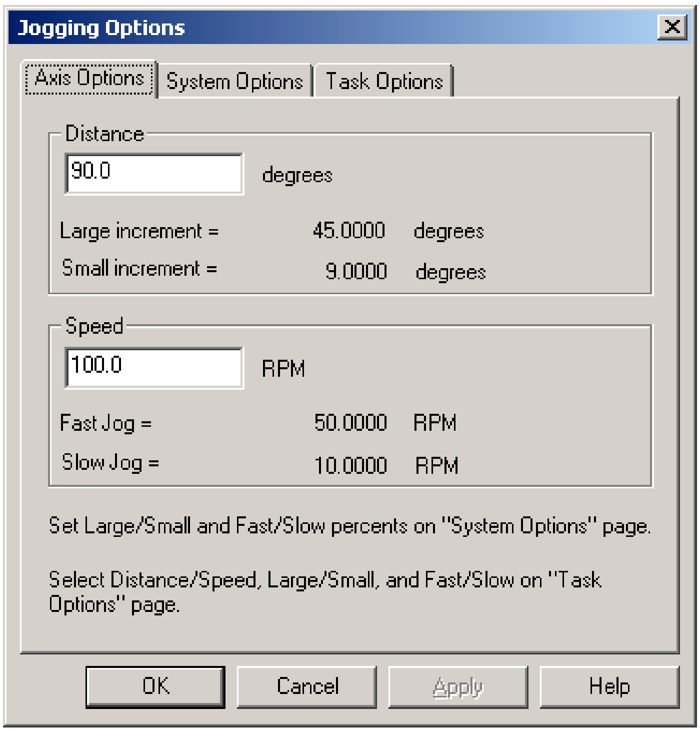
A low-to-high (0-1) transition on these bits causes motion to start in the negative (bit 3) or positive (bit 2) direction. A high-to-low (1-0) transition immediately stops the motion. Motion is also stopped when the task mode selection changes, or when a travel limit or incremental distance is reached.

Jogging Options

Selecting the ***Options*** menu opens the *Jogging Options* window where the axis option, system option and task options can be configured.

Axis Options

Selecting the *Axis Options* tab displays the distance and speed setting for all axes.



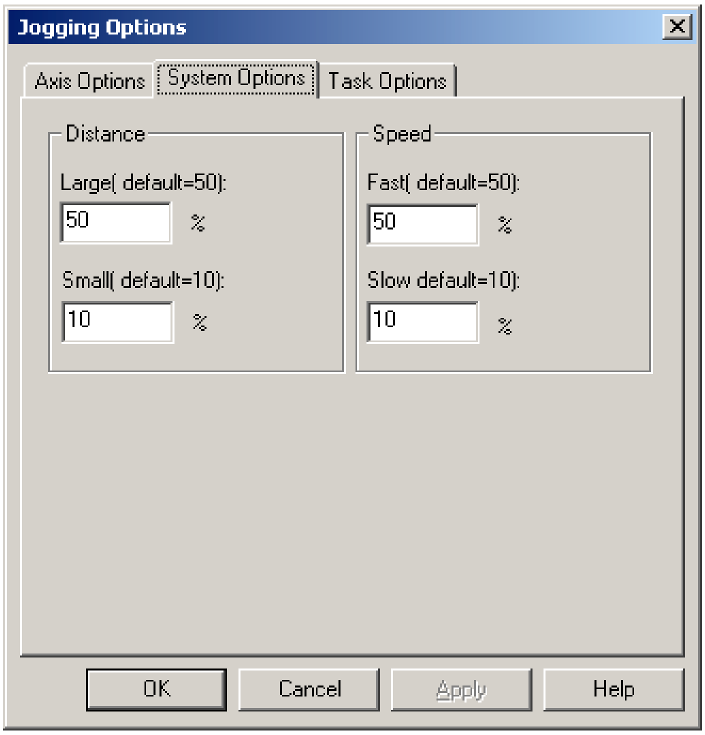
Jogging_Axis.tif

Fig. 7-85: “Axis Options” Window

The *Axis Jogging Options* allows input of the maximum jog distance and speed. The maximum distance value is stored in parameter A-0-0025. The maximum speed is stored in parameter A-0-0026. The large/small increment values and the fast/slow velocities are calculated according to the values entered under the *System Options* tab.

System Options

Selecting the *System Options* tab displays the distance and speed percentages that are used to calculate the actual speed and distance for jogging.



Jogging_System.tif

Fig. 7-86: “System Options” Window

The *System Options* window is used for setting the increments and velocities used for fast and slow jogging. The *Increment* data area is used to set the *Large* and *Small* percentage of the maximum distance for a single-step jog operation. The maximum is defined by axis parameter A-0-0025, Maximum Jog Increment. Similarly, the *Speed* data area is used to set the *Fast* and *Slow* jog speeds as a percentage of the maximum velocity, which is defined by axis parameter A-0-0026, Maximum Jog Velocity.

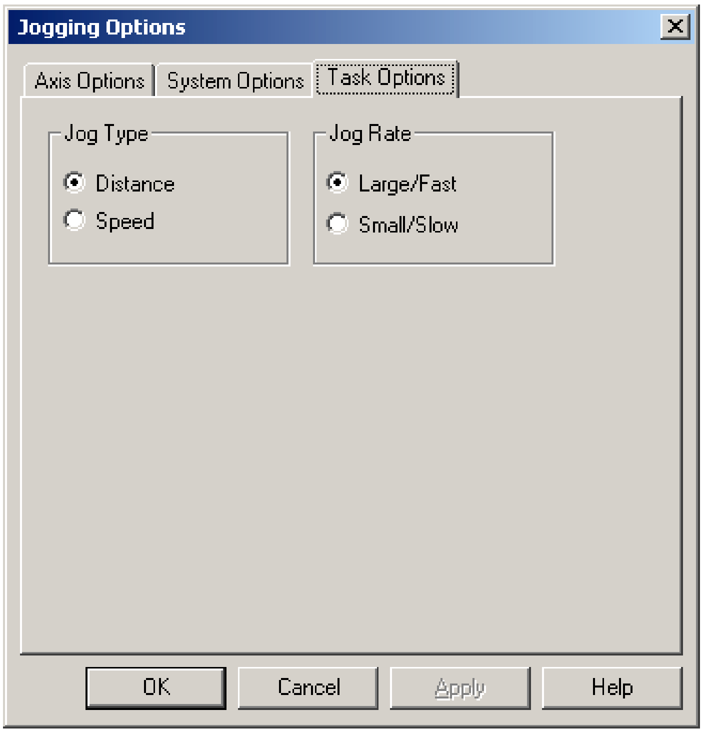
The values are stored in the following parameter locations:

- Large Increment - C-0-0052
- Small Increment - C-0-0053
- Fast Speed - C-0-0055
- Slow Speed - C-0-0056

Rexroth Bosch Group

Task Options

Selecting the *Task Options* tab displays the setting for *Jog Type* and *Jog Rate*.



Jogging_Task_Options.tif

Fig. 7-87: “Task Options” Window

The *Task Options* tab allows the user to select the *Jog Type* and *Rate*. The *Type* can be Distance (incremental) or Speed (continuous). The selected jog type takes effect with the next transition on the jog forward or reverse buttons. The following table lists the drive's operating mode based on the selected jog type:

Jog Type	Functionality	Drive's Operating Mode
Distance	jogging motion stops after the large or small travel limit is reached	switches to single axis during jog
Speed	jogging motion continues until the jog button is released or the travel limit is reached	switches to velocity mode during jog

Table 7-41: Jog Type

The *Rate* can be Large/Fast or Small/Slow**.** When *Distance* is selected, the options are Large or Small. When *Speed* is selected, the options are Fast or Slow.

Refer to the relevant *Task Jog Control Register, Bits 1 and 6* for details.

Control Selection

The ***Tools*** menu selection is only available for service mode. In project mode, the communication settings are handled by the project. Selecting Tools opens the *Control Selection* window. Refer to section 13.10 in volume 2 of the *VisualMotion Functional Description* for details.

Position Limits

Position Limits Enabled

When a jog forward is started, the control sets the target position of the axis to the positive travel limit. When a reverse jog is started, the target position is set to the negative travel limit. When the jog is stopped, the target velocity is set to zero, but the target position remains at the travel limit.

Position Limits Not Enabled - Diax03, Version 04 and Greater

On Diax03 drives, version 4 and later, the drive is switched to velocity mode. The ramps are generated internally by the drive using the axis jog acceleration parameter A-0-0023.

Before performing single-axis positioning using the axis_move command, it is necessary to execute the els_mode command to switch the drive back into single-axis mode. A cycle stop followed by a cycle start will also reset the drives to single-axis mode.

Position Limits Not Enabled - Other Drives

The drive remains in single-axis mode. The control will continually increase or decrease the target position by a small amount to keep the drive moving, until the jog is stopped. It sets the acceleration to a value high enough so that the drive does not decelerate. The jog acceleration parameter will be used only if it is lower that this value.

Jog Commands for HMI Devices

Windows based HMI devices can safely jog an axis by writing to axis control registers using DDE (ASCII Protocol) or OPC commands. The following figure shows the communication layers between an HMI application and the control. The commands described in this section are "**write-only**" and are transmitted from the client to the server.

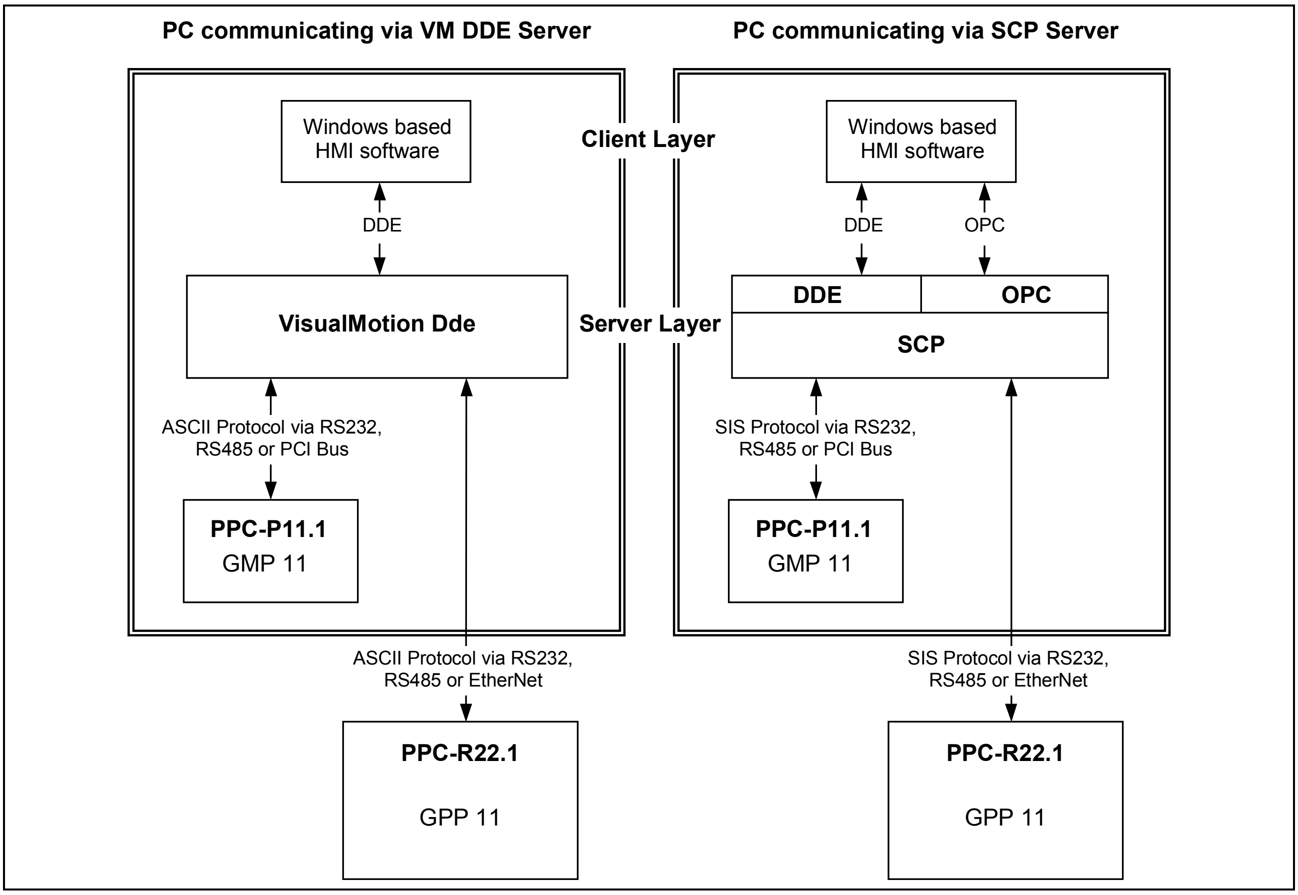


Fig. 7-88: VM Dde and SCP Communication Overview

Jog Commands

The following table is an overview of the DDE and OPC jog commands that are supported by VisualMotion. Although control registers can be modified by other methods, the protection of a communication timeout period (i.e., "Safe Jogging") is only available when using the following commands:

DDE Command		OPC Command		
Item	Data	Item string	Synch/Asynch Write	Description
JA X_1,X_2,X_3	1	Device Name, VJA,X_1,X_2,X_3	1	jog command without error detection
JB X_1,X_2,X_3	1	Device Name, VJB,X_1,X_2,X_3	1	jog command with error detection
JC X_1,X_2,X_3	0	Device Name, VJC,X_1,X_2,X_3	0	cancel jog immediately

X_1 – always 0

X_2 – axis control register number

X_3 – axis control register bit

Table 7-42: Dde and OPC Jog Commands

Relevant Axis Control Registers

The following table lists the axis control registers that can be used for jogging:

Register Number	Axis Number
11 – 18	Axis 1 – Axis 8
209 – 240	Axis 9 – Axis 40
441-464	Axis 41 – Axis 64

Table 7-43: Axis Control Registers

Jog an Axis via DDE

The DDE interface in the VisualMotion Dde and SCP servers support three DDE (ASCII Protocol) commands for safely jogging an axis via the respective axis control register while in manual mode. To jog an axis in automatic mode, refer to Jogging on page 7-85 for details.

The jogging of an axis via a DDE command is timed by the communication time-out period value in control parameter C-0-0016. If a jog command is unexpectedly cutoff during transmission (i.e., the serial cable is removed during transmission), the axis will stop jogging when the communication time-out value in C-0-0016 is reached.

Jog without Error Detection

While jogging an axis using the JA command if the communication time-out period is reached before a new JA command is sent, the axis will stop and no error is reported to the control.

Jog with Error Detection

The JB command is identical to the JA command but differs in that if the time-out period is reached, error "504 Communication Timeout" is reported and logged in the control.

Jog an Axis via the Dde Interface

To safely jog an axis using a DDE command,

1. Set the communication time-out period (C-0-0016) to a value between 50 and 2000 ms (default = 2000).
2. Switch the Task_Control register assigned to the relevant axis to manual mode by setting the Mode_Auto_nManual bit to 1.
3. Send the following *Item* and *Data* to the DDE interface:

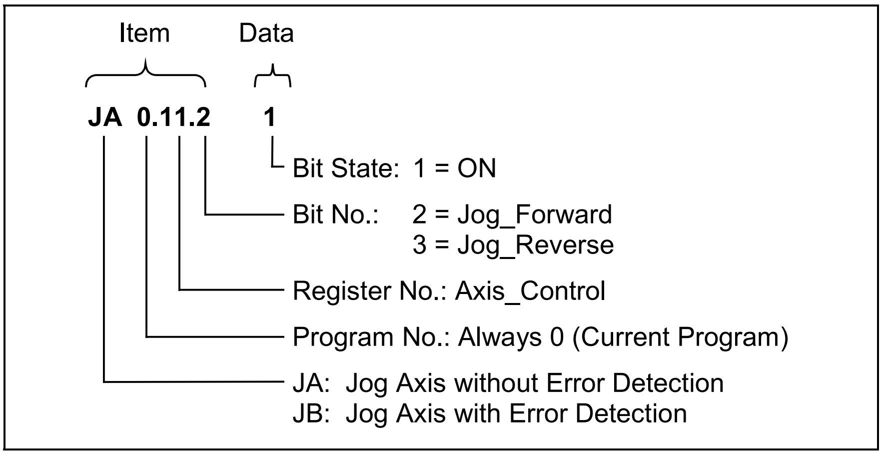


Fig. 7-89: Jog an Axis via Dde

Rexroth
Bosch Group

Note: To keep an axis jogging, the application must keep sending a JA or JB command to the control within the set time in C-0-0016.

Cancel Jog Immediately via Dde Interface

To stop an axis that is currently jogging, the application can stop sending the JA or JB commands and wait for the communication time-out period (C-0-0016) to be reached or send the following JC command to immediately stop the axis:

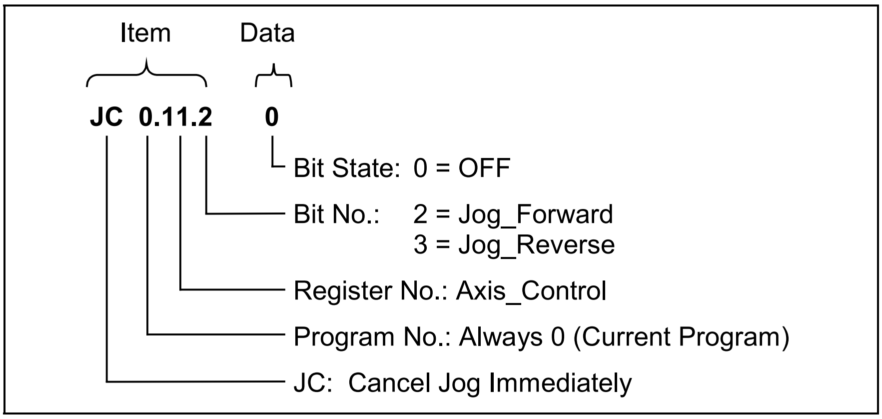


Fig. 7-90: Cancel a Dde Jog Command

Jog an Axis via OPC

The OPC interface in the SCP server supports three commands for safely jogging an axis via the relevant axis control register. These commands function the same as the DDE timed transmission commands JA, JB and JC but use a slightly different syntax structure as follows:

SCP Device Name, SCP-Command

Refer to *OPC Item Structure* in section 18.9 of volume 2 for details.

Jog an Axis via OPC Interface

To safely jog an axis using an OPC command,

1. Set the communication time-out period (C-0-0016) to a value between 50 and 2000 ms (default = 2000).
2. Switch the Task_Control register assigned to the relevant axis to manual mode by setting the Mode_Auto_nManual bit to 1.
3. Send the following OPC *Item* and *Synch/Asynch Write* command:

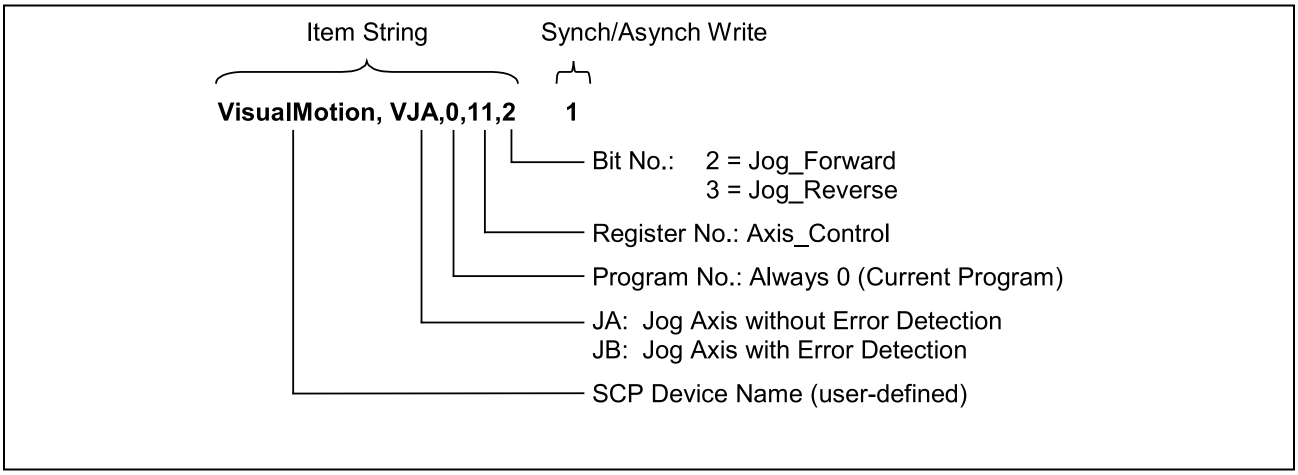


Fig. 7-91: Jog an Axis via OPC

Note: To keep an axis jogging, the application must keep sending an OPC synch/asynch write command to the control within the set time in C-0-0016.

Cancel Jog Immediately via OPC Interface

To stop an axis that is currently jogging, the application can stop sending the OPC command and wait for the communication time-out period (C-0-0016) to be reached or send the following OPC *Item* and *Synch/Asynch Write* command:

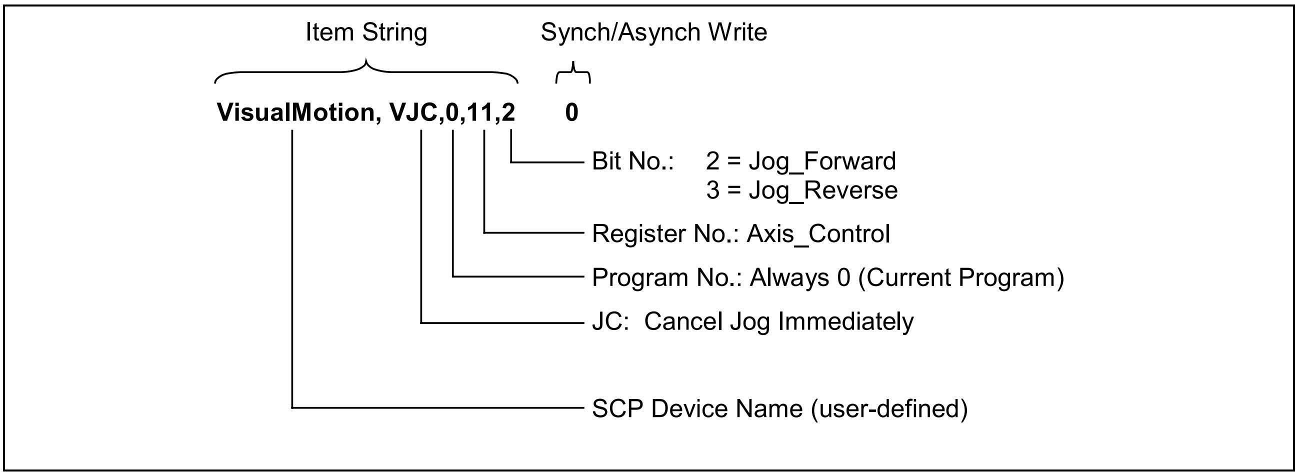


Fig. 7-92: Cancel an OPC Jog Command

Rexroth
Bosch Group

7.6 Profibus Fieldbus Slave Interface

General Information

Version Note:
Information in this document is based on VisualMotion Toolkit software version 10VRS and PPC-R firmware version GPP 11VRS.

Note: Fieldbus master devices are configured using IndraLogic, refer to *Configuring a Profibus Fieldbus Master* on page 7-36 for details.

VisualMotion 11 software is downward compatible with GPP firmware, but, depending on the hardware platform selected, the type of fieldbus communication selection may be limited. The following table lists the fieldbus firmware versions and the available fieldbus interfaces for each version.

Fieldbus Interfaces	PPC-R GPP07VRS	PPC-R GPP08VRS	PPC-R GPP09VRS/ GPP 11VRS	PPC-P GMP09VRS/ GMP 11VRS
Profibus	●	●	●	No Fieldbus Slave Support

Table 7-44: Fieldbus Firmware Version and Interface Type

Note: For fieldbus hardware information, refer to the *VisualMotion 11 Project Planning Manual*.

PPC-R System Description with a Fieldbus

The PPC-R can operate on a serial fieldbus interface (network) by means of a fieldbus expansion card that communicates with the PPC-R via dual-port RAM. The function of the fieldbus card is similar to that of a network card in a PC; it allows communication with other devices on the network.

In Fig. 7-93, a typical fieldbus interface is illustrated with the following:

- **Fieldbus Master** - PLC fieldbus interface
- **Fieldbus Slave** - PPC-R fieldbus interface

In this document, we will refer to the PLC as the **fieldbus master** and the PPC-R as the **fieldbus slave**.

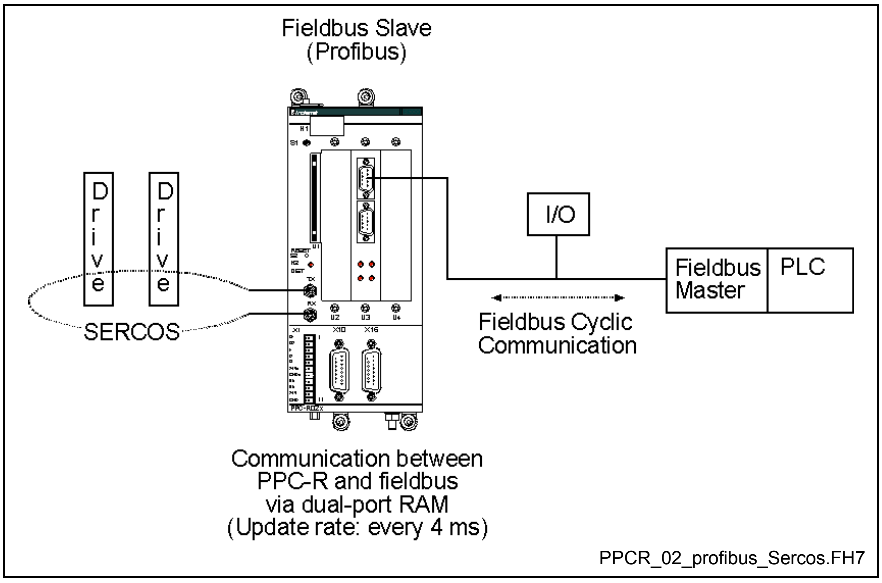
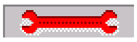
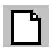


Fig. 7-93: Sample Master/Slave Setup with Fieldbus Card

With the PPC-R, the fieldbus card can be used **only** as a **slave** card in a master/slave setup.

The VisualMotion Fieldbus Mapper

In the VisualMotion software package, the Fieldbus Mapper is a tool used to set up fieldbus configuration and data mapping. Fieldbus hardware platform selections are made through the Fieldbus Mapper window with VisualMotion Toolkit in Service Mode. To select the Fieldbus hardware platform:

1. Open VisualMotion in Service Mode, indicated by the service mode symbol () in the lower right corner of the VisualMotion Toolkit window.
2. Select **Commission** ⇒ **Fieldbus Mapper** to open the FBMapper window.
3. Click  **or** select ***File*** ⇒ ***New*** to open the *Fieldbus Slave Definition* window.

When a hardware platform is selected, only the fieldbus types available for that platform can be selected, refer to Table 7-44.

Data Transfer Direction (Output vs. Input)

In the VisualMotion Fieldbus Mapper, output and input are always described with respect to the fieldbus master. The definitions for output and input are:

output: the communication from the PLC to the PPC-R (i.e. from the fieldbus master to the fieldbus slave).
Synonyms for this type of communication: **send** or **write** data.

input: the communication from the PPC-R to the PLC (i.e. from the fieldbus slave to the fieldbus master).
Synonyms for this type of communication: **receive** or **read** data.

Fieldbus Data Channel Descriptions

The Rexroth Profibus fieldbus interface card for the PPC-R supports the cyclic (DP) channel, which is made up of the following two parts:

- ***Real-Time Channel*** (for **single** and **multiplex** channels)
- ***Parameter Channel*** (for systems requiring non-cyclic transmissions)

Cyclic (DP) Channel

Cyclic data is user-defined. It is stored in two ordered lists (C-0-2600 for input data, C-0-2601 for output data) and transmitted serially over the bus.

The cyclic data channel is limited to 64 input words and 64 output words. PPC-R data types consume these words in either one-word (or 16-bit) groups for PPC-R registers or two-word (or 32-bit) groups for all other data types.

The PPC-R mapping list is scanned every 4 ms and data is sent and received to/from the fieldbus slave board's dual port RAM.

The cyclic data channel can be made up of any combination of the following data types:

- Real-Time Channel
  - Single Channel
  - Multiplex Channel
- Parameter Channel

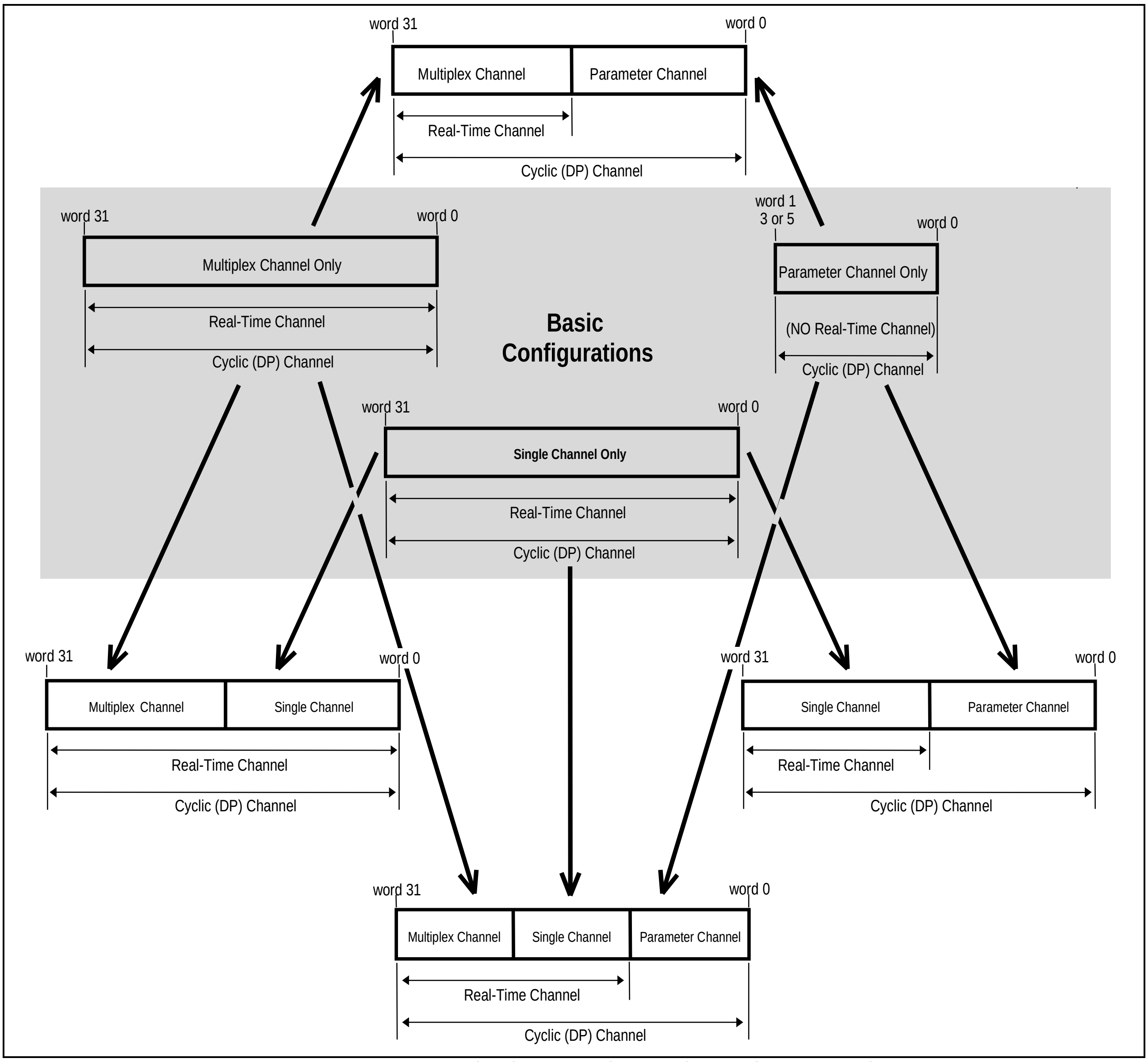


Fig. 7-94: Configuration Options for the Cyclic Data Channel

The Real-Time Channel

In the real-time channel, data is updated cyclically between the fieldbus master and slave. This channel contains two possible data types: **single** and **multiplex**.

Cyclic Data: Types and Sizes

The following table outlines the PPC-R data types that can be transmitted via the cyclic channel and the amount of space (in 16-bit data words) that each data type consumes.

Note: The cyclic data mapping lists support only 16- and 32-bit data of the following types for reading and writing:
- Integer
- Float
- Binary (used in control parameters)
- Hex (used in control parameters)

For all other data types (e.g. diagnostic messages - “strings”), use the Parameter Channel.

PPC-R Data Type	Data Size (in 16-Bit Words)
Register	1
Program Integer (currently active program ONLY *)	2
Program Float (currently active program ONLY *)	2
Global Integer	2
Global Float	2
Card Parameter	2
Axis Parameter	2
Task Parameter	2
Note: Drive parameters "S" or "P" cannot be transmitted cyclically because of the inherent delay of parameter access over the SERCOS service channel. See ***"Parameter Channel."*** However, if a drive parameter is mapped to an Axis Parameter, that Axis parameter could be used in cyclic data (see description of Axis Parameters 180-196 in the ***VisualMotion Functional Description***).	
*** Important Note:** Integers and floats are shown only for the currently active program. Each time you activate a new progam, the fieldbus reads/writes to the newly-activated program.	

Table 7-45: PPC-R Cyclic Data Types and Sizes

Single Data Types

Single data types are mapped directly in the cyclic mapping ordered lists (C-0-2600, C-0-2601). The data types are updated every 4 ms via dual-port RAM.

Multiplex Data Types Cyclic Data Channel)

(In some multi-axis applications, 64 words of cyclic data transfer are not sufficient to meet the requirement of the application.

When insufficient data transfer space is available, multiplex data can be set up within the cyclic channel. One multiplex container acts as a placeholder for multiple possible PPC-R data types (all of the same word size). The currently transmitted PPC-R data type is based on an index value placed in a multiplex control or status word attached to the end of the cyclic list. Depending on the index specified by the master, the multiplex channel permits a different set of data within the cyclic channel to be transferred as current real-time data. Multiplex containers can be added to the input and output lists separately, and the input and output indexes can be designated separately (in the control and status words).

Note:	Using the multiplex channel reduces the maximum number of usable words for storing control data to 63. The 64th word (or last used word, if fewer than 63 words) is used as the multiplex entry control/status word.

Note:	When using VisualMotion 11 with GPP 7 firmware, a maximum of 15 multiplex containers and a maximum of 180 mapping items can be transmitted in the input or output list. This limitation of mapping objects means that you cannot multiplex all 15 containers with all 32 available indexes (=480 items). For VisualMotion 11 with GPP 8 or 9 firmware, there is no limitation for multiplexing (each of the first 63 words may be multiplexed with up to 32 indexes).

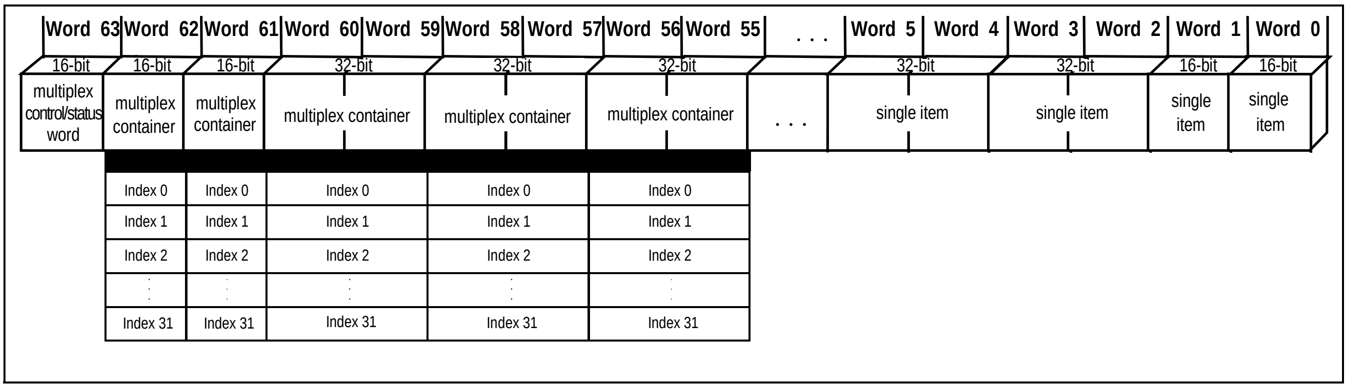


Fig. 7-95: Sample Command (PLC⇒PPC-R) or Response (PPC-R ⇒PLC)

The multiplex control and status words serve to command and acknowledge multiplex data transferred between the fieldbus master and the fieldbus slave. The **control** word is associated with **output** communication (PLC⇒PPC-R). The **status** word is associated with **input** communication (PPC-R⇒PLC). Single data items are not affected by the multiplex control and status words.

Note:	For specific information about how the fieldbus master uses the multiplex control and status words, refer to Multiplexing on page 7-110.

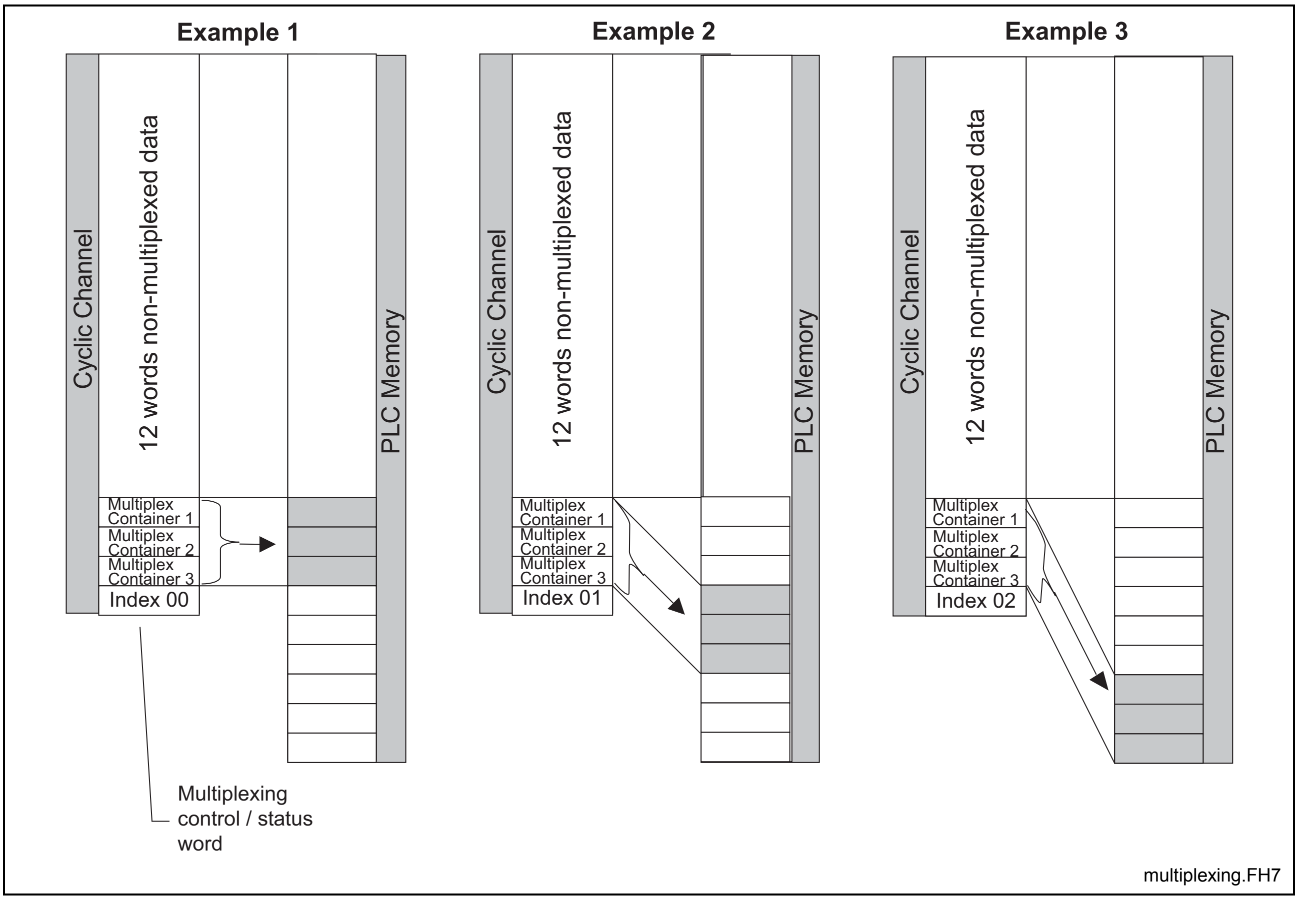


Fig. 7-96: Examples for Reading Data via the Multiplex Channel

Parameter Channel

For Profibus systems using the PPC-R/VisualMotion configuration, a subset of the cyclic (DP) channel can be allocated for non-cyclic communications (e.g. parameterization and extended diagnostic information). This subset of the cyclic channel is called the **Parameter Channel**.

Note: The Parameter Channel is always allocated as the first 2, 4 or 6 words of the Profibus cyclic (DP) channel. The length of the Parameter Channel plus the length of the Real-time Channel used to exchange cyclic data represents the entire length of the DP channel (maximum total length: 64 words). Refer to *Fig. 7-94* for DP channel configuration options.

Two messaging formats are available in the Parameter Channel, to allow for a varying degree of implementation, depending on application requirements:

- **Short Format 3**- messaging format that provides direct access to Rexroth mapped objects (Registers, Global Floats, Global Integers, Floats, Integers, as well as Card, Axis, Task, and Drive S and P parameters).

Note: List parameters can be accessed using the Data Exchange Object. For further explanation of how this format functions, refer to *Messaging Formats* on page 7-115.

- **VisualMotion ASCII Format**- provided for backward-compatibility with previous VisualMotion versions. For specific information about this format, refer to the *VisualMotion 11 Functional Description* manual.

Rexroth Bosch Group

Fieldbus Mapper Functionality

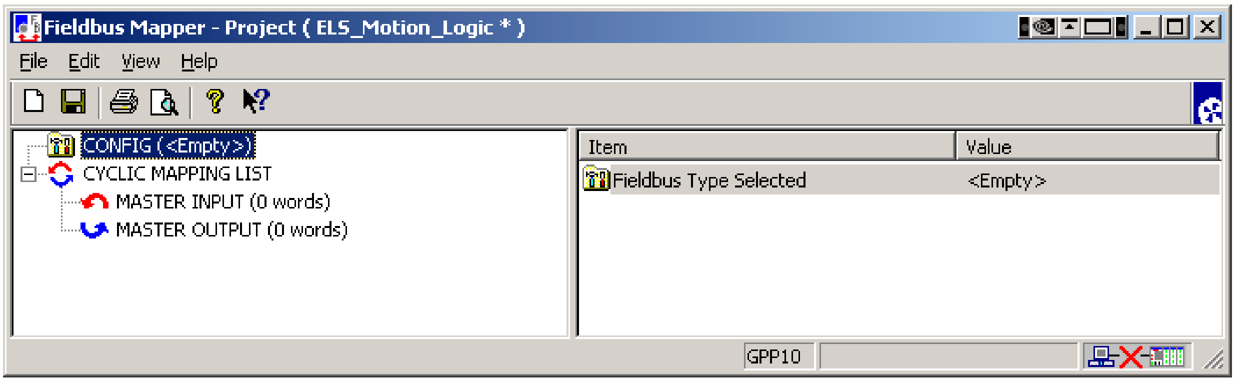
Initializing the Fieldbus Mapper from VisualMotion 11

When configuring a new Fieldbus Mapper, the procedure will vary depending on the mode of VisualMotion.

Project Mode

To configure a Fieldbus Mapper in Project mode:

1. Open an existing program or create a new program. You must be using PPC-R hardware with GPP firmware to use the Fieldbus Mapper described in this document.
2. Select ***Commission*** ⇒ ***Fieldbus Mapper***. The main Fieldbus Mapper window opens empty (refer to Fig. 7-97).



FB_Mapper_New.tif

Fig. 7-97: FBMapper – Project Window

3. Click **or select *File* ⇒ *New***.

A “setup wizard” goes through three steps:

- Fieldbus Slave Definition
- Fieldbus Slave Configuration
- Cyclic Data Configuration

4. Enter the information requested in the setup windows. For more details on each step, refer to *Fieldbus Slave Definition*, *Fieldbus Slave Configuration*, and *Cyclic Data Configuration* for detailed information about each configuration step.
5. Save the file (automatically has a ***.prm** extension) to the project folder on the computer if you are in offline mode If you are in online mode, download the configured Fieldbus Mapper to the control.

If you are offline, the configured Fieldbus Mapper is saved to the project file. If you are online, download the configured Fieldbus Mapper to the control and synchronize project.

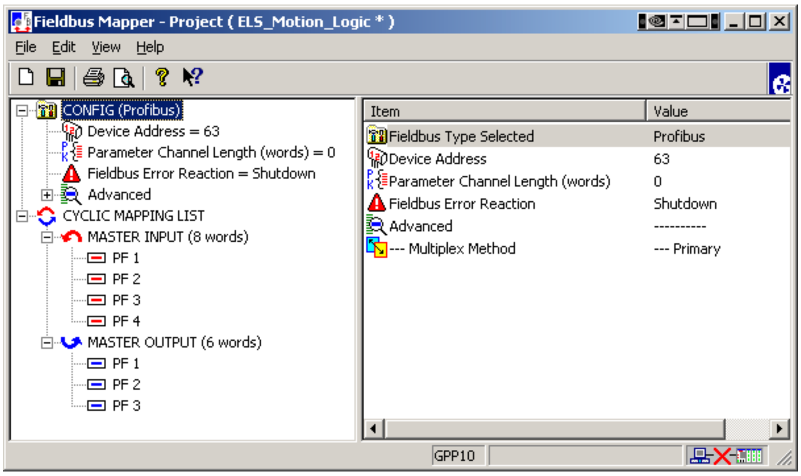
When editing an existing Fieldbus Mapper, refer to the *Programming* chapter for information on how commissioning tools are handled in Project Mode, online and offline, and in Service mode.

Service Mode

If you are configuring a new Fieldbus Mapper while VisualMotion is in Service mode, the Fieldbus Mapper window will open empty when commissioned. After configuring the new Fieldbus Mapper, download the data to the control. Selecting ***File*** ⇒ ***Save*** will save the configured Fieldbus Mapper to a separate file from the project file. To bring the Fieldbus Mapper file into the project, use the import procedure:

1. Select File ⇒ Import.
2. Browse to find the desired file (***.prm** extension).

3. Click ***Open***. The main Fieldbus Mapper window appears, which lists the configuration information. Refer to Fig. 7-98 below.

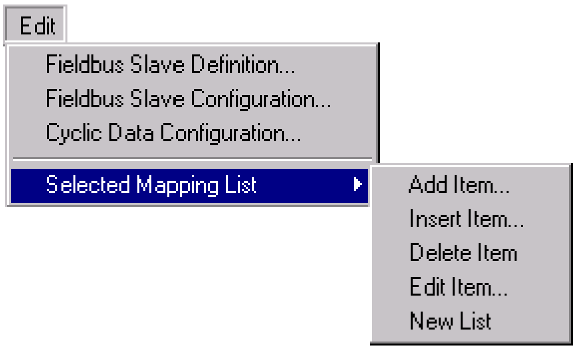


FB_Mapper_Main.tif

Fig. 7-98: FBMapper-Project Window

Editing a Fieldbus Mapper

To Add/Insert, Edit, or Delete an item in a Fieldbus Mapper file, open the ***Selected Mapping List*** menu from the Edit menu (refer to Fig. 7-99 below). For more information about each step, refer to *Fieldbus Slave Definition*, *Fieldbus Slave Configuration*, and *Cyclic Data Configuration* for detailed information about each configuration step.



FB_Mapper_Edit_Menu.tif

Fig. 7-99: Fieldbus Mapper Edit Menu

Note: You can also directly add, insert, delete, edit an item, or create a new list by:

- clicking on the item to be edited in the main Fieldbus Mapper window and selecting the desired function under ***Edit ⇒ Selected Mapping List***

OR

- right-clicking on an item to display a menu of functions

Fieldbus Slave Definition

To configure a Fieldbus Slave, select the fieldbus type and hardware platform in the *Fieldbus Slave Definition window*, refer to Fig. 7-100. Refer to Table 7-44 for a list of the hardware platforms available for Profibus. The hardware platform can only be selected when VisualMotion is in Unsynchronized mode.

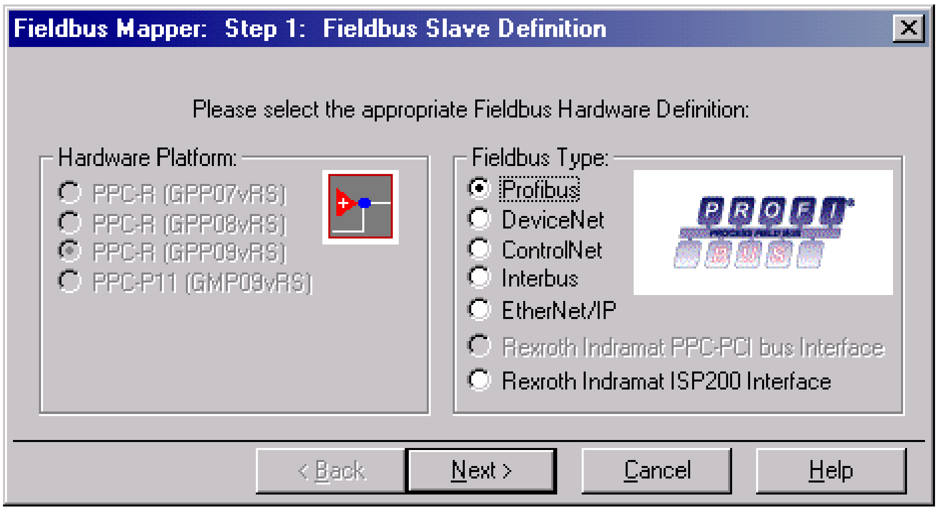


Fig. 7-100: Fieldbus Slave Definition Window

Fieldbus Slave Configuration

The Profibus Fieldbus Slave Configuration window is shown in Fig. 7-101 below.

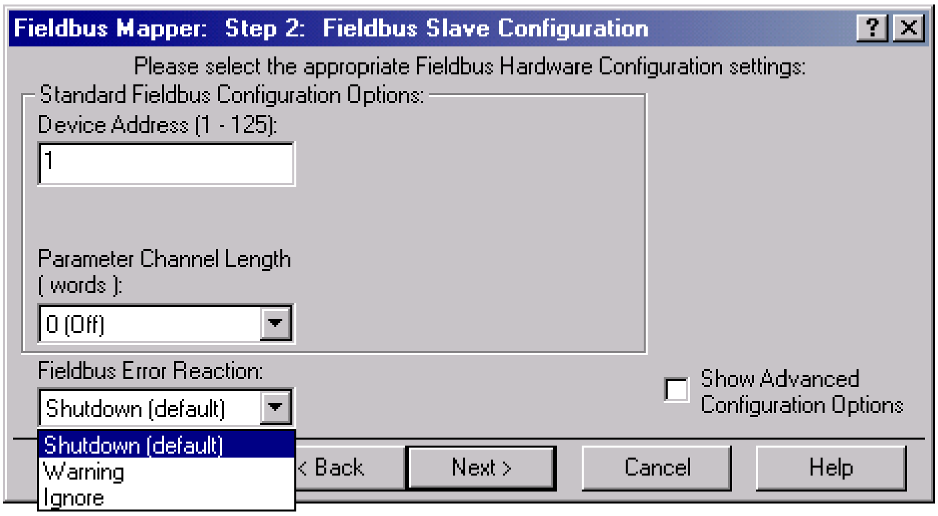


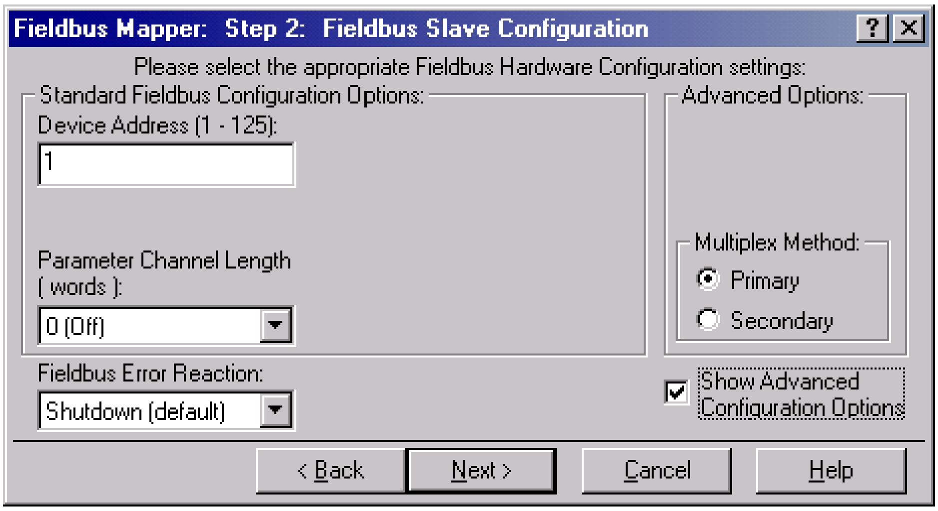
Fig. 7-101: Fieldbus Slave Configuration

Standard Fieldbus Configuration Options:

- **Device Address (0-125):** set to a unique number for the devices on the bus
- **Parameter Channel Length (words):** set to 0 (Off), 2, 4 or 6 words. If 2, 4 or 6 words are selected, these are automatically allocated for the Parameter Channel in the Cyclic Data Input and Output Lists.

Fieldbus Error Reaction: Set the Error Reaction to Shutdown (default), Warning or Ignore. Refer to Fieldbus/PLC Cyclic Read/Write Monitoring on page 7-108 for detailed information about each setting.

Advanced Configuration Options The **Advanced Options** field is shown only if the checkbox next to **Show Advanced Configuration Options** is checked (refer to Fig. 7-102 below). In most cases, the default options should apply.



FB_Slave_Config_PB_Adv.tif

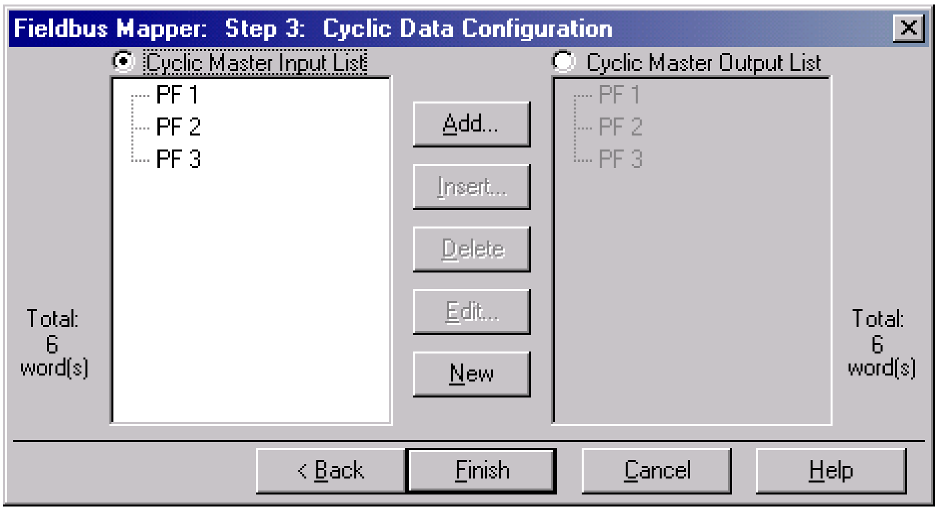
Fig. 7-102: Fieldbus Slave Configuration: Advanced

- **Multiplex Method:** select **Primary** or **Secondary** (Primary is the default). Select **Secondary** only if you have an inconsistent fieldbus master. Refer to *Multiplexing* on page 7-110 for detailed information about each method.

Cyclic Data Configuration

An example of the Cyclic Data Configuration window is shown in Fig. 7-103 below. In this window, four words have been allocated for the Parameter Channel (optional for Profibus fieldbuses only). If you are editing an existing Fieldbus Mapper file, the list will probably contain more items.

First, you must select the **Cyclic Input List** (from PPC-R to PLC) or the **Cyclic Output List** (from PLC to PPC-R).



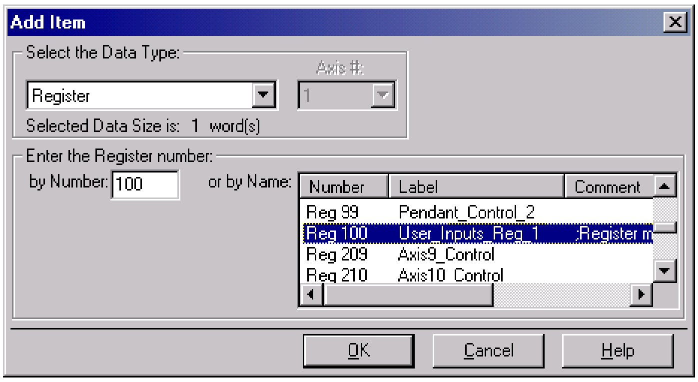
Cyclic_Data_Config.tif

Fig. 7-103: Cyclic Data Configuration

Adding an Item to the List

1. Select the Cyclic Input List or the Cyclic Output List.
2. Click ***Add***. The window in Fig. 7-104 is displayed. Select the Data Type (for example, Register).

Rexroth
Bosch Group



Cyclic_Data_Add.tif

Fig. 7-104: Add Item to Cyclic Data

3. Enter the required information (for example Register Number) or select it from the drop-down list. Only the available data types for your designated VisualMotion hardware setup and fieldbus type are listed.
4. Click ***OK*** to add the selected item to the list.

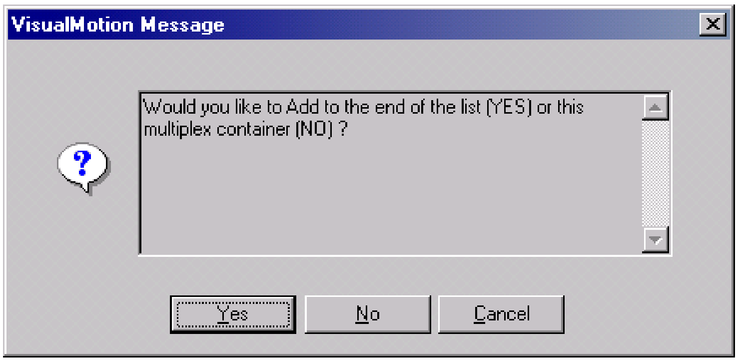
Adding Multiplex Containers to the List

1. Select the Cyclic Input List or the Cyclic Output List.
2. Click ***Add***.
3. In the *Add Item* window under **Select the Data Type**, select **Multiplex Container 16-bit** (for Registers) or **Multiplex Container 32-bit** (for all other data types).

Note: At this point, the Multiplex Containers do not yet contain any items. To add multiplex items, refer to below.

Adding Items to an Empty Multiplex Container

1. In the Cyclic Data Configuration window, select the multiplex container to which you want to add items.
2. Click ***Add***. The window in Fig. 7-105 below appears. Because it is unclear whether you would like to add to the list or to the multiplex container, the Fieldbus Mapper is requesting clarification.

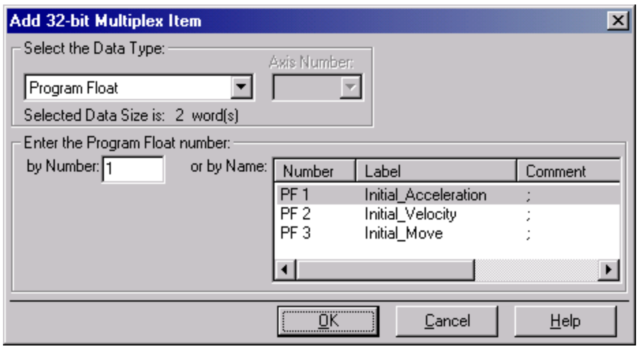


Add_Multiplex.tif

Fig. 7-105: Add Item or Multiplex Item Window

Note: For subsequent items, highlight any of the indexes within the multiplex container before clicking ***Add***, and the Fieldbus Mapper will know you want to add to that container.

3. To add to the selected multiplex container, click ***No***. The window in Fig. 7-106 below is an example for adding a 32-bit multiplex item.
4. Select the desired item to be added to the multiplex container in the *Add 32-bit Multiplex Item* window.



Add_Multiplex_Item.tif

Fig. 7-106: Select Data Type for Multiplex Item

Note: In addition to the data types that can be added to the multiplex list, an empty item called **Multiplex Empty Item** is available to fill a space within the multiplex container, if nothing is to be mapped to a particular index.

5. Click ***OK***. The item is automatically placed in the multiplex container as the next unassigned index item (e.g. the first item is index 00, the last is index 31).
6. Repeat for as many items as you want to add to the multiplex container, up to 32 items.

Editing the Cyclic Data Lists

To make changes to an existing list, use the following buttons:

Button	Function
Add...	Inserts a new item at the end of the list.
Insert...	Inserts a new item into the list directly before the selected item.
Delete	Removes the selected item from the list.
Edit...	Allows editing of the selected item. (To edit a list item, you may also double-click on it.)
New	Clears up the current list.

Table 7-46: Button Functions in the Cyclic Data Configuration Window

Rexroth
Bosch Group

Additional Functions

Several additional functions are available in the Fieldbus Mapper:

Menu Item	Function
Print	Print the current fieldbus configuration data.
Print Preview	Preview the printout of the current fieldbus configuration data
Print Setup	Configure printer settings

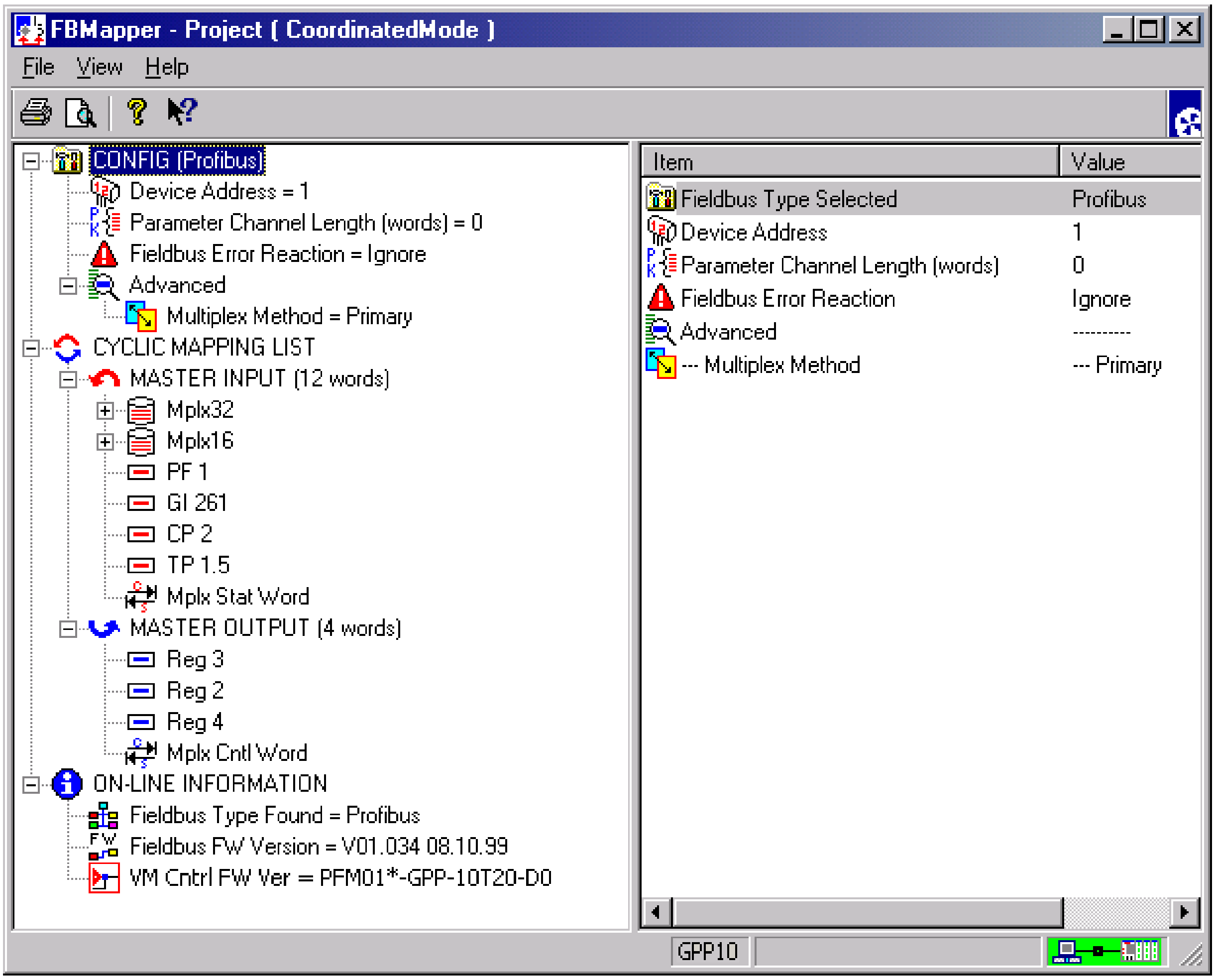
Table 7-47: Additional Functions

Getting the Fieldbus Configuration from the PPC

After getting the fieldbus configuration from the PPC while in Service Mode or Project mode (online), the following information is detected by the system and appears in the configuration list:

- Fieldbus Type Found
- Fieldbus FW (Firmware) Version
- GPP Control FW (Firmware) Version

An example is shown in Fig. 7-107 below.



FB_Mapper_Online.tif

Fig. 7-107: Online Fieldbus Configuration Information

Information for the GPP Programmer

Fieldbus Status

VisualMotion Register 19 holds the information for "Fieldbus Status." The register information can be referenced in a VisualMotion application program to respond to the status of each bit. The use of these bits is application-dependent.

Table 7-48 below contains the bit assignment for the diagnostic object 5ff2. The assigned bits are labeled with "x" and the bit number in the second row. Unassigned bits are labeled with "---."

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
---	x15	---	---	---	---	---	---	---	---	---	x5	x4	---	x2	x1

Table 7-48: Bit Assignment for VisualMotion Register 19

Bit Definitions

x1, x2 Status bits for the internal DPR (Dual-Port RAM) communication between the fieldbus slave and the PPC-R:

x1: FB Init. OK , LSB (least significant bit)

x2: FB Init. OK, MSB (most significant bit)

The bit combinations for x1 and x2 are as follows:

Bit 2 (PPC-R)	Bit 1 (Fieldbus)	Description
0	0	A reset has been executed on the DPR, **or** neither the PPC-R nor the fieldbus card have initialized the DPR.
0	1	The DPR is initialized by the fieldbus card, but **not** yet by the PPC-R.
1	0	The DPR initialization is complete. DPR has been initialized by the fieldbus card and PPC-R. Fieldbus to PPC-R communications system is ready.
1	1	Fieldbus to PPC-R communications system is ready.

Table 7-49: Possible Settings for Bits 1 and 2, Status Bits for DPR Communication

x4 Status bit for the active bus capabilities of the fieldbus slaves (FB Slave Ready)

This bit is monitored for the Fieldbus Error Reaction. Whenever this bit goes to 0 after a fieldbus card was initially found by the PPC-R, the selected Error Reaction (system shutdown, error message, or ignore) is initiated. Refer to Fieldbus/PLC Cyclic Read/Write Monitoring on page 7-108 for an explanation of the Fieldbus Error Reaction setting.

0--> The fieldbus slave is not (yet) ready for data exchange.

1--> The fieldbus slave can actively participate on the bus.

x5 Status bit for the non-cyclic channel (Parameter Channel) (Non-Cyc Ready)

0--> The non-cyclic channel (Parameter Channel) cannot (yet) be used.

1--> The non-cyclic channel (Parameter Channel) is ready for use by the fieldbus master.

x15 Status bit for the cyclic data output (Cyclic Data Valid):

0--> The cyclic data outputs (coming in to the PPC-R) are INVALID.

1--> The cyclic data outputs (coming in to the PPC-R) are VALID. The system looks for this bit to be 1 before allowing data transfer.

Fieldbus Diagnostics

VisualMotion Register 20 holds the information for "Fieldbus Diagnostics."

Table 7-50 below contains the bit assignment for fieldbus diagnostics. The assigned bits are labeled with "x" and the bit number in the second row. Unassigned bits are labeled with "---."

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
X16	x15	x14	x13	---	---	---	---	---	---	---	---	---	---	---	---

Table 7-50: Bit Assignment for VisualMotion Register 20

Bit Definitions

x13 - x16 Identification of the fieldbus interface card (FB Card Found)

The bit combinations for x13, x14 and x15 are as follows:

Bit 16	Bit 15	Bit 14	Bit 13	Fieldbus Type
0	0	0	0	<NO CARD>
0	0	0	1	<Not Defined>
0	0	1	0	Interbus
0	0	1	1	DeviceNet
0	1	0	0	Profibus
0	1	0	1	ControlNet
0	1	1	0	<Not Defined>
0	1	1	1	EtherNet/IP (10 or 100 MB)
1	1	1	1	Indramat PLC Interface

Table 7-51: Identification of the Fieldbus Interface

Fieldbus/PLC Cyclic Read/Write Monitoring

Monitoring of Fieldbus read/write capabilities to the cyclic channel are associated with three parameters:

- C-0-2611 Fieldbus/PLC Cyclic Channel: Current Number of Misses– displays the current number of transfers to/from the cyclic channel.
- C-0-2612 Fieldbus/PLC Cyclic Channel: Peak Number of Misses – displays the maximum number of missed transfers to/from the cyclic channel.
- C-0-2613 Fieldbus/PLC Cyclic Channel: Timeout Counter – displays the number of timeouts in the cyclic channel. If after 4 ms, the Cyclic Mapping Lists are not successfully transmitted, a "miss" is noted.

For more information about these parameters, refer to the *VisualMotion 11 Functional Description* manual.

Fieldbus Error Reaction

Note: The Fieldbus Error Reaction setting is active only in Sercos Phase 4. In all other Sercos phases, it will be inactive.

You can select how you would like the PPC-R system to react in case of a fieldbus error. This reaction can be set in the "Fieldbus Slave Configuration" window, using the combo box labeled "Fieldbus Error Reaction."

Three options are available for the Error Reaction setting. Depending on the selected setting, the value 0, 1, or 2 is stored in Parameter C-0-2635:

Setting	Value in Parameter C-0-2635
Shutdown	0 (default)
Warning Only	1
Ignore	2

Table 7-52: Parameter C-0-2635 Values for Error Reaction Settings

Fieldbus Mapper Timeout

The Fieldbus Mapper continually scans the system for sufficient resources to process the cyclic data mapping lists (2600 and 2601 lists). If 10 out of 10 consecutive attempts of the mapping list updates are incomplete, the system is considered to have insufficient resources and the selected error reaction is evoked, as follows:

If "Shutdown" (0) is set in Parameter C-0-2635, the following error is generated from the PPC-R card: **520 Fieldbus Mapper Timeout**

If "Warning Only" (1) is set in Parameter C-0-2635, the following error is generated: **209 Fieldbus Mapper Timeout**

If "Ignore" (2) is set in Parameter C-0-2635, the system will update as resources become available, but there is no way to monitor whether or not updates actually occur.

Lost Fieldbus Connection

Register 19, bit 4 indicates the status of the fieldbus. Refer to Fieldbus Status on page 7-107 for more specific bit information. The system monitors this bit and evokes the selected error reaction if the bit is low (0), after a fieldbus card is found. A typical situation that will cause this condition is the disconnection of the fieldbus cable from the fieldbus card.

If "Shutdown Control" (0) is set in Parameter C-0-2635, the following error is generated from the PPC-R (active in Sercos Phase 4 only):

519 Lost Fieldbus Connection

If "Warning Only" (1) is set in Parameter C-0-2635, the following error is generated (active in Sercos Phase 4 only):

208 Lost Fieldbus Connection

If "Ignore" (2) is set in Parameter C-0-2635, there is no noticeable reaction when Register 19 status bits go low, unless the GPP application program is customized to evoke a special reaction.

Troubleshooting Tip:

If a fieldbus card is not found on the system, the Error Reaction setting will be ignored. If you have a fieldbus card and the Error Reaction is not responding as expected, the system may not "see" your fieldbus card.

Information for the PLC Programmer

*.gsd File

Rexroth supplies a *.gsd file on the VisualMotion 11 CD containing supporting information for the PPC-R with a Profibus slave configuration. Contact a Rexroth technical representative for the location of this file.

Multiplexing

Primary Multiplex Method (for Consistent Masters only)

Important: You should not use the Primary Multiplex Method for a master that is not consistent over the entire cyclic channel. The Secondary Multiplex Method is available for inconsistent masters. Refer to *Secondary Multiplex Method (for Inconsistent Masters)* on page 7-113.

The advantage of the Primary Method is easier handling of input data for consistent masters.

Control Word and Status Word

Control Word

The control word is transferred in the multiplex channel from master to slave. It tells the slave in which index the data is being transferred from master to slave and in which index the data is requested from slave to master.

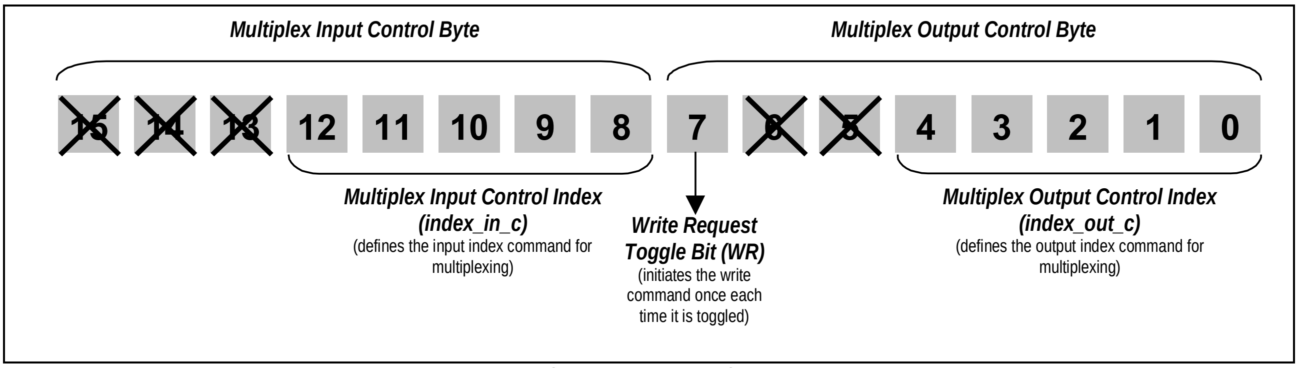


Fig. 7-108: Control Word Definition, Primary Multiplex Method

Index_out_c: tells the slave in which index the data are transferred from master to slave (out = master -> slave, _c = element of control word).

Index_in_c: tells the slave in which index the data is requested from slave to master (in = slave -> master, _c = element of control word).

WR (Write Request): handshake bit (refer to meaning of WR and WA).

Note: Input data via the Multiplex Channel is continually being updated.

Status Word

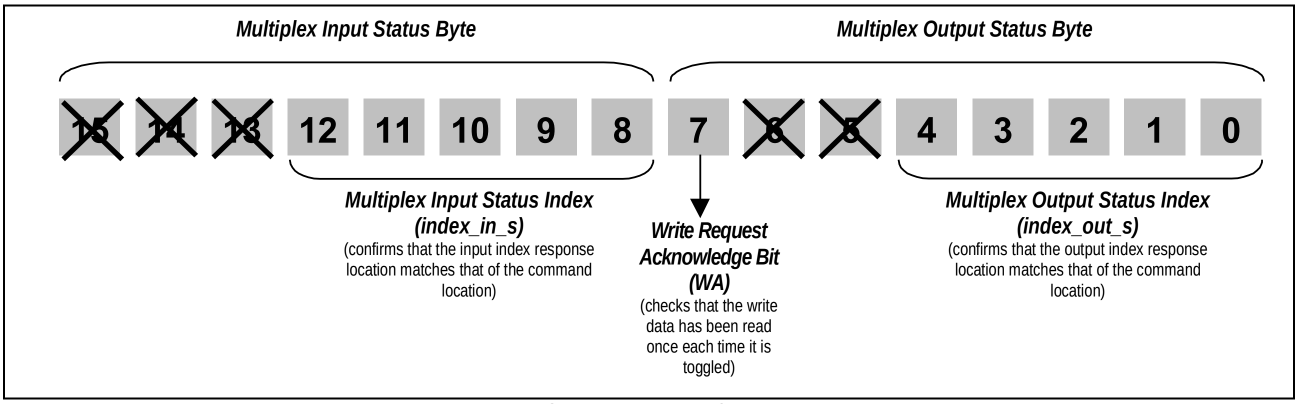


Fig. 7-109: Status Word Definition, Primary Multiplex Method

- **Index_out_s:** acknowledges index written by the master (out = master -> slave, _s = element of status word).
- **Index_in_s:** tells the master which index is transferred from slave to master in the actual process data cycle (in = slave -> master, _s = element of status word).
- **WA (Write Acknowledge):** Handshake bit (refer to meaning of WR and WA).

Handshake Bits WR and WA

WR and WA are handshake bits that allow the controlled writing of data via the multiplex channel. WR and WA control the data transfer for writing data_out (data send from master to slave).

WR == WA:

- tells the master that the slave has received the last multiplex data_out. The master can now send new data_out.
- tells the slave to do nothing, because the master has not yet put new consistent data_out on the bus.

WR! = WA:

- tells the slave to do something, because the master has now put consistent new data_out on bus.
- tells the master to do nothing, because the slave has not yet received the latest multiplex data_out.

Master Communications (Primary Multiplex Method)

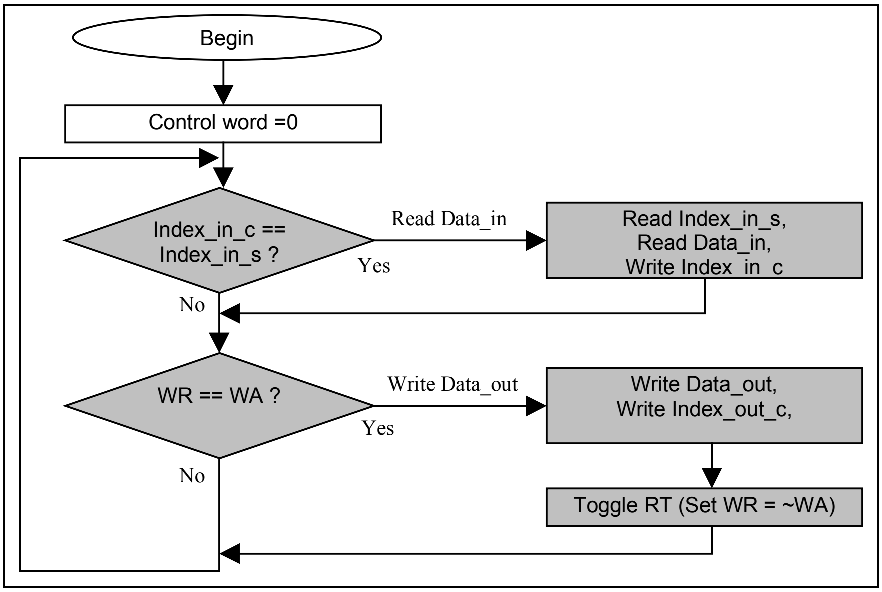


Fig. 7-110: Primary Multiplex Method, Master Communications

Programming Example

To aid in implementing the multiplex function in a PLC program, the following flow chart shows two ways of reading and writing data. Reading and writing can be executed separately, which allows the input data to be updated about 30% faster. The “Read Data” example would be placed at the beginning of a PLC program the “Write Data” example at the end.

Combined reading and writing makes the PLC program simpler, especially when using the same index for both transfer actions.

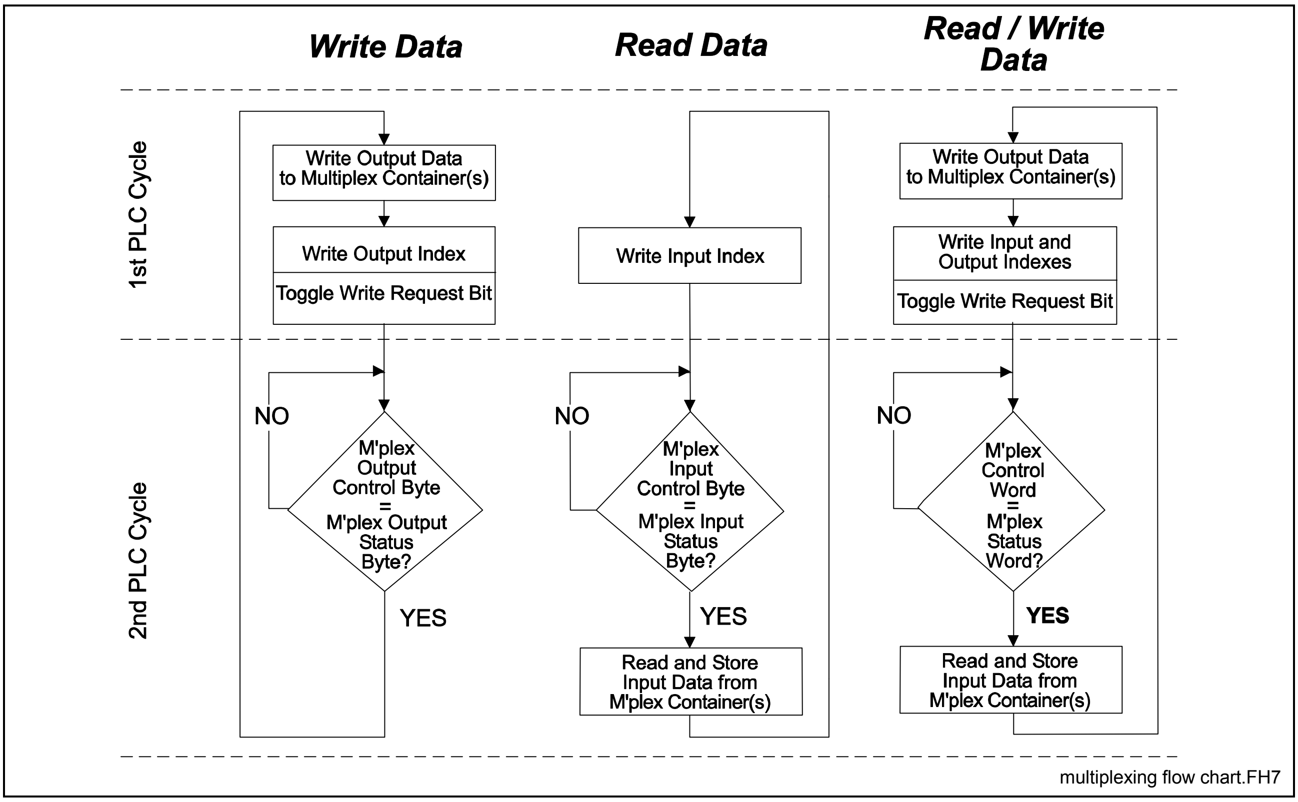


Fig. 7-111: Flow Chart of Multiplex Programming Examples (Primary Method)

Secondary Multiplex Method (for Inconsistent Masters)

Explanation of the Master Consistency Problem

The PPC-R fieldbus slave interfaces can guarantee consistency, however, some fieldbus masters can only guarantee byte, word, or double word consistency. Therefore, it is necessary to have a second multiplex method where both input data and output data require the handshake bits to update via the fieldbus.

Note: The meanings of the control and status words are the same as for the Primary Multiplex Method. The only difference is the toggle bits RR and RA, which are used in the Secondary Method.

Fig. 7-112 below illustrates the control word definition for the Secondary Multiplex Method.

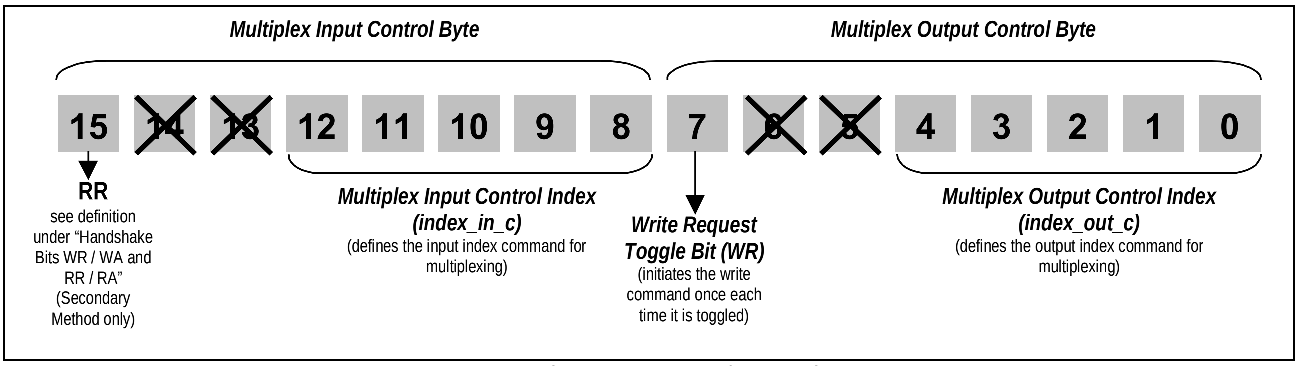


Fig. 7-112: Control Word Definition, Secondary Multiplex Method

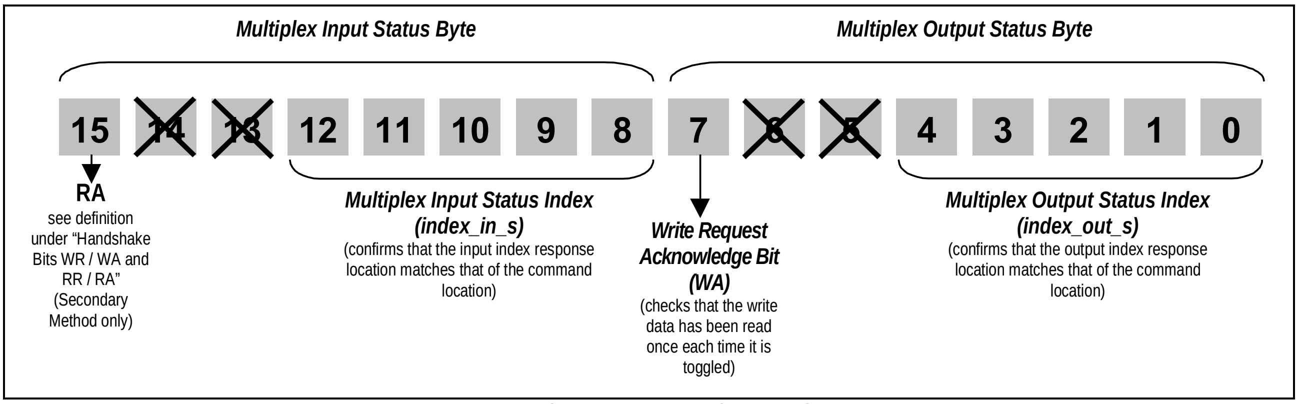


Fig. 7-113: Status Word Definition, Secondary Multiplex Method

The Secondary Multiplex Method has the following features:

- You can transfer a different index from master to slave as from slave to master.
- The handshake bits for both reading and writing of this multiplex channel make the multiplexing possible on inconsistent systems (masters).

Handshake Bits RR and RA

RR (Read Request) and RA (Read Acknowledge) are handshake bits that allow a controlled data transfer and use of the multiplex channel on inconsistent masters. RR and RA control the data transfer for reading data_in (data send from slave to master).

RR == RA:

- tells the master that the slave has sent the requested data_in. The master can now read the data_in and request new data_in.
- tells the slave to do nothing, because the master has not yet put new consistent data on the bus.

RR != RA:

- tells the slave to put new data_in on the bus, because the master requests new data_in.
- tells the master to do nothing, because the slave has not yet put the latest requested multiplex data_in on the bus.

Master Communications (Secondary Multiplex Method)

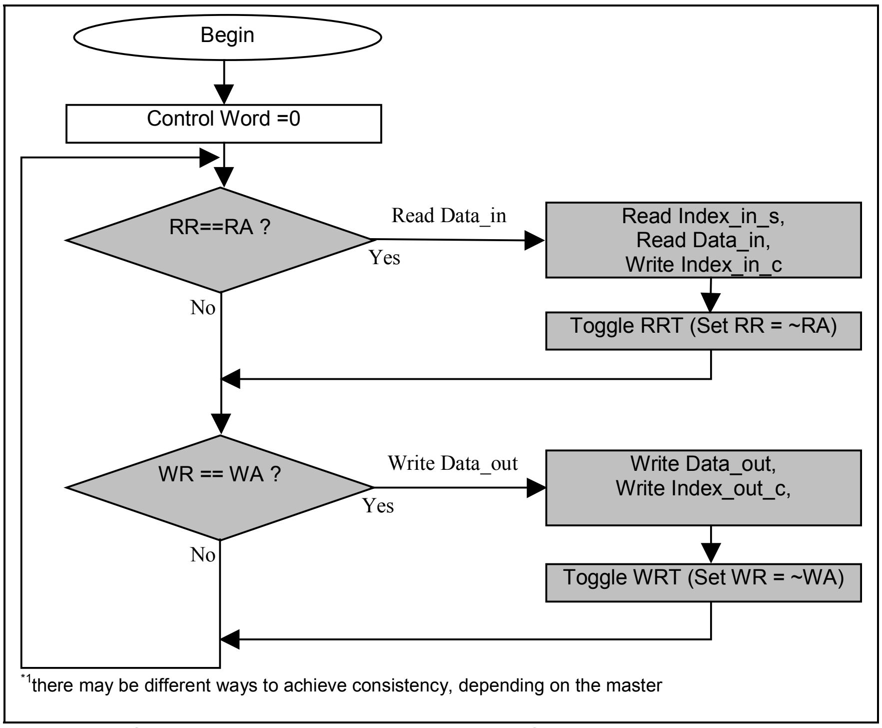


Fig. 7-114: Secondary Multiplex Method, Master Communications

For some masters, it could be enough to first write data and then the control word. For other masters, you may have to implement a delay time (this time could be different from master to master) before writing WR = ~WA.

Non-Cyclic Data Access via the Parameter Channel

Important: The fieldbus master's access of the cyclic channel must be consistent over the entire length of the assigned Parameter Channel in order to establish reliable Parameter Channel communications.

To support the configuration of drives and the access to parameters through the Profibus DP channel, Rexroth has established the Parameter Channel.

If the Parameter Channel is used with the PPC-R, the first 2, 4 or 6 data words of the cyclic channel for the slave board must be allocated for non-cyclic transmissions.

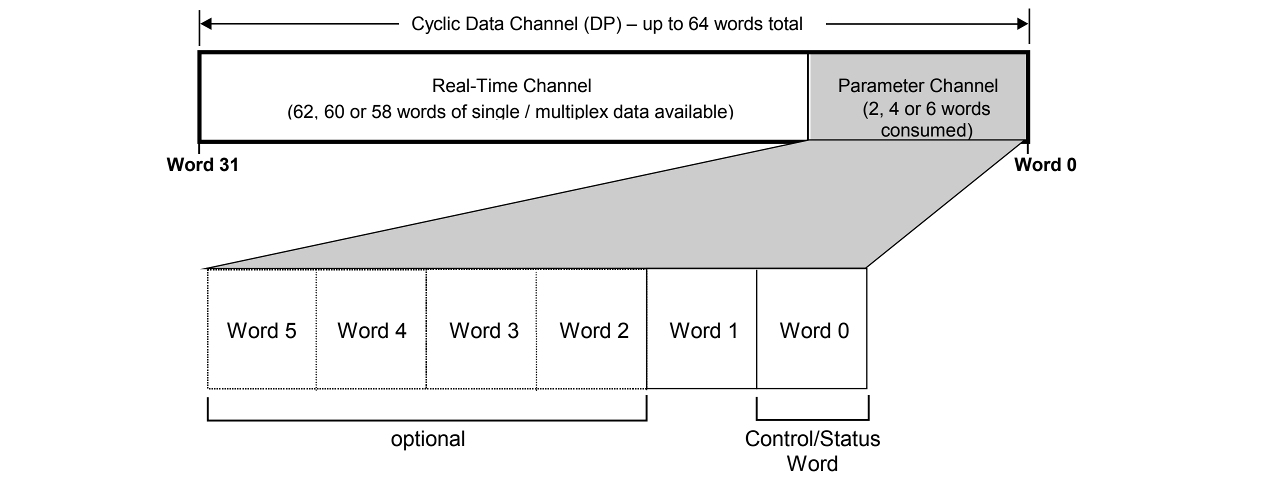


Fig. 7-115: The Parameter Channel inside the Profibus DP Channel

Messaging Formats

Two messaging formats are available in the Parameter Channel:

- **Short Format 3**
- **VisualMotion ASCII Format** - This format is provided for backward-compatibility with VisualMotion 6.0 / GPS firmware. For detailed information, refer to the VisualMotion 6.0 Startup Guide.

Short Format 3: General Explanation

To read or write a VisualMotion data type non-cyclically, a protocol is used inside the Parameter Channel. The protocol requires one word of the Parameter Channel for protocol functions. Thus, depending on the channel length 1, 3, or 5 data words can be transferred in one cycle. The protocol supports multiple transmissions, but the maximum length of data that can be transferred from or to an object is 128 bytes.

Short Format 3 Data Transfer

The following methods for transferring data are available in Short Format 3:

- Mapped Data
- Data Exchange Objects

Mapped Data

Mapped data is the most powerful feature of the PPC-R non-cyclic fieldbus interface. Through mapped data, the user has access to virtually every PPC-R data type over the fieldbus. It is easy to implement from the PLC side and requires no setup on the PPC-R side.

To access a data type over the fieldbus, it has to be specified by an address that consists of an index and a subindex. The index and subindex for each data type can be calculated by a formula (refer to *Accessing Mapped Data* on page 7-125).

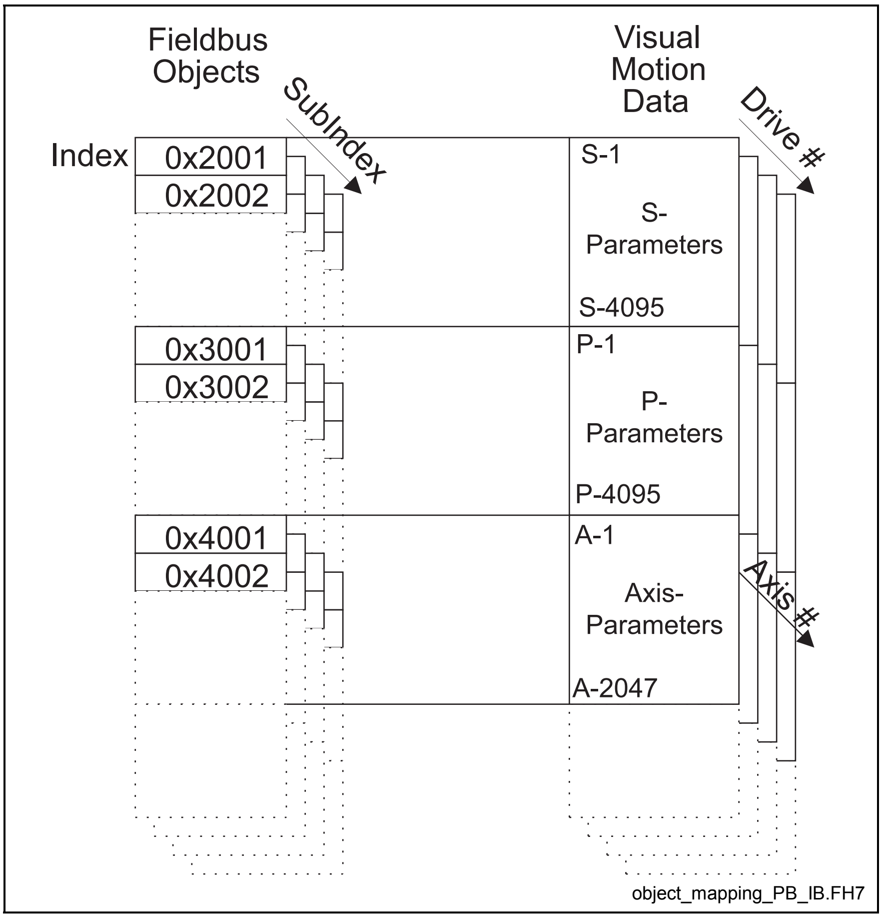


Fig. 7-116: Mapped Data

Mapped data can be used with the following parameters and values:

- S-Parameters (SERCOS Drive S-Parameters)
- P-Parameters (SERCOS Drive P-Parameters)
- A-Parameters (PPC Axis Parameters)
- C-Parameters (PPC Control parameters)
- T-Parameters (PPC Task parameters)

size and format depend on parameter *,1

PF-Values (PPC Program Float data, 32 bit – 2 words, IEEE format) *,2

GI-Values (PPC Global Integer data, 32 bit – 2 words) *,2

GF-Values (PPC Global Float data, 32 bit – 2 words, IEEE format) *,2

PI-Values (PPC Program Integer data, 32 bit – 2 words) *,2

Reg.-Values (PPC Register data, 16 bit – 1 word) *,3

Data Exchange Objects (0x5E70 – 0x5E73) (embedded ASCII Protocol)

*You may notice that parameters accessed via the non-cyclic (Parameter) channel are not always the same size as reported from the attribute field. This is so that the data sizes correspond with the way the different data types are handled in the cyclic channel (Registers are always set to 16-bit size and Parameters are cast to 32-bit size, even if they actually use less space).

1. When **writing** mapped data to a VisualMotion Parameter, you must send the size data corresponding to that of the attribute field within the parameter.
   - a.) For 32-bit parameters, you must send a data size of 32 bits (otherwise, VM error #07 is returned).
   - b.) For 16-bit parameters, you must send a data of size 16-bits. If, for this case, you send data of size 32 bits, one of the following occurs:
     - i.) For parameters of type 16-bit unsigned, only the Low word is stored, and the High word is ignored.
     - ii.) For parameters of type 16-bit signed, bits 0-14 of the low word along with the sign bit #31 are used, and the remaining bits are ignored.
   - c.) For String Parameters (e.g. S-0-0142), you must send the size of the string to write.

d.) All other Parameter Types (list parameters, command parameters, etc), are not supported for mapped data.

When reading mapped data from a VisualMotion Parameter, there are 3 possible cases of sizes returned:

a.) If the parameter type is a string, you receive the number of bytes corresponding to the length of the string.

b.) If the parameter is 32-bit or less, you receive a cast 32-bit value for this parameter. This implies that 16-bit parameters are returned as cast in to 32-bit values.

c.) All other parameter types (e.g. list parameters, command parameters, etc.), are not supported for mapped data.

2. When writing mapped data to a VisualMotion Program Float, Program Integer, Global Float, or Global Integer, the data size must be 32-bits (2 words). Any other size returns a VM error #07 (Invalid Data Format).

   When reading mapped data from a VisualMotion Program Float, Program Integer, Global Float, or Global Integer, the data size returned is always 32-bit (2 words).

3. When writing mapped data to a VisualMotion Register, the data must be 16-bits (1 word). Any other size returns a VM error #07 (Invalid Data Format).

   When reading mapped data from a VisualMotion Register, the data size returned is always 16-bit (1 word).

User Data Header – Object Index

The index refers to the particular fieldbus slave object that a VisualMotion data type is (automatically) mapped. This object allows for simple, indirect access to VisualMotion data types, and it is combined with the subindex to create a direct relationship to the VisualMotion data types. The available objects can be calculated using the formulas in *Accessing Mapped Data* on page 7-125.

User Data Header – Object SubIndex

The subindex refers to an additional piece of information necessary to obtain direct access to VisualMotion data types. The reference of the subindex depends on the data type in question. For example, the SubIndex refers to the drive number when accessing S and P parameters. However, the subindex refers to the task number when referring to task parameters. The available subindex ranges can be calculated using the formulas in *Accessing Mapped Data* on page 7-125.

Data Exchange Objects

The four data exchange objects 5E70 to 5E73 represent fixed data "containers" of varying lengths that transfer the VisualMotion ASCII Protocol to the PPC-R card. These objects serve as an open-ended possibility to access any VisualMotion data (including cams, diagnostic text, etc.), but more work is required in the master to perform a transmission of this type. Both the VisualMotion ASCII message and the fieldbus transfer message must be formulated.

Table 7-53 lists the available data exchange objects and their sizes.

Data Exchange Object	Data Length (in bytes)
5E70	16
5E71	32
5E72	64
5E73	128

Table 7-53: Length of the Data Exchange Objects

Rexroth
Bosch Group

Short Format 3 Parameter Channel (PK) Control and Status Words

PK Control Word

The PK control word is sent from the master to the slave. It is 16 bits wide and the individual bits have the following meanings:

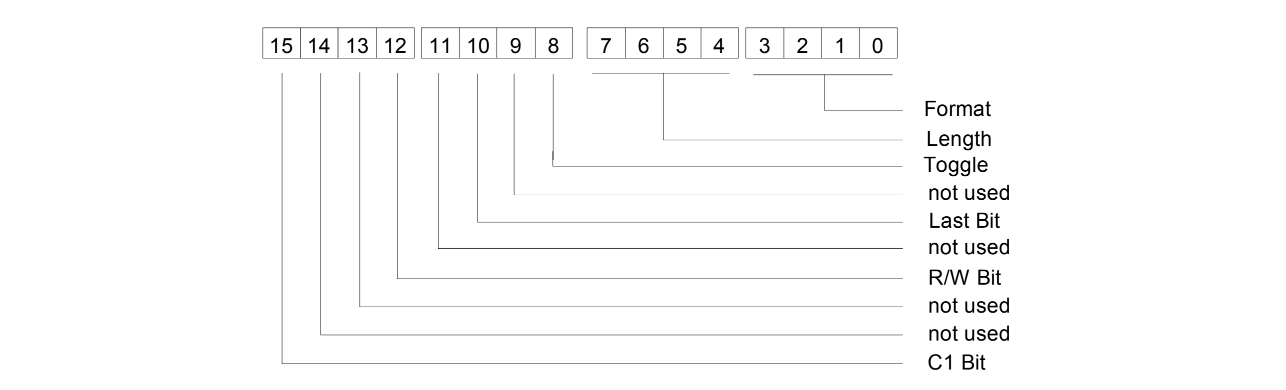


Fig. 7-117: Bits of the PK Control Word

Format: These bits describe the usage and meaning of the following data words in the Parameter Channel. Their value is fixed to 1100_b.

Length: These four bits specify the length of the valid data in bytes, without the control word. The data in the rest of the Parameter Channel is undefined.

Toggle: This bit toggles with every new set of sent data. It is used for a handshake between master and slave. The master is only allowed to toggle this bit when the toggle bit in the status word has the same level as the toggle bit sent in the control word.

L: Last bit. This bit is set when the last fragment of a data block is sent.

R/W: Read/Write; Read = 1, indicates that the master wants to read data.

C1: This bit is used to distinguish between the “old” and “new” handling of the Parameter Channel. For the “new” handling (e.g. Short Format 3), it is fixed to 1

Note: Bits that are not used are set to 0.

PK Status Word

The PK status word is sent as an answer from the slave to the master. The 16 bits have the following meanings:

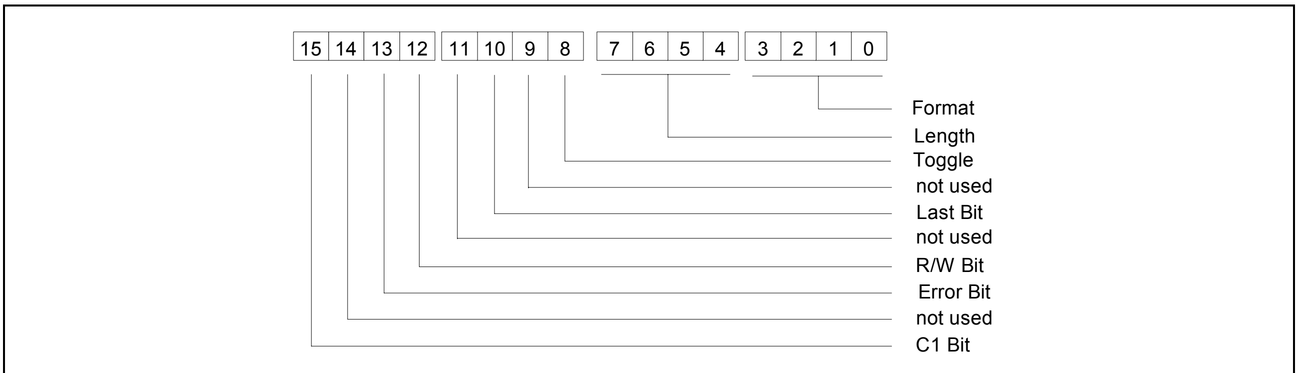


Fig. 7-118: Bits of the PK Status Word

Format: These bits describe the usage and meaning of the following data words in the Parameter Channel. Their value is fixed to 1100_b.

Length: These four bits specify the length of the valid data in bytes, without the status word. The data in the rest of the Parameter Channel is undefined.

Toggle: This bit toggles with every new set of sent data. It is used for a handshake between master and slave. The slave recognizes new data when the toggle bit it receives (control word) is different from the toggle bit in the status word.

L: Last bit. This bit is set when the last fragment of a data block is sent.

R/W: Read/Write Acknowledgement; Read = 1, indicates that the master wants to read data.

Error Bit: This bit indicates an error that occurred within the slave. The reason for the error is coded in the following data.

C1: This bit is used to distinguish between the “old” and “new” handling of the Parameter Channel. For the “new” handling (Short Format 3), it is fixed to 1

Note: Bits that are not used are set to 0.

Short Format 3: Examples

The following examples show how to write and read an object. They display the read and write access of object index 2001_h, subindex 2_h. The matching Visual Motion data according to the chart at the end of this chapter is S-Parameter 1 of Drive 2.

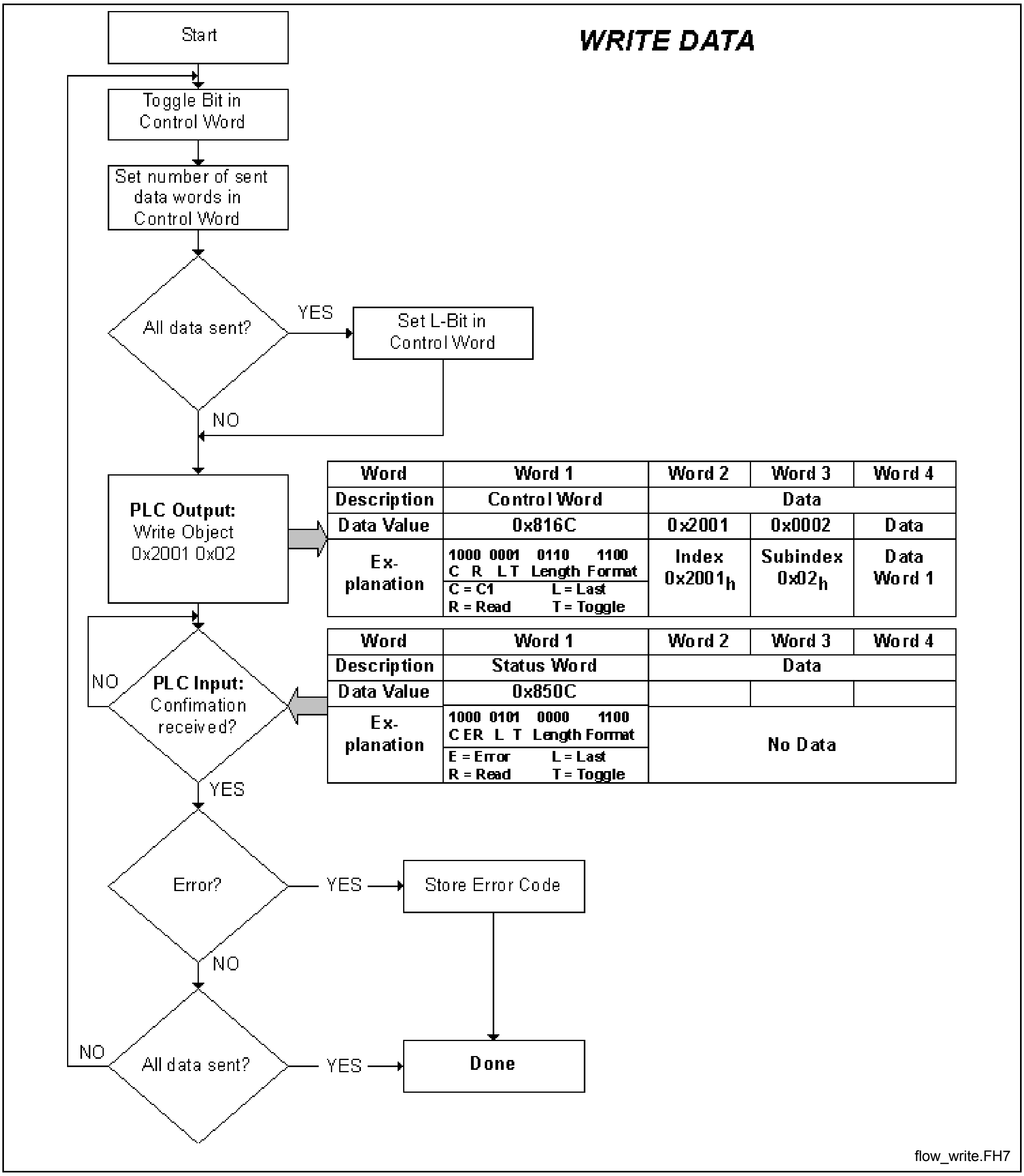
Notes for the following examples:

These flow charts assume a toggle bit value of 0 when starting. The values of the control and status words can change because of different states of toggle bit and last bit.

The master can detect new data comparing its own toggle bit with the toggle bit received from the slave. If they match, new data was received from the slave.

When writing, only the first telegram from the master contains the index and subindex.

Rexroth Bosch Group



Word	Word 1	Word 2	Word 3	Word 4
Description	Control Word	Data		
Data Value	0x816C	0x2001	0x0002	Data
Ex-planation	1000 0001 0110 1100 C R L T Length Format C = C1 L = Last R = Read T = Toggle	Index $0x2001_h$	Subindex $0x02_h$	Data Word 1

Word	Word 1	Word 2	Word 3	Word 4
Description	Status Word	Data		
Data Value	0x850C			
Ex-planation	1000 0101 0000 1100 C E R L T Length Format E = Error L = Last R = Read T = Toggle	No Data		

Fig. 7-119: Write Data Object Example

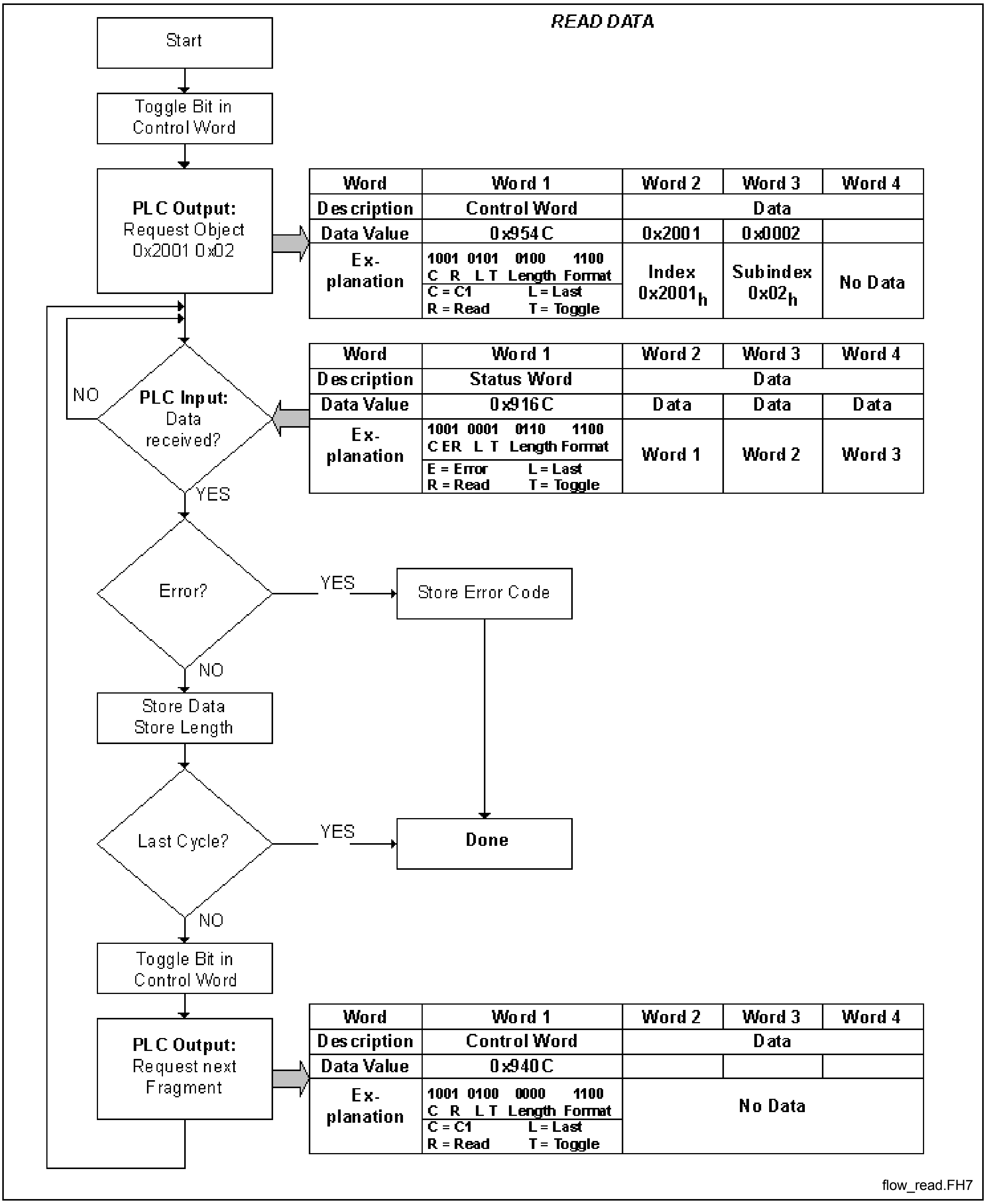


Fig. 7-120: Read Data Object Example

Rexroth Bosch Group

Canceling Data Transfer

In some cases, it might be necessary to cancel a data transfer. To request a communication reset, the master sends a cancel telegram to the slave.

Word	Word 1	Word 2	Word 3	Word 4
Description	Control Word	Data		
Data Value	0x810F			
Ex-planation	1000 0001 0000 1111 C ER L T Length Format E = Error L = Last R = Read T = Toggle	No Data		

cancel.FH7

Fig. 7-121: Cancel Telegram

The format byte in the command word is set to F_h. The length byte, the L and the R bits are set to 0.

The slave will set its internal state to expect a new command from the master.

Error Messages

If the transmission fails, the slave will respond with an error message as shown below. The status word value can be different for writing. The error bit in the status word is set and the first word contains a 16-bit error code. The toggle bit has the same state as the corresponding request telegram.

Word	Word 1	Word 2	Word 3	Word 4
Description	Status Word	Data		
Data Value	0xA42C	Error code		
Ex-planation	1011 0100 0010 1100 C ER L T Length Format E = Error L = Last R = Read T = Toggle		No Data	

error.FH7

Fig. 7-122: Error Response from Slave

The error code is two bytes long. The high byte specifies the error class and the low byte contains additional information for the application-specific errors (error class $1F_h$).

Parameter Channel Error Codes (High-Byte)

Error No. (Hex)	Error Description
0x1F	Control-specific error. Refer to Table 7-55 for additional error information, which is based on VisualMotion Communication Error Codes. Refer to the Troubleshooting Guide.
0x85	Data length too long (here >128 byte).
0x88	An error occurred during the transmission of data between the PPC-R and the fieldbus slave.
0x8B	Format (bits 0-3 of control word) specified is incorrect.
0x8C	The length set in control byte greater than Parameter Channel.
0x8D	Communication not possible. Parameter Channel too short (<2 bytes).
0x90	The format bits (0-3) of the control word were changed while transmitting several data blocks.
0x95	A read command was issued, but the length field was set to !=0.

Table 7-54: Parameter Channel Error Codes (High-Byte)

Parameter Channel Error Codes (Low-Byte) for 0x1F

Error No. (Hex)	Error Description
0xF3	Invalid Object Sub Index – Occurs when an attempt to access an incorrect or undefined location in the mapped data area or when attempting to address a sub index greater than 255.
0xF2	Invalid Object Index – Occurs when attempting to access an incorrect or undefined location in the mapped data area.
0xF1	Not used
0xF0	ASCII Format Error – occurs when attempting to communicate via the Data Exchange object where VisualMotion ASCII protocol is sent. This error also occurs if the initial characters are incorrect (such as the absence of the “>” start character).
< 0xF0	Error is based on VisualMotion Communication Error codes. Refer to the VisualMotion 11 Troubleshooting Guide for details.

Table 7-55: Parameter Channel Error Codes (Low-Byte)

Handling a Data Exchange Object

When mapped objects are not capable of transferring the desired data, a Data Exchange Object can be used.

The same procedures for writing and reading mapped objects via Short Format 3 apply to the Data Exchange Object.

Selecting a Data Exchange Object

Depending on the length of a VisualMotion ASCII message, any data exchange objects can be selected. The entire data length of the data exchange object, however, must always be transmitted even if the VisualMotion ASCII message is shorter. For example, if you want to transmit an ASCII message of 42 bytes, you must use object 5E72. To avoid a response error from the fieldbus slave, you must append 22 "Null" characters to the end of the ASCII message to complete a data size of 64 bytes.

Note: The checksum for the VisualMotion ASCII protocol is NOT used with the data exchange object. If the checksum is sent as part of the string, it will be ignored, and no checksum will be sent in the VisualMotion ASCII response messages. To ensure data integrity, the fieldbus protocols support a low-level checksum.

Transmission Sequence via a Data Exchange Object

For the data exchange object, two transmission sequences (and two response sequences) are required, to send the read or write message to and then receive the response message from the PPC-R card.

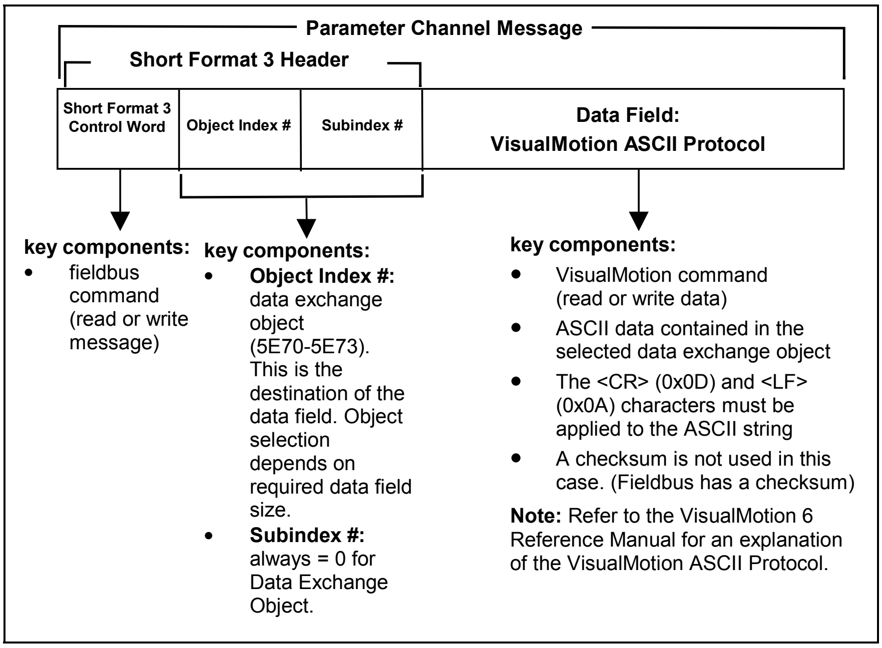


Fig. 7-123: Format of a PK Short Format 3 Message using a Data Exchange Object

The following sequence describes the communication between the fieldbus master (PLC) and the fieldbus slave (PPC-R). For details on reading and writing data in Short Format 3, refer to *Messaging Formats* on page 7-115.

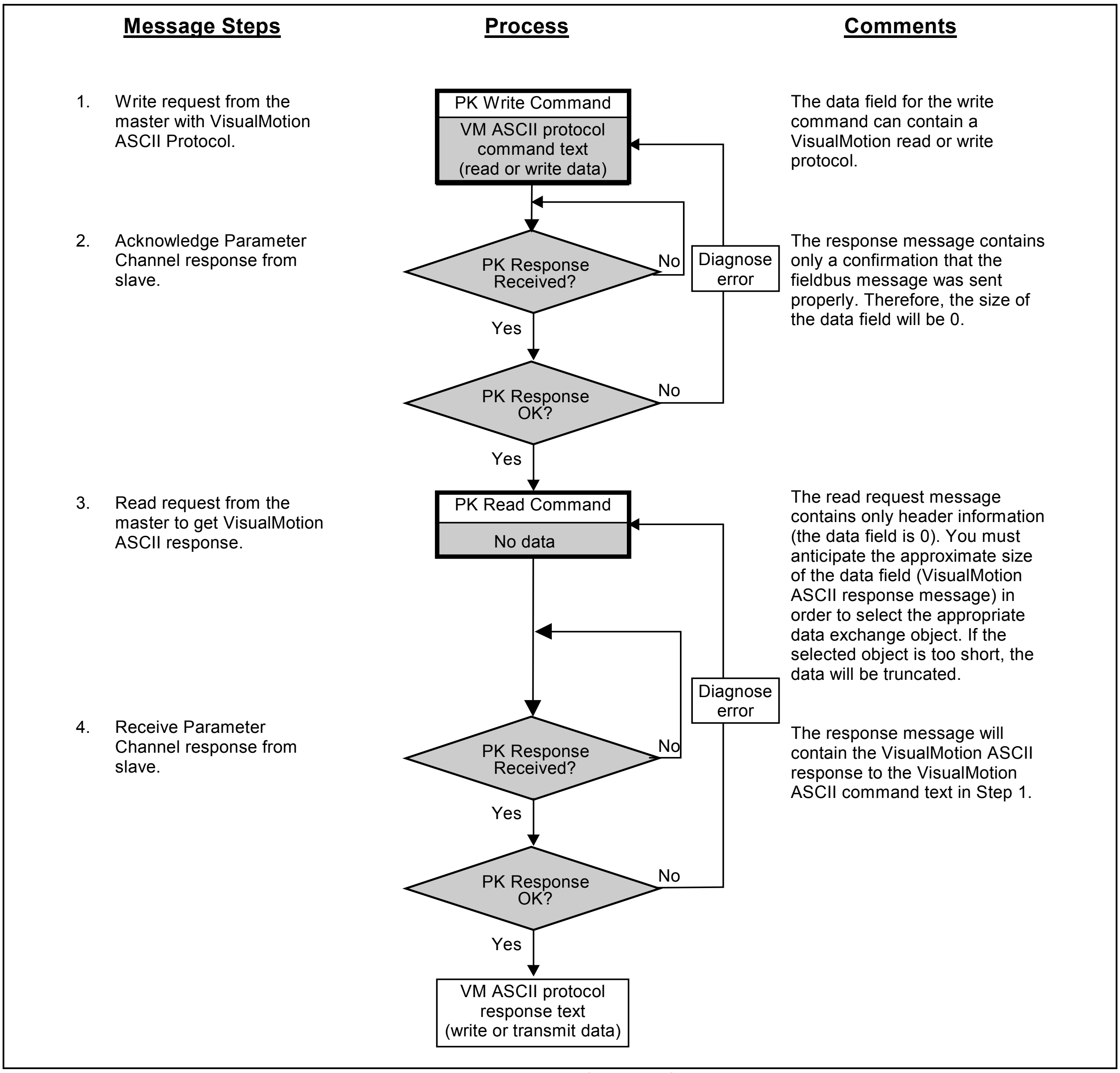


Fig. 7-124: Parameter Channel Short Format 3 Procedure, Using Data Exchange Object

Accessing Mapped Data

Rexroth has pre-configured a number of VisualMotion data types to Profibus indexes and subindexes. We call this concept **mapped data**. These data types can be accessed via the Profibus Parameter Channel. The index and subindex for each of these data types can be calculated using the formulas in *Table 7-56* below.

	Object Index #	SubIndex #	Formula
Data Exchange Object	0x5E73	0x00	
	----	----	
	0x5E70	0x00	
<FREE>	0x5E65	0xFF	
	----	----	(with SubIndex)
(349 objects available)	0x5D14	0x01	
Program Integers	0x5D13	0xFF	Index = 0x5D00 + [(Program Integer –1) \ 255]
	----	----	
(Int 1 – Int 5100)	0x5D00	0x01	SubIndex = Program Integer – [(Index – 0x5D00) * 255]
Program Floats	0x5CFF	0xFF	Index = 0x5CEC + [(Program Float –1)] \ 255]
	----	----	
(Float 1 – Float 5100)	0x5CEC	0x01	SubIndex = Program Float – [(Index – 0x5CEC) * 255]
<FREE>	0x5CEB	0xFF	
	----	----	(with SubIndex)
(235 objects available)	0x5C01	0x01	
Global Integers	0x5C00	0xFF	Index = 0x5BF7 + [(Global Integer – 1) \ 512]
	----	----	
(GInt 1 – GInt 2550*)	0x5BF7	0x01	SubIndex = Global Integer – [(Index – 0x5BF7) * 512]
Global Floats	0x5BF6	0xFF	Index = 0x5BED + [(Global Float – 1) \ 512]
	----	----	
(GFloat 1 – Gfloat 2550*)	0x5BED	0x01	SubIndex = Global Float – [(Index – 0x5BED) * 512]
<FREE>	0x5BEC	0xFF	
	----	----	(with SubIndex)
(245 objects available)	0x5AF8	0x01	
Registers	0x5AF7	0xFF	Index = 0x5AEE + [(Register – 1) \ 255]
	----	----	
(Reg. 1 – Reg. 2550**)	0x5AEE	0x01	SubIndex = Register – [(Index – 0x5AEE) * 255]
T-Parameters	0x5AED	0x04	Index = 0x56F1 + T-Parameter
	----	----	
(T-0-0001 – T-0-1020)	0x56F1	0x01	SubIndex = Task Number
<FREE>	0x56F0	0xFF	
	----	----	(with SubIndex)
(241 objects available)	0x5600	0x01	
C-Parameters	0x55FF	0x01	Index = 0x4800 + C-Parameter
	----	----	
(C-0-0001 - C-0-3583)	0x4801	0x01	SubIndex = 1
A-Parameters	0x47FF	0x63	Index = 0x4000 + A-Parameter
	----	----	
(A-0-0001 - A-0-2047)	0x4001	0x01	SubIndex = Axis Number
P-Parameters	0x3FFF	0x63	Index = 0x3000 + P-Parameter
	----	----	
(P-0-0001 - P-0-4095)	0x3001	0x01	SubIndex = Drive Number

	Object Index #	SubIndex #	Formula
S-Parameters	0x2FFF	0x63	Index = 0x2000 + S-Parameter
	----	----	
(S-0-0001 - S-0-4095)	0x2001	0x01	SubIndex = Drive Number
<Reserved>	0x1FFF	----	
	----	----	
	0x0000	----	

* current limitation: C-0-0080/C-0-0081 - Maximum number global integers/floats..

**current limitation: first 1024 registers.

Table 7-56: Formulas for Determining Mapped Objects

Example Lookup Tables for Mapped Objects

Card (C) Parameters

The following is an example lookup table for C-Parameters, when using mapped objects.

Example Look-up Chart for: **C-Parameters** CP 0.Y ==> CP = Card Parameter
Y = Parameter Number

Index

		0x4801	0x4802	0x4803	....	0x48FF	0x4900	....	0x55FE	0x55FF
SubIndex =	0x01	CP 0.1	CP 0.2	CP 0.3		CP 0.255	CP 0.256		CP 0.3582	CP 0.3583

Table 7-57: Mapped Object Lookup Table for C-Parameters

Axis(A) Parameters

The following is an example lookup table for A-Parameters, when using mapped objects. The same formula also applies to Sercos (S) and Task (T) Parameters.

Example Look-up Chart for: **A-Parameters** AP X.Y ==> AP = Axis Parameter
X = Axis Number
Y = Parameter Number

Index

		0x4001	0x4002	0x4003	....	0x40FF	0x4100	....	0x47FE	0x47FF
	0x01	AP 1.1	AP 1.2	AP 1.3		AP 1.255	AP 1.256		AP 1.2046	AP 1.2047
SubIndex =	0x02	AP 2.1	AP 2.2	AP 2.3		AP 2.255	AP 2.256		AP 2.2046	AP 2.2047
	0x03	AP 3.1	AP 3.2	AP 3.3		AP 3.255	AP 3.256		AP 3.2046	AP 3.2047
	:	:	:	:	:	:	:	:	:	:
	:	:	:	:	:	:	:	:	:	:
	0x28	AP 40.1	AP 40.2	AP 40.3		AP 40.255	AP 40.256		AP 40.2046	AP 40.2047

Table 7-58: Mapped Object Lookup Table for A-Parameters

Product-Specific (P) Parameters

The following is an example lookup table for P-Parameters, when using mapped objects.

Example Look-up Chart for: **P-Parameters** PP X.Y ==> PP = SERCOS P-Parameter (set 0 only)
X = Drive Number
Y = Parameter Number

Index = (Class ID && Instance ID for DeviceNet)

SubIndex = (Attribute ID for DNet)

	C118, In1	C118, In2	C118, In3	....	C118, In255	C119, In1	....	C134, In14	C134, In15
	0x3001	0x3002	0x3003	....	0x30FF	0x3100	....	0x3FFE	0x3FFF
0x01	PP 1.1	PP 1.2	PP 1.3		PP 1.255	PP 1.256		PP 1.4094	PP 1.4095
0x02	PP 2.1	PP 2.2	PP 2.3		PP 2.255	PP 2.256		PP 2.4094	PP 2.4095
0x03	PP 3.1	PP 3.2	PP 3.3		PP 3.255	PP 3.256		PP 3.4094	PP 3.4095
:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:
0x28	PP 40.1	PP 40.2	PP 40.3		PP 40.255	PP 40.256		PP 40.4094	PP 40.4095

Table 7-59: Mapped Object Lookup Table for P-Parameters

Integers

The following is an example lookup table for Integers, when using mapped objects. The same formula also applies to Floats, Global Integers, Global Floats and Registers.

Example Look-up Chart for: **VM Program Integers** PI 0.Y ==> PI = Program Integer
Y = Program Integer Number

Index

SubIndex =

	0x5D00	0x5D01	0x5D02	....	0x5D13
0x01	PI 1	PI 256	PI 511		PI 4846
0x02	PI 2	PI 257	PI 512		PI 4847
0x03	PI 3	PI 258	PI 513		PI 4848
:	:	:	:	:	:
:	:	:	:	:	:
0xFF	PI 255	PI 510	PI 765		PI 5100

Table 7-60: Mapped Object Lookup Table for Integers

7.7 DeviceNet, ControlNet, & EtherNet/IP Fieldbus Slave

General Information

Version Note:

Information in this document is based on VisualMotion Toolkit software version 11VRS and PPC-R firmware version GPP 11VRS (for DeviceNet, ControlNet, and EtherNet/IP). GMP 11VRS firmware does not have a fieldbus slave interface, but can be used with the Rexroth PPC-PCI bus interface to allow cyclic and non-cyclic data transfer.

Note: Fieldbus master devices are configured using IndraLogic, refer to Configuring a DeviceNet Fieldbus Master on page 7-38 for details.

VisualMotion 11 software is downward compatible with GPP firmware, but, depending on the hardware platform selected, the type of fieldbus communication selection may be limited. The following table lists the fieldbus firmware versions and the available fieldbus interfaces for each version.

<table>
<tr><th>Fieldbus Interfaces</th><th>PPC-R GPP07VRS</th><th>PPC-R GPP08VRS</th><th>PPC-R GPP09VRS/ GPP 11VRS</th><th>PPC-P GMP09VRS/ GMP 11VRS</th></tr>
<tr><td>DeviceNet</td><td>●</td><td>●</td><td>●</td><td rowspan="3">No Fieldbus Slave Support</td></tr>
<tr><td>ControlNet</td><td></td><td>●</td><td>●</td></tr>
<tr><td>EtherNet/IP 10 & 100 MB</td><td></td><td></td><td>●</td></tr>
</table>

Table 7-61: Fieldbus Firmware Version and Interface Type

Note: For detailed fieldbus hardware information, refer to the *VisualMotion 11 Project Planning Manual*.

ControlLogix with ControlNet or EtherNet/IP

The following configuration information is relevant for systems using ControlLogix with a ControlNet or EtherNet/IP fieldbus slave:

Instance	Details
Input	Instance (Transmitter ⇒ Originator) = 1 Size = FBMapper Master Input list size + 2 words (status information is not used)
Output	Instance (Originator ⇒ Transmitter) = 2 Size = FBMapper Master Output list size
Configuration	Instance is not relevant, use 4 Size = 0 The configuration instance is not used with ControlNet or EtherNet/IP fieldbus slave. Therefore, the size = 0 is the important part of the configuration.

Table 7-62: ControlLogix with ControlNet or EtherNet/IP

Rexroth Bosch Group

PPC-R System Description with a Fieldbus

The PPC-R can operate on a serial fieldbus interface (network) by means of a fieldbus expansion card that communicates with the PPC-R via dual-port RAM. The function of the fieldbus card, which is similar to that of a network card in a PC, allows communication with other devices on the network.

Fig. 7-125 illustrates the fieldbus slave and master interface in a system. In this document, we will refer to the PLC as the **fieldbus master** and the PPC-R as the **fieldbus slave**.

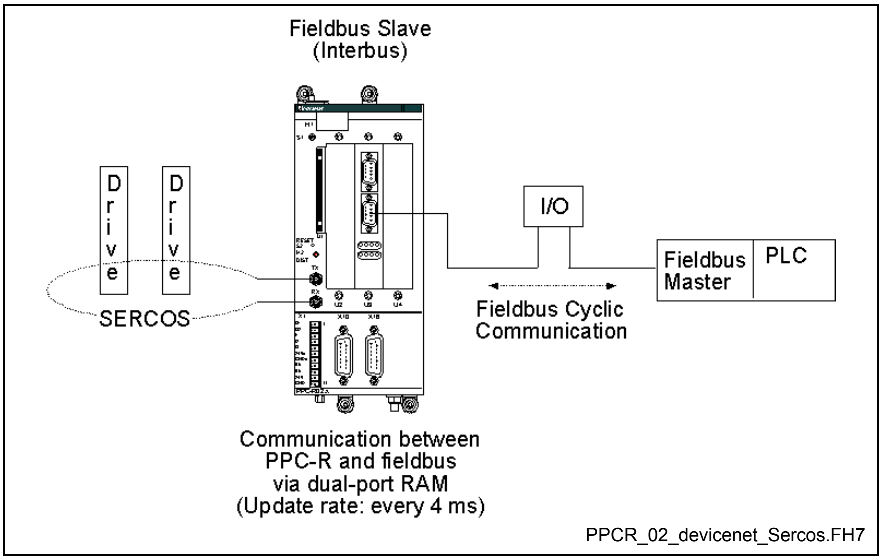


Fig. 7-125: Sample Master/Slave Setup with Fieldbus Card

With the PPC-R, the fieldbus card can be used **only** as a **slave** card in a master/slave setup.

Note: When using an EtherNet/IP type fieldbus card in a VisualMotion 11 system, no other fieldbus interface (i.e., Profibus, DeviceNet, ControlNet, Interbus) or MTS-R PLC interface can be used.

The onboard EtherNet and EtherNet option card both support standard TCP/IP and EtherNet/IP communication. When both EtherNet interfaces are installed, only the option card will support EtherNet/IP in addition to TCP/IP. Otherwise, a single EtherNet interface will support both TCP/IP and EtherNet/IP communication.

The VisualMotion Fieldbus Mapper

In the VisualMotion software package, the Fieldbus Mapper is a tool used to set up fieldbus configuration and data mapping.

Fieldbus hardware platform selections are made through the Fieldbus Mapper window with VisualMotion Toolkit in Service Mode. To select the Fieldbus hardware platform:

1. Open VisualMotion in Service Mode, indicated by the service mode symbol ( ) in the lower right corner of the VisualMotion Toolkit window.
2. Select ***Commission*** ⇒ ***Fieldbus Mapper*** to open the *FBMapper* window.

3. Click  **or** select ***File* ⇒ *New*** to open the *Fieldbus Slave Definition* window.

When a hardware platform is selected, only the fieldbus types available for that platform can be selected, refer to Table 7-61.

Data Transfer Direction (Output vs. Input)

In the VisualMotion Fieldbus Mapper, output and input are always described with respect to the fieldbus master. The definitions for output and input follow:

output: the communication from the PLC to the PPC-R (i.e. from the fieldbus master to the fieldbus slave).
Synonyms for this type of communication: **send** or **write** data.

input: the communication from the PPC-R to the PLC (i.e. from the fieldbus slave to the fieldbus master).
Synonyms for this type of communication: **receive** or **read** data.

Fieldbus Data Channel Descriptions

The Rexroth DeviceNet, ControlNet, and EtherNet/IP fieldbus interface cards for the PPC-R support the following communication channels:

- **Cyclic Channel:** Polled I/O (for single and multiplex channels)
- **Non-Cyclic Channel:** Explicit Messaging

Cyclic (Polled I/O) Channel

Cyclic data is user-defined. It is stored in two ordered lists (C-0-2600 for input data, C-0-2601 for output data) and transmitted serially over the bus. In the cyclic channel, data is updated cyclically between the fieldbus master and slave.

The cyclic data channel is limited to 64 input words and 64 output words. PPC-R data types consume these words in either one-word (or 16-bit) groups for PPC-R registers or two-word (or 32-bit) groups for all other data types.

The PPC-R mapping list is scanned every 4 ms and data is sent and received to/from the fieldbus slave board's dual port RAM.

The cyclic data channel can be made up of any combination of the following data types:

- Single
- Multiplex

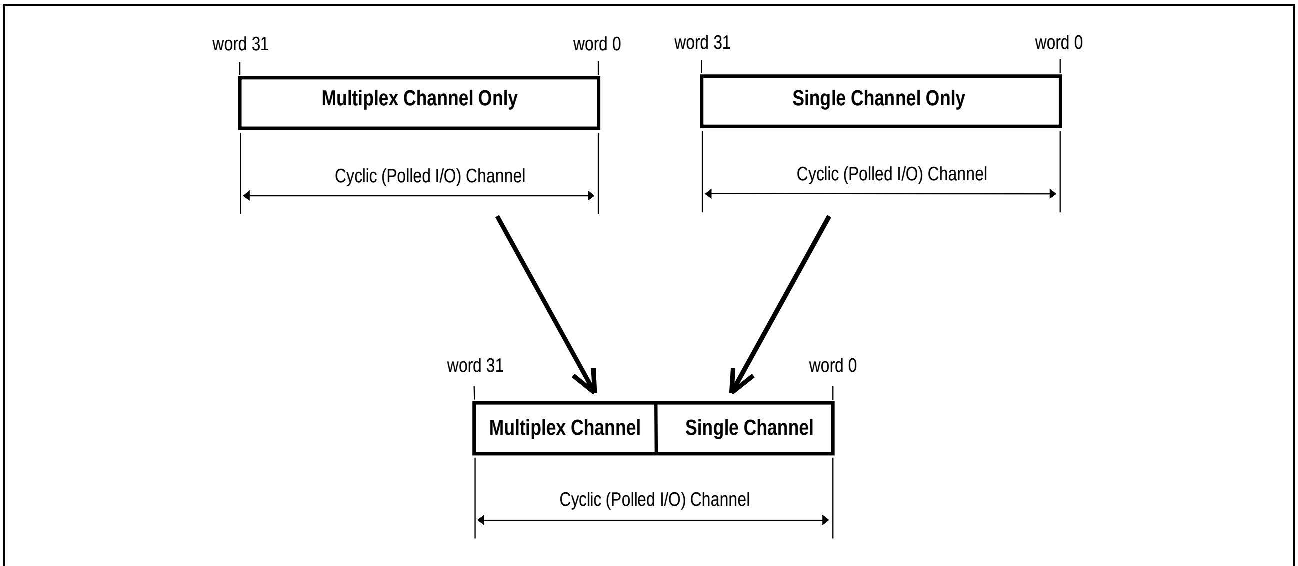


Fig. 7-126: Configuration Options for the Cyclic Data Channel

Rexroth Bosch Group

Cyclic Data: Types and Sizes

The following table outlines the PPC-R data types that can be transmitted via the cyclic channel and the amount of space (in 16-bit words) that each data type consumes.

Note: The cyclic data mapping lists support only 16- and 32-bit data of the following types for reading and writing:
- Integer
- Float
- Binary (used in PPC-R parameters)
- Hex (used in PPC-R parameters)

For all other data types (e.g. diagnostic messages - “strings”), use Explicit Messaging.

PPC-R Data Type	**Data Size (in 16-Bit Words)**
Register	1
Program Integer (currently active program ONLY *)	2
Program Float (currently active program ONLY *)	2
Global Integer	2
Global Float	2
Card Parameter	2
Axis Parameter	2
Task Parameter	2
Note: Drive parameters "S" or "P" cannot be transmitted cyclically because of the inherent delay of parameter access over the SERCOS service channel. However, if a drive parameter is mapped to an Axis Parameter, that Axis parameter could be used in cyclic data (see description of Axis Parameters 180-196 in the *VisualMotion Functional Description*).	
*** Important Note:** Integers and floats are shown only for the currently active program. Each time you activate a new progam, the fieldbus reads/writes to the newly-activated program.	

Table 7-63: PPC-R Cyclic Data Types and Sizes

Single Data Types

Single data types are mapped directly in the cyclic mapping ordered lists (C-0-2600, C-0-2601).

Multiplex Data Types Cyclic Data Channel)

(In some multi-axis applications, 64 words of cyclic data transfer are not sufficient to meet the requirement of the application.

When insufficient data transfer space is available, multiplex data can be set up within the cyclic channel. One multiplex container acts as a placeholder for multiple possible PPC-R data types (all of the same word size). The currently transmitted PPC-R data type is based on an index value placed in a multiplex control or status word attached to the end of the cyclic list. Depending on the index specified by the master, the multiplex channel permits a different set of data within the cyclic channel to be transferred as current real-time data. Multiplex containers can be added to the input and output lists separately and the input and output indexes can be designated separately (in the control and status words).

Note: Using the multiplex channel reduces the maximum number of usable words for storing PPC-R data to 63. The 64th word (or last used word, if fewer than 64 words) is used as the multiplex entry control/status word.

Note: When using VisualMotion 11 with GPP 7 firmware, a maximum of 15 multiplex containers and a maximum of 180 mapping items can be transmitted in the input or output list. This limitation of mapping objects means that you cannot multiplex all 15 containers with all 32 available indexes (=480 items).
For VisualMotion 11 with GPP 8 or 9 firmware, there is no limitation for multiplexing (each of the first 63 words may be multiplexed with up to 32 indexes).

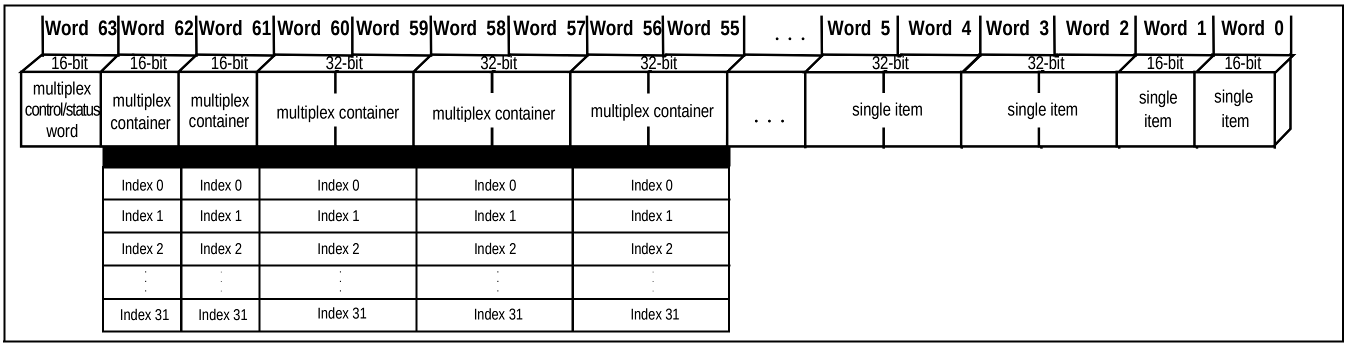


Fig. 7-127: Sample Command (PLC⇒PPC-R) or Response (PPC-R ⇒PLC)

Multiplex Control and Status Words

The multiplex control and status words serve to command and acknowledge multiplex data transferred between the fieldbus master and the fieldbus slave. The **control** word is associated with **output** communication (PLC⇒PPC-R). The **status** word is associated with **input** communication (PPC-R⇒PLC). Single data items are not affected by the multiplex control and status words.

Note: For specific information about how the fieldbus master uses the multiplex control and status words, refer to *Multiplexing* on page 7-149.

Rexroth
Bosch Group

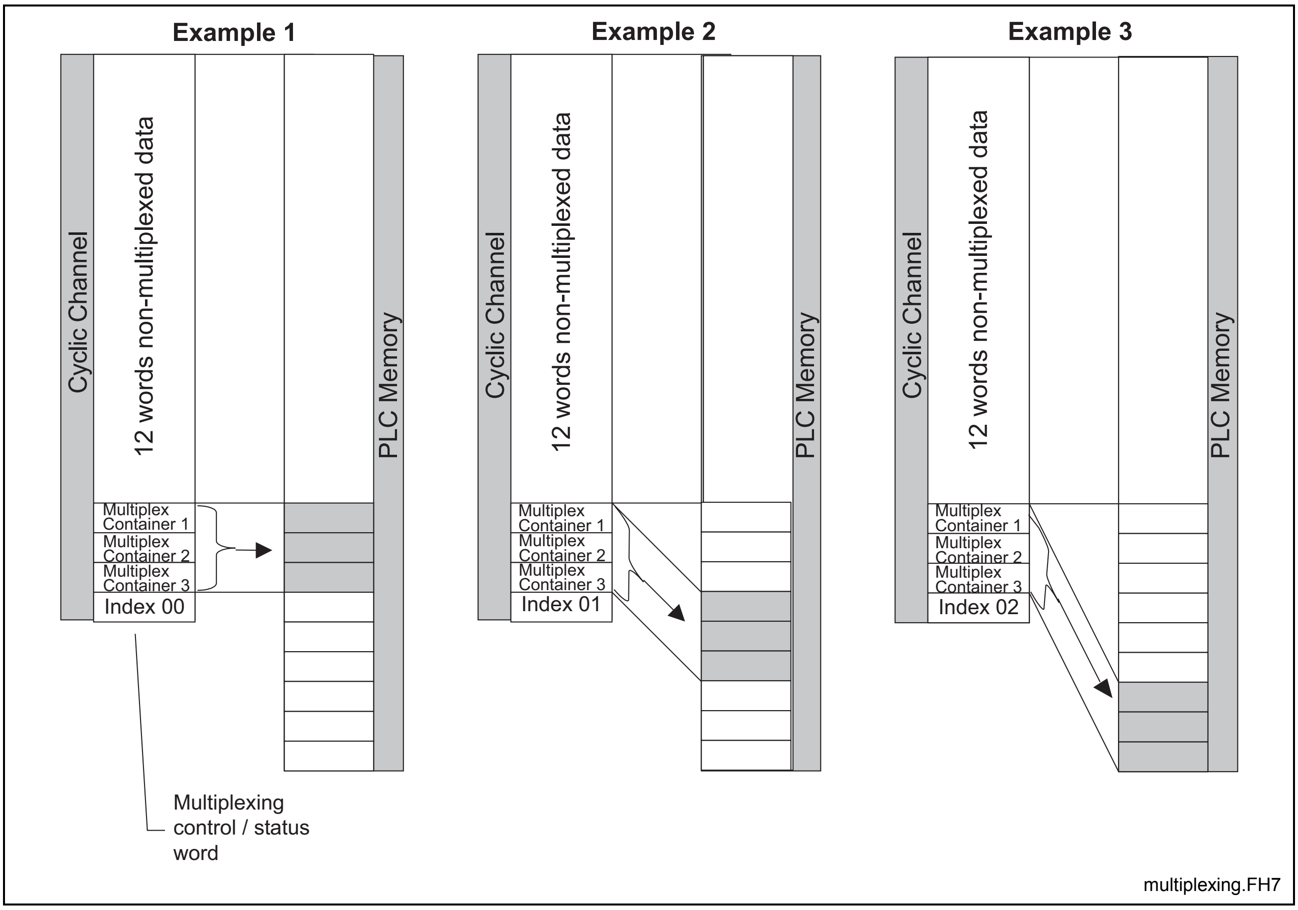


Fig. 7-128: Examples for Reading Data via the Multiplex Channel

Non-Cyclic Channel (Explicit Messaging)

The non-cyclic channel is used for data that needs to be transferred only once or sporadically, such as:

- the transmission of lists
- parameterization of axes or programs

Instead of being updated during each cycle, non-cyclic data is transferred using a command initiated by the master. Though any data type can be transferred non-cyclically, diagnostic messages and drive parameters (S and P) **must** be transferred non-cyclically because of the non-cyclic retrieval for drive parameters through Sercos and the length of the diagnostic messages.

There are two types of non-cyclic data transmissions for the PPC-R/VisualMotion system:

- mapped data (directly to PPC-R data types)
- data exchange object

Non-cyclic data can be accessed via Explicit Messaging support of the Fieldbus master.

Mapped Data

Mapped data is the most powerful feature of the PPC-R non-cyclic fieldbus interface. Through mapped data, the user has access to virtually every PPC-R parameter over the fieldbus. It is easy to implement from the PLC side and requires no setup on the PPC-R side.

Data Exchange Objects

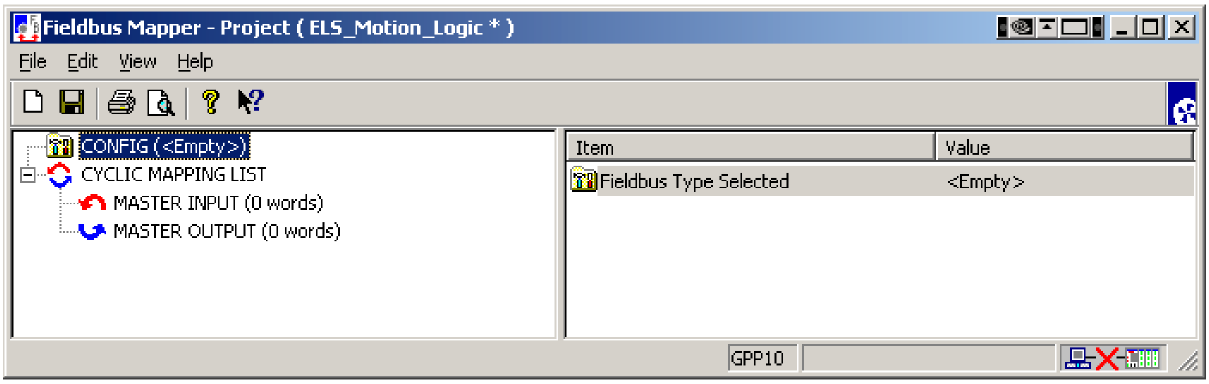
Four data exchange objects Class 100, Instance 1-4, Attribute 100 are available for the transfer of non-cyclic data. These objects represent fixed data "containers" of varying lengths that transfer the VisualMotion ASCII Protocol to the PPC-R card, in the same way that data is transferred using the VisualMotion ASCII Format via an Explicit Message. These objects serve as an open-ended possibility to access any VisualMotion data (including cams, diagnostic text, etc.), but more work is required in the

master to perform a transmission of this type. For more specific information about these objects, refer to
Data Exchange Objects on page 7-157.

Fieldbus Mapper Functionality

Initializing the Fieldbus Mapper from VisualMotion 11

1. Open an existing program or create a new program. You must be using PPC-R hardware with GPP firmware to use the Fieldbus Mapper described in this document.
2. Select ***Commission*** ⇒ ***Fieldbus Mapper***. The main Fieldbus Mapper window is displayed (refer to Fig. 7-129 below).



FB_Mapper_New.tif

Fig. 7-129: FBMapper – Project Window

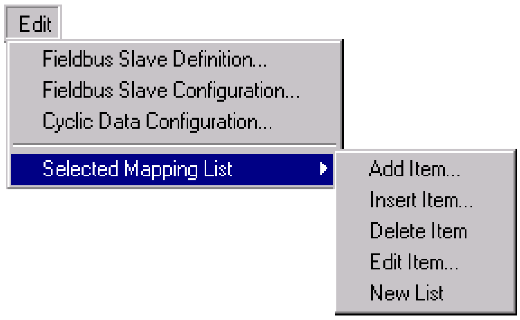
Creating a New Fieldbus Mapper File

To create a Fieldbus Mapper file:

1. Click **or select *File* ⇒ *New***.
   A “setup wizard” goes through three steps:
   – Fieldbus Slave Definition
   – Fieldbus Slave Configuration
   – Cyclic Data Configuration
2. Enter the information requested in the setup windows. For more details on each step, refer to *Fieldbus Slave Definition*, *Fieldbus Slave Configuration*, and *Cyclic Data Configuration* for detailed information about each configuration step.
3. Save the file (automatically has a **.prm** extension).

To Add/Insert, Edit, or Delete an item in a file, open the ***Selected Mapping List*** menu from the Edit menu (refer to Fig. 7-130 below). For more information about each step, refer to Fieldbus Slave Definition, Fieldbus Slave Configuration, and Cyclic Data Configuration for detailed information about each configuration step.

Rexroth Bosch Group



FB_Mapper_Edit_Menu.tif

Fig. 7-130: Fieldbus Mapper Edit Menu

Note: You can also directly add, insert, delete, edit an item, or create a new list by:

- clicking on the item to be edited in the main Fieldbus Mapper window and selecting the desired function under ***Edit ⇒ Selected Mapping List***

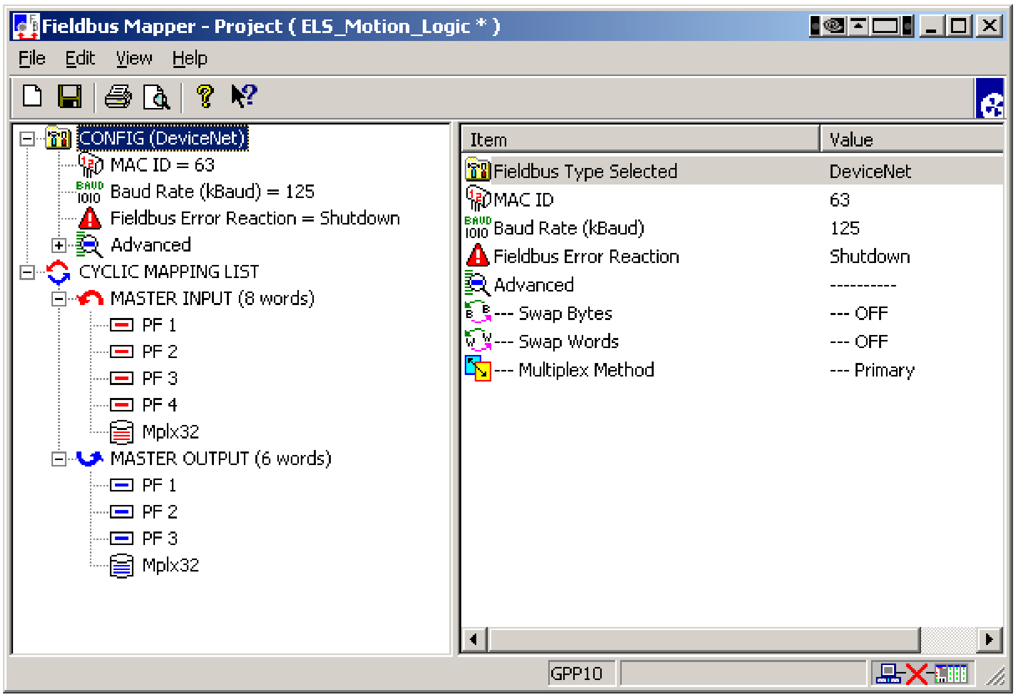
OR

- right-clicking on an item to display a menu of functions

Importing a Fieldbus Mapper File

A Fieldbus Mapper file can be imported from another project. To import the file:

1. Select ***File*** ⇒ ***Import***.
2. Browse to find the desired file (***.prm** extension).
3. Click ***Open***. The main Fieldbus Mapper window appears, which lists the configuration information. Refer to Fig. 7-131 below.

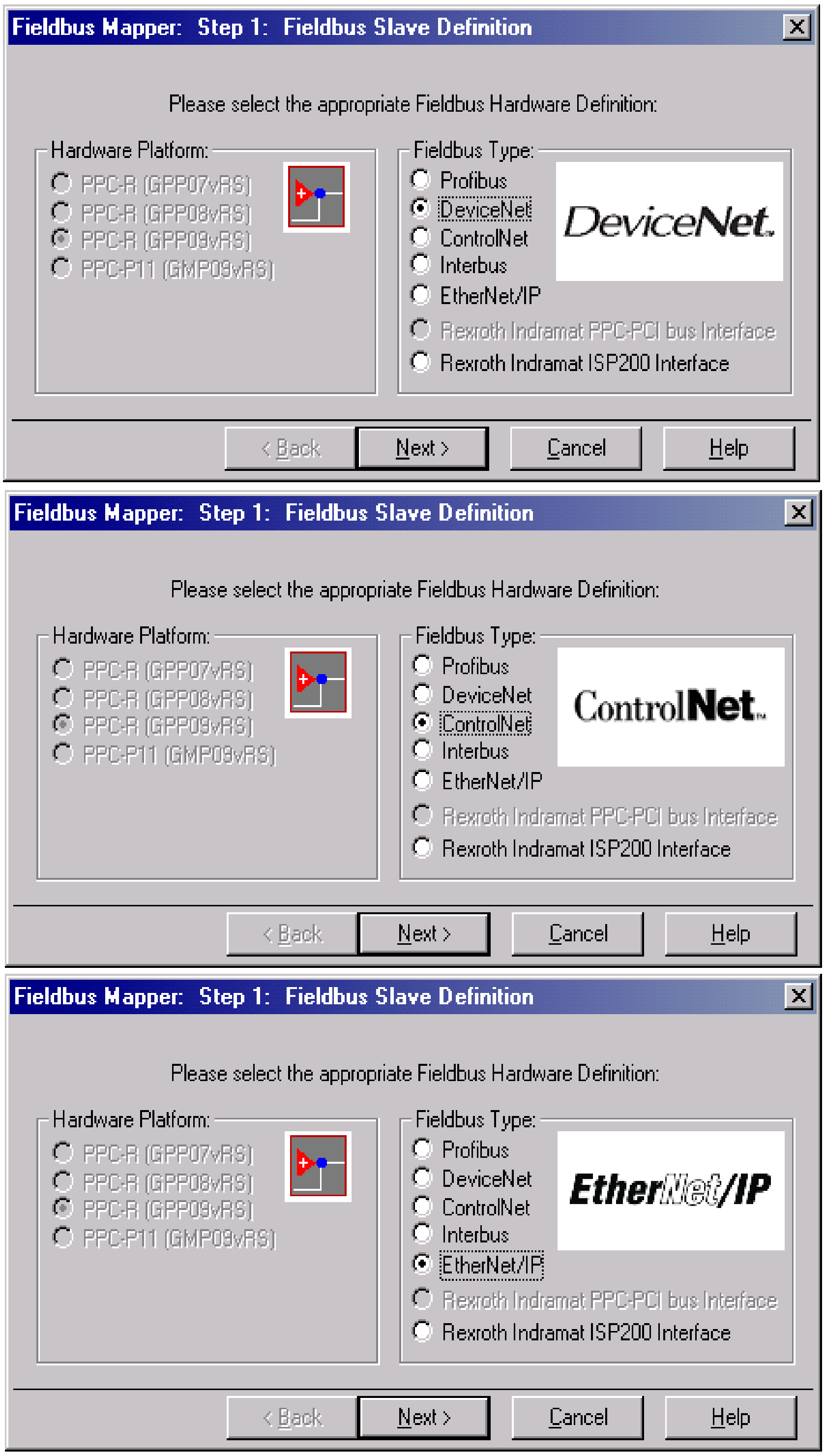


FB_Mapper_Main_DN.tif

Fig. 7-131: Fieldbus Mapper Main Window (Complete)

Fieldbus Slave Definition

From the Fieldbus Slave Definition window, select the desired Hardware Platform and **DeviceNet, ControlNet** or **EtherNet/IP** as the *Fieldbus Type* (refer to Fig. 7-132 below). Refer to Table 7-61 for a list of the available hardware platforms for the fieldbus types. The hardware platform can only be selected when the project is in Service mode.

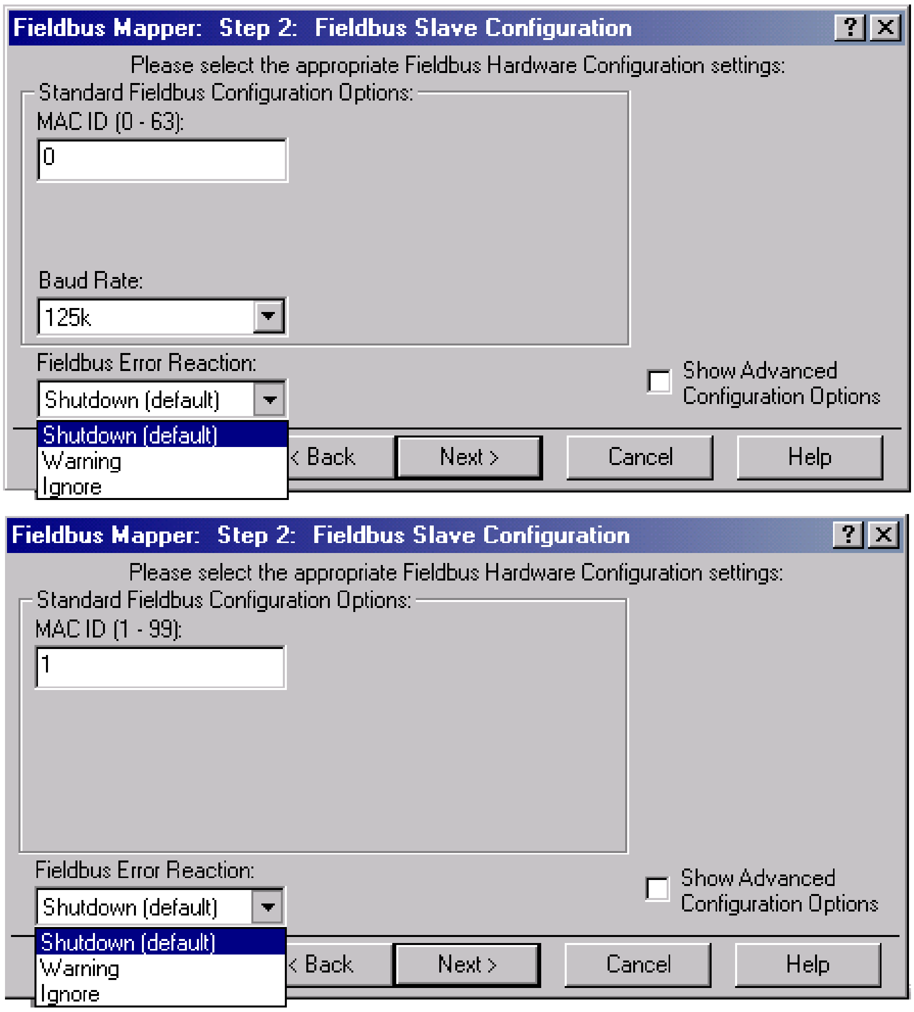


FB_Slave_Def.tif

Fig. 7-132: Fieldbus Slave Definition Window

Fieldbus Slave Configuration

The *Fieldbus Slave Configuration* windows for DeviceNet and ControlNet are shown in figure Fig. 7-133 below.



DeviceNet

ControlNet

FB_Slave_Config.tif

Fig. 7-133: Fieldbus Slave Configuration

Standard Fieldbus Configuration Options

- **MAC ID (0-63 for DeviceNet, 1-99 for ControlNet):** set to a unique number for this device on the bus.
- **Baud Rate (DeviceNet only):** set to match that of the master.

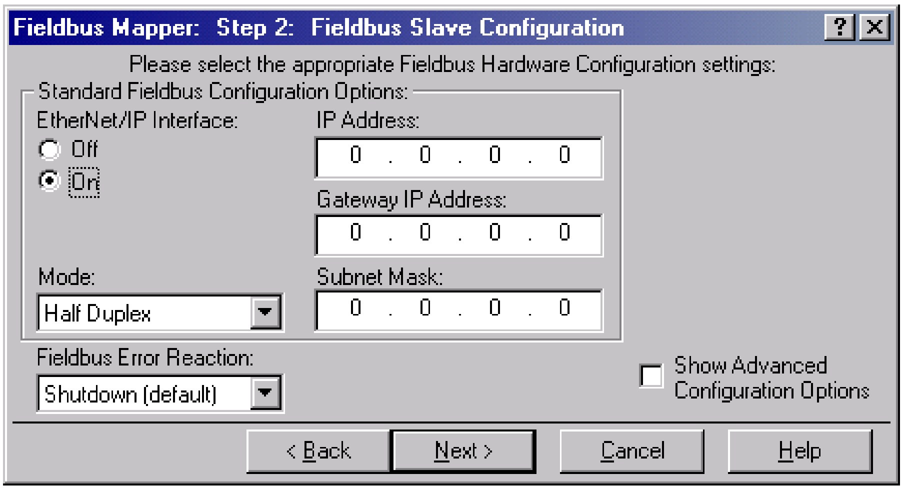
Fieldbus Error Reaction

Set the Error Reaction to Shutdown (default), Warning or Ignore. Refer to *Fieldbus Error Reaction* on page 7-147 for detailed information about each setting.

Advanced Configuration Options

The *Advanced Options:* are shown only if the checkbox next *to Show Advanced Configuration Options* is checked (refer to Fig. 7-135 below). In most cases, the default options should apply.

The *Fieldbus Slave Configuration* window for EtherNet/IP is shown below:



EtherNet/IP

EtherNetIP_IP_Address.tif

Fig. 7-134: Fieldbus Slave Configuration for EtherNet/IP

EtherNet/IP Interface

Selecting **On** for the EtherNet/IP interface will allow the control to distinguish between EtherNet/IP and standard EtherNet during communication when both are being used.

Mode

Specify whether the control is running in **Half Duplex** or **Full Duplex** mode.

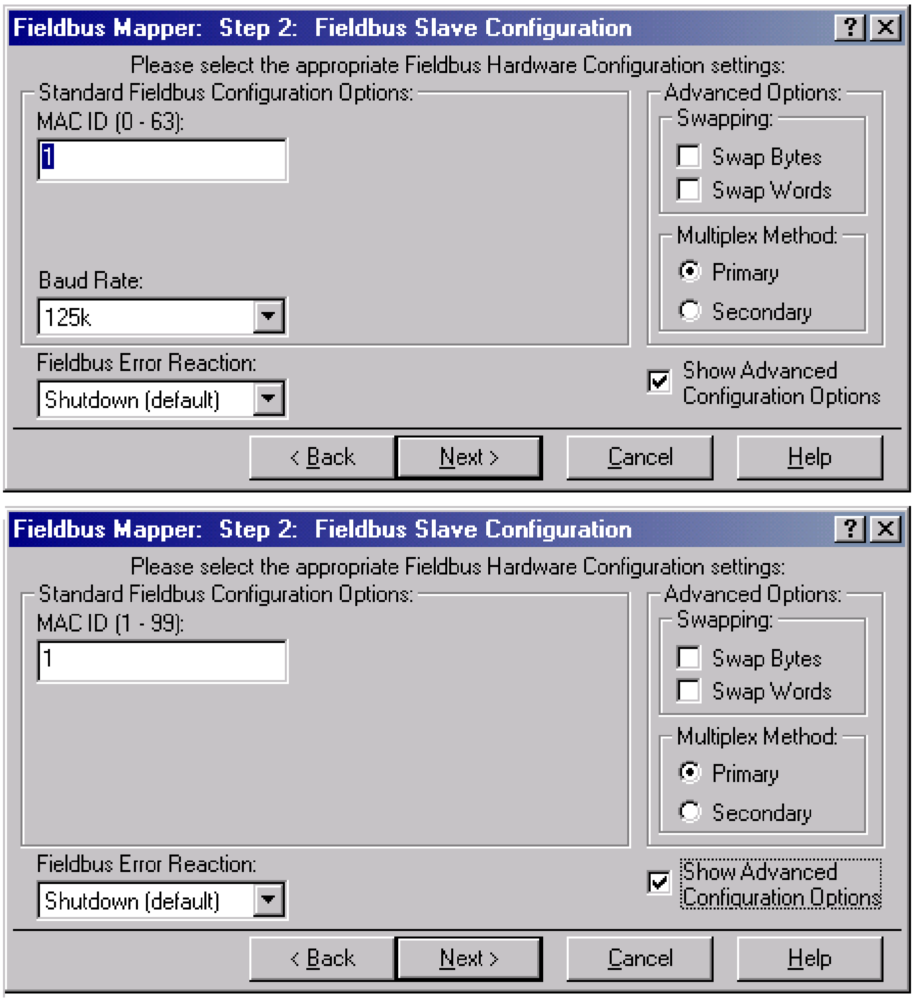
Note: Using an EtherNet switch and running in Full Duplex mode is recommended. Manual configuration of ports is required as the EtherNet card does not support auto negotiation.

Fieldbus Error Reaction

Set the Error Reaction to Shutdown (default), Warning or Ignore. Refer to the *Fieldbus Error Reaction* on page 7-147 for detailed information about each setting.

Advanced Configuration Options

The *Advanced Options:* are shown only if the checkbox next *to Show Advanced Configuration Options* is checked (refer to Fig. 7-136 below). In most cases, the default options should apply.



DeviceNet

ControlNet

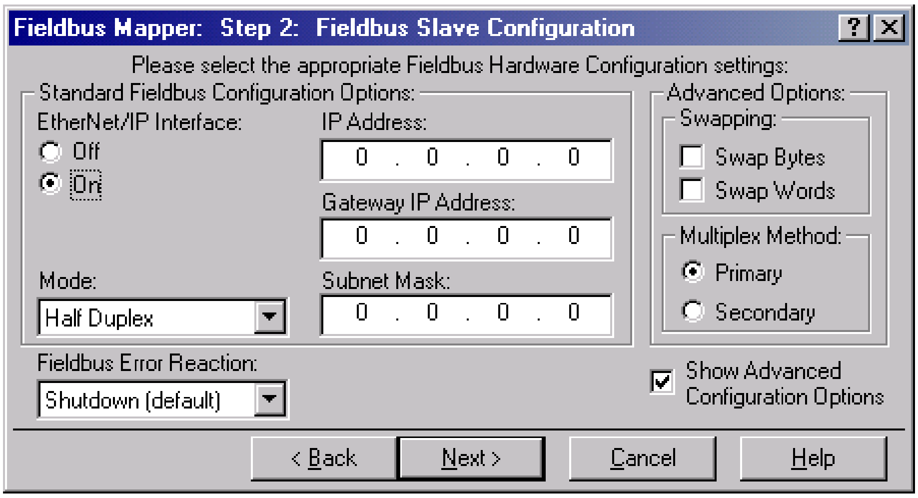
FB_Slave_Config_Adv.tif

Fig. 7-135: Fieldbus Slave Configuration: Advanced

- **Swapping:** If word and byte swapping is required by your PLC, select the checkboxes next to “Swap Bytes” and “Swap Words." Bytes and words are not swapped if the boxes are left unchecked. Refer to *Word and Byte Swapping* on page 7-148.

Note: When the Allen-Bradley 1747-SDN (DeviceNet Scanner) Module for the SLC-Series PLC is used, both **Swap Bytes** and **Swap Words** can be checked, so the order of resulting data appears correctly.

- **Multiplex Method:** select Primary or Secondary (Primary is the default). Select Secondary only if you have an inconsistent fieldbus master. Refer to *Multiplexing* on page 7-149 for detailed information about each method.



EtherNet/IP

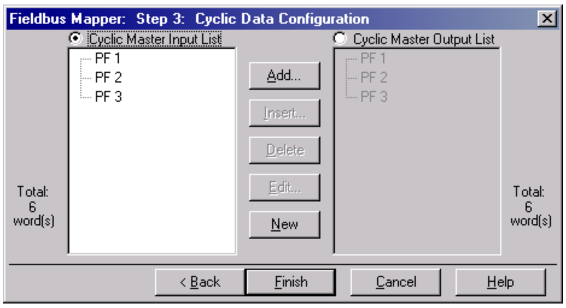
EtherNet_IP_Address_Adv.tif

Fig. 7-136: Fieldbus Slave Configuration: Advanced, for EtherNet/IP

Cyclic Data Configuration

An example of the Cyclic Data Configuration window is shown in Fig. 7-137 below. If you are editing an existing Fieldbus Mapper file, the list will probably contain more items.

First, you must select the Cyclic Input List (from PPC-R to PLC) or the Cyclic Output List (from PLC to PPC-R).

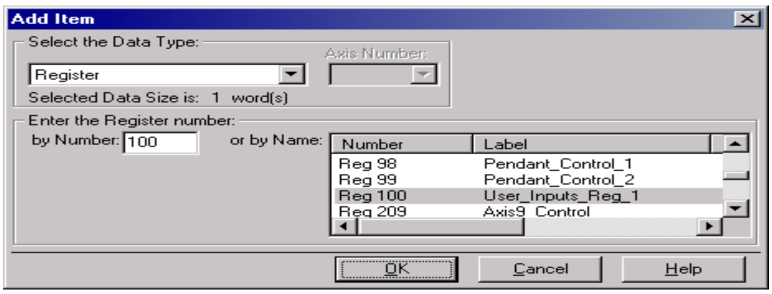


Cyclic_Data_Config.tif

Fig. 7-137: Cyclic Data Configuration

Adding an Item to the List

1. Select the Cyclic Input List or the Cyclic Output List.
2. Click ***Add***. The window in Fig. 7-138 appears. Select the Data Type (for example, Register).



FB_Add_Item.tif

Fig. 7-138: Add Item to Cyclic Data

Note: Registers and 16-bit Multiplex Containers (used only for Registers) require one data word (16 bits), and all other data types require two data words (32 bits) of space.

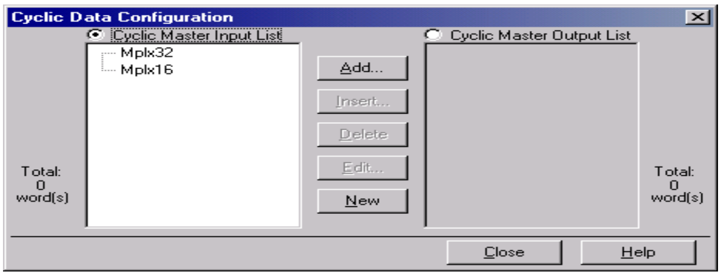
3. Enter the required information (for example Register Number) or select it from the list below. Only the available data types for your designated VisualMotion hardware setup and fieldbus type are listed.

Note: If your project is in Service mode and you check the box next to "Get Latest (Online)," the data type label list is updated based on your firmware version and the currently active program.

4. Click ***OK*** to add the selected item to the list.

Adding Multiplex Containers to the List

1. Select the Cyclic Input List or the Cyclic Output List.
2. Click ***Add***.
3. In the *Add Item* window under *Select the Data Type,* select Multiplex Container 16-bit (for Registers) or Multiplex Container 32-bit (for all other data types).
4. Click ***OK*** to add the Multiplex Container to the List. The window (Fig. 7-139) below is an example where a 16-bit Multiplex Container and a 32-Bit Multiplex Container have been added.



Cyclic_Data_Config_mltplx.tif

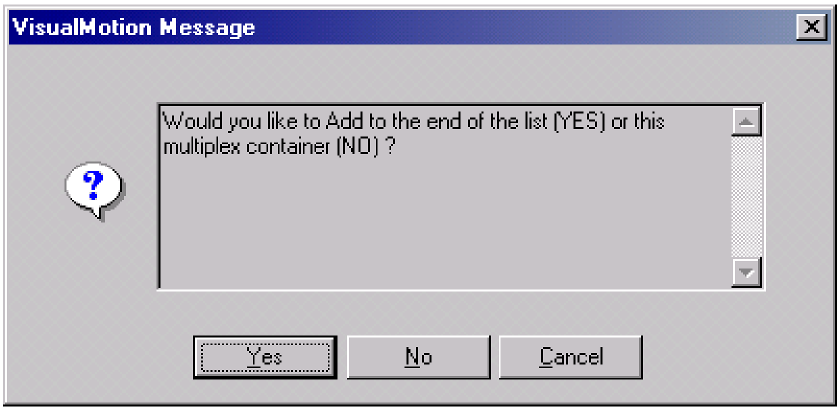
Fig. 7-139: Cyclic Data Configuration, Multiplex Containers

Note: At this point, the Multiplex Containers do not yet contain any items. To add multiplex items refer to *Adding Items to an Empty Multiplex Container* below.

Adding Items to an Empty Multiplex Container

1. In the *Cyclic Data Configuration* window, select the multiplex container to which you want to add items.

2. Click ***Add***. The window in Fig. 7-140 below appears. Because it is unclear whether you would like to add to the list or to the multiplex container, the Fieldbus Mapper is requesting clarification.



Add_Multiplex.tif

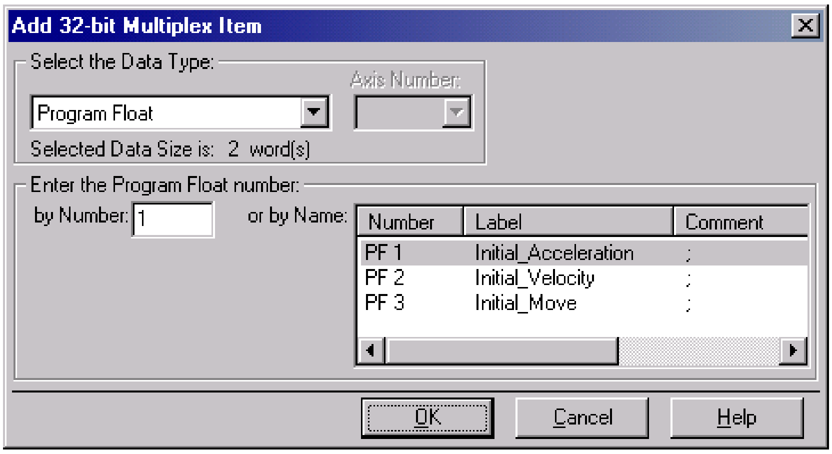
Fig. 7-140: Add Item or Multiplex Item Window

Note: For subsequent items, highlight any of the indexes within the multiplex container before clicking ***Add***, and the Fieldbus Mapper will know you want to add to that container.

3. To add to the selected multiplex container, click ***No***. The window in Fig. 7-141 below is an example for adding a 32-bit multiplex item.
4. Select the desired item to be added to the multiplex container.

Note: In addition to the data types that can be added to the multiplex list, an empty item called *Multiplex Empty Item* is available to fill a space within the multiplex container, if nothing is to be mapped to a particular index.

5. Click ***OK***. The item is automatically placed in the multiplex container as the next unassigned index item (e.g. the first item is index 00, the last is index 31).
6. Repeat for as many items as you want to add to the multiplex container, up to 32 items.



Add_Multiplex_Item.tif

Fig. 7-141: Adding a Multiplex Item to the Container (32-bit example)

Editing the Cyclic Data Lists

To make changes to an existing list, use the following buttons:

Button	Function
Add...	Inserts a new item at the end of the list.
Insert...	Inserts a new item into the list directly before the selected item.
Delete	Removes the selected item from the list.
Edit...	Allows editing of the selected item. (To edit a list item, you may also double-click on it.)
New	Clears up the current list.

Table 7-64: Button Functions in the Cyclic Data Configuration Window

Additional Functions

Several additional functions are available in the Fieldbus Mapper:

Menu Item	Function
Print	Print the current fieldbus configuration data.
Print Preview	Preview the printout of the current fieldbus configuration data
Print Setup	Configure printer settings

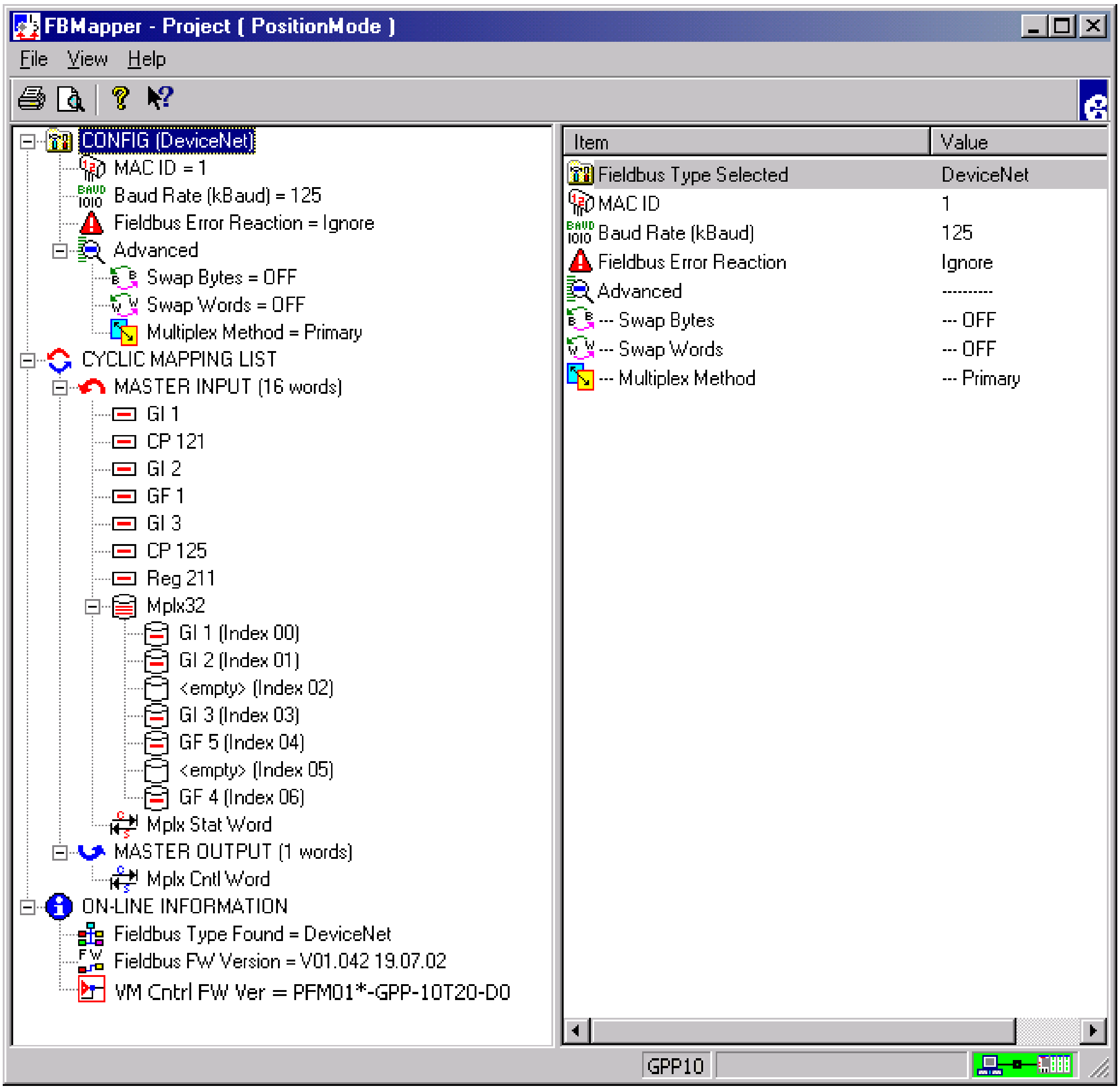
Table 7-65: Additional Functions

Getting the Fieldbus Configuration from the PPC

After getting the fieldbus configuration from the PPC, the following information is detected by the system and appears in the configuration list:

- Fieldbus Type Found
- Fieldbus FW (Firmware) Version
- GPP Control FW (Firmware) Version

An example is shown in Fig. 7-142 below. For ControlNet and EtherNet/IP fieldbuses, the configuration tree would have different elements in it.



Online info_DN.tif

Fig. 7-142: Online Fieldbus Configuration Information (DeviceNet Example)

Rexroth
Bosch Group

Information for the GPP Programmer

Fieldbus Status

VisualMotion Register 19 holds the information for "Fieldbus Status." The register information can be referenced in a VisualMotion application program to respond to the status of each bit. The use of these bits is application-dependent.

Table 7-66 below contains the bit assignment for the fieldbus status. The assigned bits are labeled with "x" and the bit number in the second row. Unassigned bits are labeled with "---."

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
---	x15	---	---	---	---	---	---	---	---	---	x5	x4	---	x2	x1

Table 7-66: Bit Assignment for VisualMotion Register 19

Bit Definitions

x1, x2 Status bits for the internal DPR (Dual-Port RAM) communication between the fieldbus slave and the PPC-R:

x1: FB Init OK , LSB (least significant bit)

x2: FB Init OK, MSB (most significant bit)

The bit combinations for x1 and x2 are as follows:

Bit 2 (PPC-R)	Bit 1 (Fieldbus)	Description
0	0	A reset has been executed on the DPR, **or** neither the PPC-R nor the fieldbus card have initialized the DPR.
0	1	The DPR is initialized by the fieldbus card, but **not** yet by the PPC-R.
1	0	The DPR initialization is complete. DPR has been initialized by the fieldbus card and PPC-R. Fieldbus to PPC-R communications system is ready.
1	1	Fieldbus to PPC-R communications system is ready.

Table 7-67: Possible Settings for Bits 1 and 2, Status Bits for DPR Communication

x4 Status bit for the active bus capabilities of the fieldbus slaves (FB Slave Ready)

0--> The fieldbus slave is not (yet) ready for data exchange.

1--> The fieldbus slave can actively participate on the bus.

x5 Status bit for the non-cyclic channel (Explicit Messaging) (Non-Cyc Ready)

0--> The non-cyclic channel (Explicit Messaging) cannot (yet) be used.

1--> The non-cyclic channel (Explicit Messaging) is ready for use by the fieldbus master.

x15 Status bit for the cyclic data output (Cyclic Data Valid):

0--> The cyclic data outputs (coming in to the PPC-R) are INVALID.

1--> The cyclic data outputs (coming in to the PPC-R) are VALID. The system looks for this bit to be 1 before allowing data transfer.

This bit is monitored for the Fieldbus Error Reaction. Whenever this bit goes to 0 after a fieldbus card was initially found by the PPC-R, the selected Error Reaction (system shutdown, error message, or ignore) is initiated. Refer to *Fieldbus Error Reaction* on page 7-147 for an explanation of the Fieldbus Error Reaction setting.

Fieldbus Diagnostics

VisualMotion Register 20 holds the information for "Fieldbus Diagnostics."

Table 7-68 below contains the bit assignment for the diagnostics. The assigned bits are labeled with "x" and the bit number in the second row. Unassigned bits are labeled with "---."

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
X16	x15	x14	x13	---	---	---	---	---	---	---	---	---	---	---	---

Table 7-68: Bit Assignment for VisualMotion Register 20

Bit Definitions

x13 - x16 Identification of the fieldbus interface card (FB Card Found)

The bit combinations for x13, x14 and x15 are as follows:

Bit 16	Bit 15	Bit 14	Bit 13	Fieldbus Type
0	0	0	0	<NO CARD>
0	0	0	1	<Not Defined>
0	0	1	0	Interbus
0	0	1	1	DeviceNet
0	1	0	0	Profibus
0	1	0	1	ControlNet
0	1	1	0	<Not Defined>
0	1	1	1	EtherNet/IP (10 MB)
1	1	1	1	Indramat PLC Interface

Table 7-69: Identification of the Fieldbus Interface

Fieldbus/PLC Cyclic Read/Write Monitoring

Monitoring of Fieldbus read/write capabilities to the cyclic channel are associated with three parameters:

- C-0-2611 Fieldbus/PLC Cyclic Channel: Current Number of Misses– displays the current number of transfers to/from the cyclic channel.
- C-0-2612 Fieldbus/PLC Cyclic Channel: Peak Number of Misses – displays the maximum number of missed transfers to/from the cyclic channel.
- C-0-2613 Fieldbus/PLC Cyclic Channel: Timeout Counter – displays the number of timeouts in the cyclic channel. If after 4 ms, the Cyclic Mapping Lists are not successfully transmitted, a "miss" is noted.

For more information about these parameters, refer to the *VisualMotion 11 Functional Description* manual.

Fieldbus Error Reaction

Note: The Fieldbus Error Reaction setting is active only in Sercos Phase 4. In all other Sercos phases, it will be inactive.

You can select how you would like the PPC-R system to react in case of a fieldbus error. This reaction can be set in the "Fieldbus Slave Configuration" window, using the combo box labeled "Fieldbus Error Reaction."

Three options are available for the Error Reaction setting. Depending on the selected setting, the value 0, 1, or 2 is stored in Parameter C-0-2635:

Setting	Value in Parameter C-0-2635
Shutdown	0 (default)
Warning Only	1
Ignore	2

Table 7-70: Parameter C-0-2635 Values for Error Reaction Settings

Fieldbus Mapper Timeout

The Fieldbus Mapper continually scans the system for sufficient resources to process the cyclic data mapping lists (2600 and 2601 lists). If 10 out of 10 consecutive attempts of the mapping list update are incomplete, the system is considered to have insufficient resources and the selected error reaction is evoked, as follows:

If "Shutdown" (0) is set in Parameter C-0-2635, the following error is generated from the PPC-R card: **520 Fieldbus Mapper Timeout**

If "Warning Only" (1) is set in Parameter C-0-2635, the following error is generated: **209 Fieldbus Mapper Timeout**

If "Ignore" (2) is set in Parameter C-0-2635, the system will update as resources become available, but there is no way to monitor whether or not updates actually occur.

Lost Fieldbus Connection

Register 19, bit 4 indicates the status of the fieldbus. Refer to *Fieldbus Status* for more specific bit information. The system monitors this bit and evokes the selected error reaction if the bit is low (0), after a fieldbus card is found. A typical situation that will cause this condition is the disconnection of the fieldbus cable from the fieldbus card.

If "Shutdown" (0) is set in Parameter C-0-2635, the following error is generated from the PPC-R (active in Sercos Phase 4 only):
519 Lost Fieldbus Connection

If "Warning Only" (1) is set in Parameter C-0-2635, the following error is generated (active in Sercos Phase 4 only):
208 Lost Fieldbus Connection

If "Ignore" (2) is set in Parameter C-0-2635, there is no noticeable reaction when the Register 19 status bits go low, unless the GPP application program is customized to evoke a special reaction.

Troubleshooting Tip:

If a fieldbus card is not found on the system, the Error Reaction setting will be ignored. If you have a fieldbus card and the Error Reaction is not responding as expected, the system may not "refer to" your fieldbus card.

Information for the PLC Programmer

*.eds File

Rexroth supplies an *.eds file containing supporting information for the PPC-R with a DeviceNet or ControlNet slave configuration. This file is provided on the VisualMotion 11 installation CD.

Word and Byte Swapping

In the Fieldbus Mapper, it is possible to enable automatic word and byte swapping for DeviceNet, ControlNet, and EtherNet/IP fieldbuses (for both input and output), depending on the type of PLC used.

- **32-bit Object Word Swapping** - The setting of this option determines the order in which the two data words in **any 32-bit (double word) cyclic or non-cyclic mapped object** are transmitted. The default setting, "Do not swap words" (“Swap Words” checkbox unchecked under the Advanced Options) causes the words to be transmitted in their usual order: [Word 1], [Word 2]. The "Swap Words" setting (“Swap Words” checkbox checked under the Advanced Options) causes the words to be transmitted in inverted order: [Word 2], [Word 1]. The setting of this option is stored in Card Parameter C-0-2636, bit 0.
- **Explicit Message Byte Swapping** - The setting of this option determines the order in which the bytes of **non-cyclic data >4 bytes long** are transmitted. The default setting, "Do not swap bytes" (“Swap Bytes” checkbox unchecked under the Advanced Options) causes the bytes to be transmitted in their usual order: [Byte 1], [Byte 2], [Byte 3], [Byte 4], [Byte 5], [Byte 6].... The "Swap Bytes" setting (“Swap Bytes” checkbox checked under the Advanced Options) causes each pair of bytes to be transmitted in inverted order: [Byte 2], [Byte 1], [Byte 4], [Byte 3], [Byte 6], [Byte 5].... The setting of this option is stored in Card Parameter C-0-2636, bit 1.

Example: Allen-Bradley 1747-SDN Module for the SLC-Series PLC

When the Allen-Bradley 1747-SDN (DeviceNet Scanner) Module for the SLC-Series PLC is used, both **Swap Words** and **Swap Bytes** can be checked in the Fieldbus Mapper, so the order of resulting data appears correctly.

Multiplexing

Primary Multiplex Method (for Consistent Masters only)

Important: You should use the Primary Multiplex Method only for a master that is consistent over the entire cyclic channel. The Secondary Multiplex Method is available for inconsistent masters. Refer to *Explanation of the Master Consistency Problem* on page 7-152.

The advantage of the Primary Method is easier handling of input data for consistent masters.

Control Word and Status Word

Control Word

The control word is transferred in the multiplex channel from master to slave. It tells the slave in which index the data is being transferred from master to slave and in which index the data is requested from slave to master.

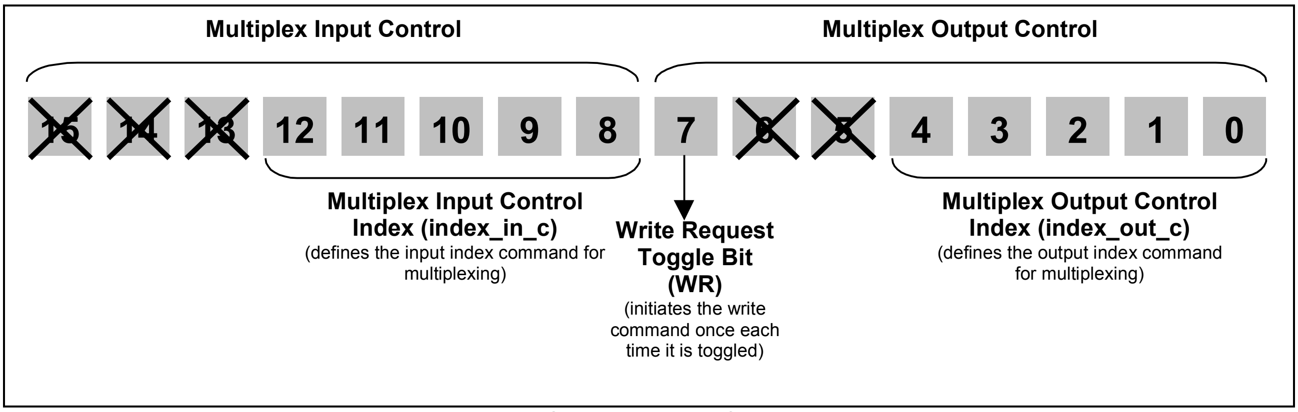


Fig. 7-143: Control Word Definition, Primary Multiplex Method

- **Index_out_c:** tells the slave in which index the data are transferred from master to slave (out = master -> slave, _c = element of control word).
- **Index_in_c:** tells the slave in which index the data is requested from slave to master (in = slave -> master, _c = element of control word).
- **WR (Write Request):** handshake bit (refer to meaning of WR and WA).

Note: Input data via the Multiplex Channel is continually being updated.

Status Word

The status word is transferred in the multiplex channel from slave to master. It acknowledges the written index and the requested index.

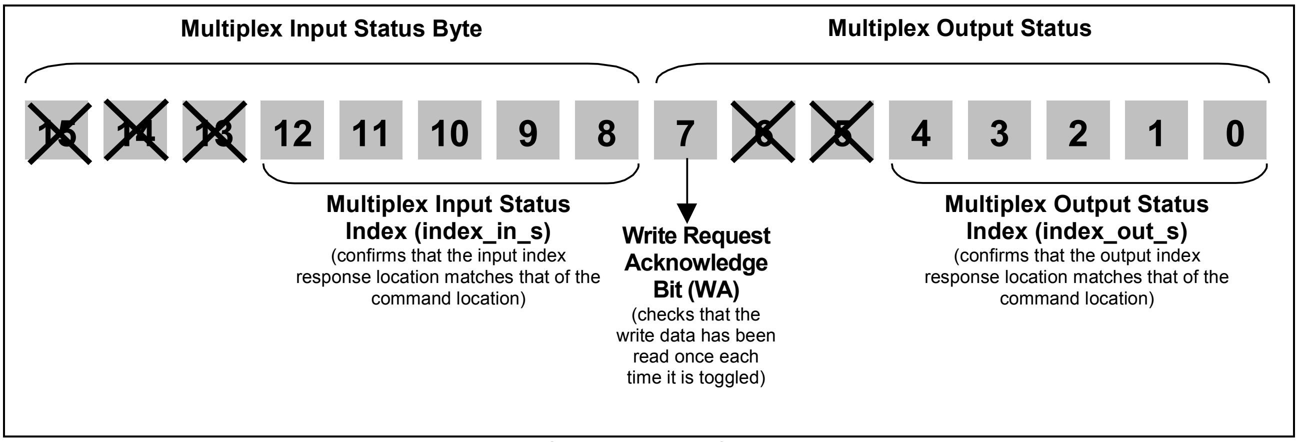


Fig. 7-144: Status Word Definition, Primary Multiplex Method

- **Index_out_s:** acknowledges index written by the master (out = master -> slave, _s = element of status word).
- **Index_in_s:** tells the master which index is transferred from slave to master in the actual process data cycle (in = slave -> master, _s = element of status word).
- **WA (Write Acknowledge):** Handshake bit (refer to meaning of WR and WA).

Handshake Bits WR and WA

WR and WA are handshake bits that allow the controlled writing of data via the multiplex channel. WR and WA control the data transfer for writing data_out (data send from master to slave).

WR == WA:

- tells the master that the slave has received the last multiplex data_out. The master can now send new data_out.
- tells the slave to do nothing, because the master has not yet put new consistent data_out on the bus.

WR! = WA:

- tells the slave to do something, because the master has now put consistent new data_out on bus.
- tells the master to do nothing, because the slave has not yet received the latest multiplex data_out.

Master Communications (Primary Multiplex Method)

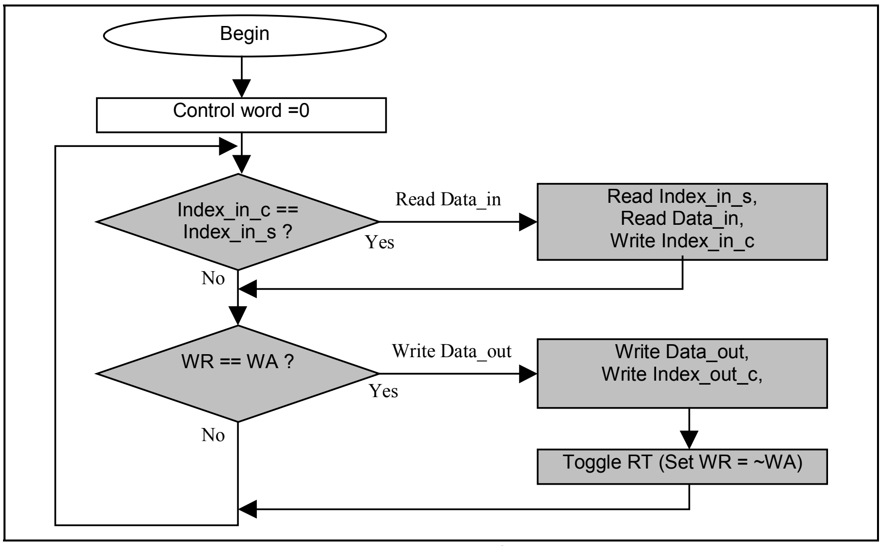


Fig. 7-145: Primary Multiplex Method, Master Communications

Programming Example

To aid in implementing the multiplex function in a PLC program, the following flow chart shows two ways of reading and writing data. Reading and writing can be executed separately, which allows the input data to be updated about 30% faster. The “Read Data” example would be placed at the beginning of a PLC program the “Write Data” example at the end.

Combined reading and writing makes the PLC program simpler, especially when using the same index for both transfer actions.

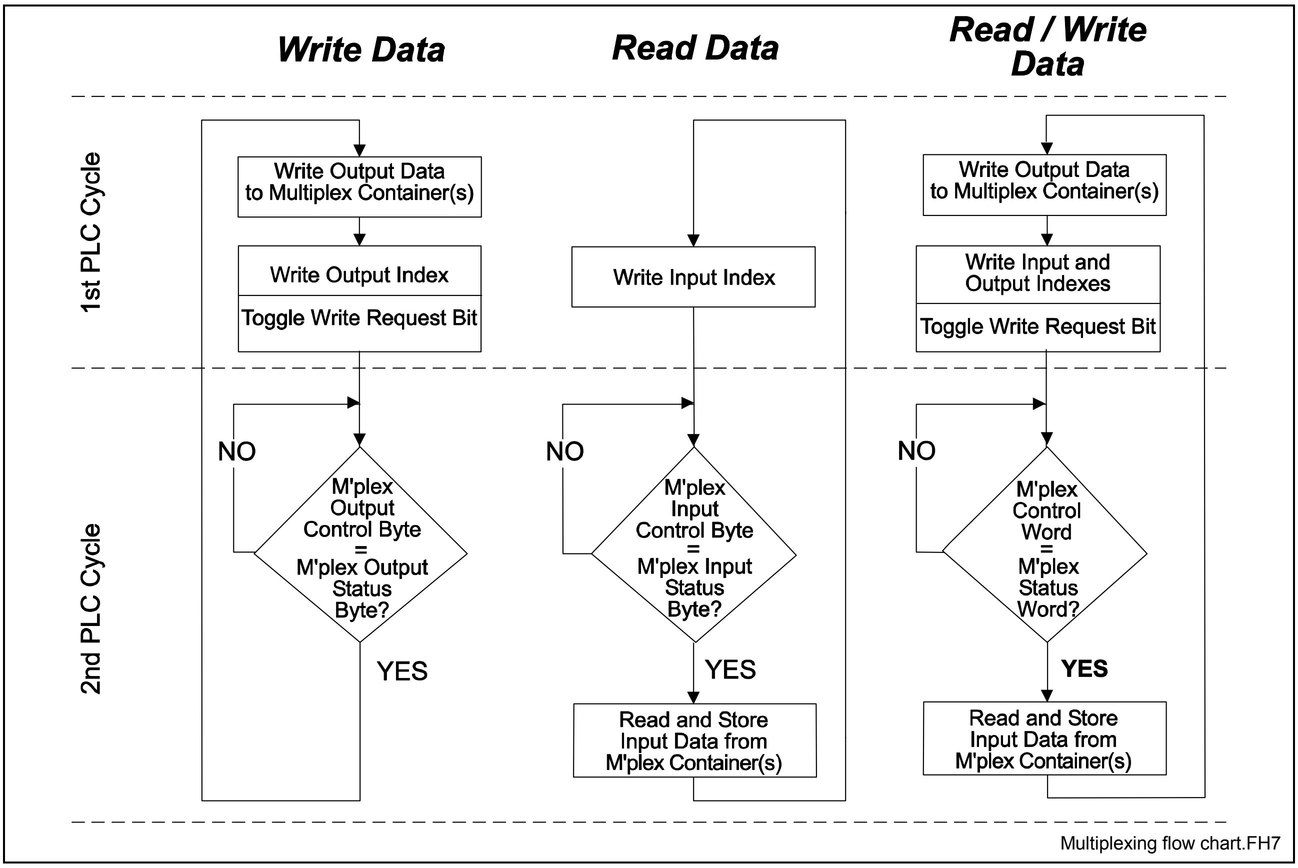


Fig. 7-146: Flow Chart of Multiplex Programming Examples (Primary Method)

Rexroth Bosch Group

Secondary Multiplex Method (for Inconsistent Masters)

Explanation of the Master Consistency Problem

The PPC-R fieldbus slave interfaces can guarantee consistency, however, some fieldbus masters can only guarantee byte, word or double word consistency. If the master is only word-consistent, it is possible that the master cannot transfer the data and the control word of one multiplex index consistently from the PLC to the fieldbus. Therefore, it is necessary to have a second multiplex method where both input data and output data require the handshake bits to update via the fieldbus.

Note: The meanings of the control and status words are the same as for the Primary Multiplex Method. The only difference is that toggle bits RR and RA are used in the Secondary Method.

Fig. 7-147 and *Fig. 7-148* below illustrate the control and status word definitions for the Secondary Multiplex Method.

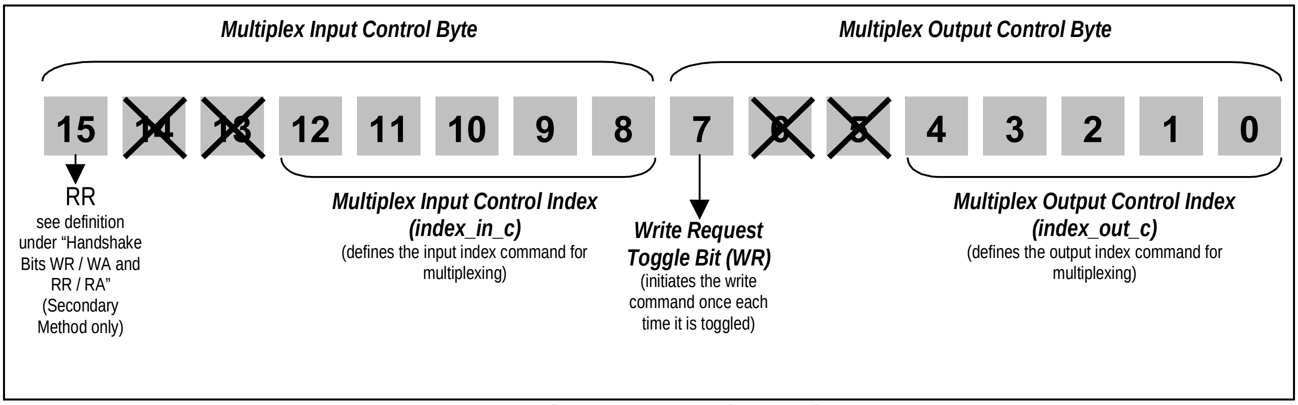


Fig. 7-147: Control Word Definition, Secondary Multiplex Method

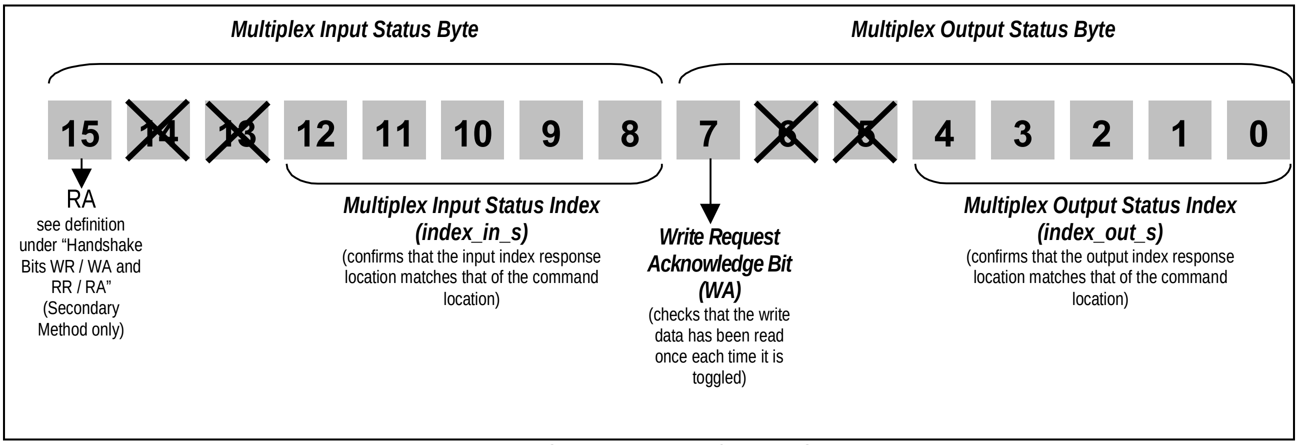


Fig. 7-148: Status Word Definition, Secondary Multiplex Method

The Secondary Multiplex Method has the following features:

- You can transfer a different index from master to slave as from slave to master.
- The handshake bits for both reading and writing of this multiplex channel make the multiplexing possible on inconsistent systems (masters).

Handshake Bits RR and RA

RR (Read Request) and RA (Read Acknowledge) are handshake bits that allow a controlled data transfer and use of the multiplex channel on inconsistent masters. RR and RA control the data transfer for reading data_in (data send from slave to master).

RR == RA:

- tells the master that the slave has sent the requested data_in. The master can now read the data_in and request new data_in.
- tells the slave to do nothing, because the master has not yet put new consistent data on the bus.

RR != RA:

- tells the slave to put new data_in on the bus, because the master requests new data_in.
- tells the master to do nothing, because the slave has not yet put the latest requested multiplex data_in on the bus.

Master Communications (Secondary Multiplex Method)

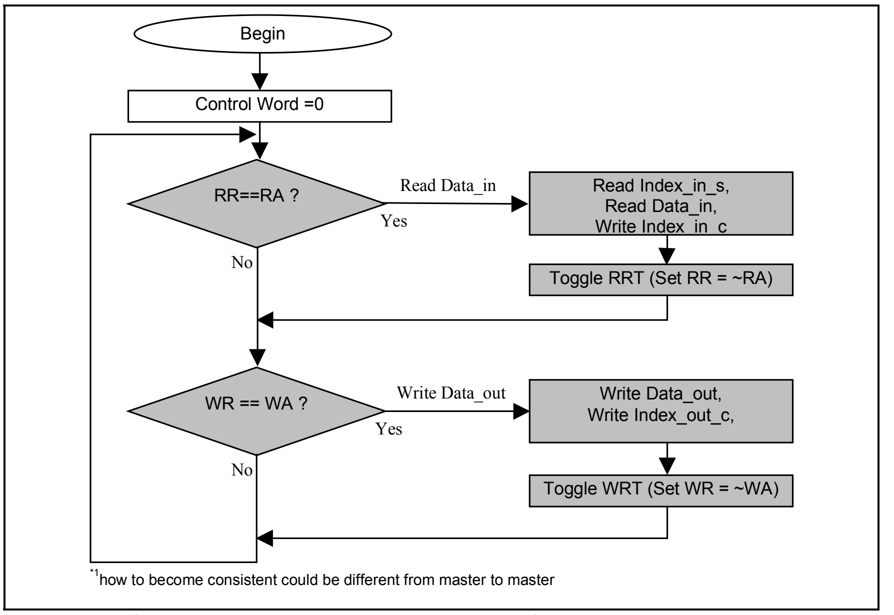


Fig. 7-149: Secondary Multiplex Method, Master Communications

For some masters, it could be enough to first write data and then the control word. For other masters, you may have to implement a delay time (this time could be different from master to master) before writing WR = ~WA.

Non-Cyclic Data (Explicit Messaging)

The following methods for transferring data are available via DeviceNet, ControlNet, and EtherNet/IP Explicit Messaging:

- Mapped Data
- Data Exchange Objects

Mapped Data

Mapped data is the most powerful feature of the PPC-R non-cyclic fieldbus interface. Through mapped data, the user has access to virtually every PPC-R parameter over the fieldbus. It is easy to implement from the PLC side and requires no setup on the PPC-R side.

To access a VisualMotion data type over the fieldbus, it has to be specified by an address that consists of a Class, Instance and Attribute. The Class, Instance and Attribute for each data type can be calculated by a formula (refer to
Example Lookup Tables for Mapped **Data** on page 7-163).

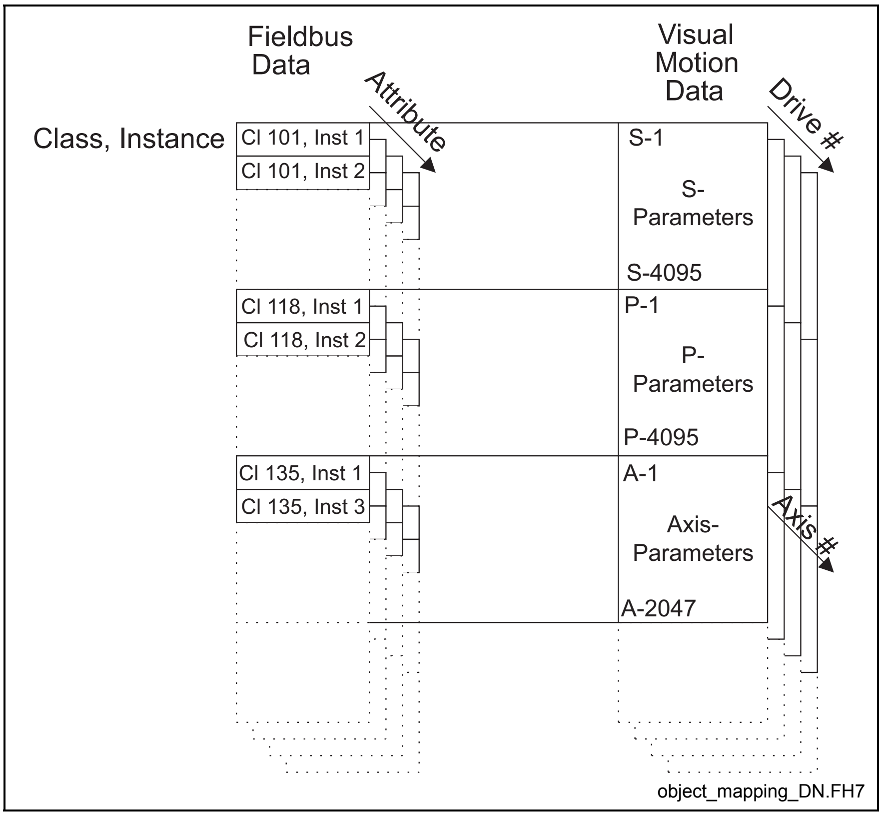


Fig. 7-150: Mapped Data

Mapped data can be used with the following parameters and values:

- S-Parameters (SERCOS Drive S-Parameters)
- P-Parameters (SERCOS Drive P-Parameters)
- A-Parameters (PPC Axis Parameters)
- C-Parameters (PPC C System parameters)
- T-Parameters (PPC Task parameters)

size and format depend on parameter *,1

PF-Values (PPC Program Float data, 32 bit – 2 words, IEEE format) *,2

GI-Values (PPC Global Integer data, 32 bit – 2 words) *,2

GF-Values (PPC Global Float data, 32 bit – 2 words, IEEE format) *,2

PI-Values (PPC Program Integer data, 32 bit – 2 words) *,2

Reg.-Values (PPC Register data, 16 bit – 1 word) *,3

Data Exchange Objects (0x5E70 – 0x5E73) (embedded ASCII Protocol)

*You may notice that parameters accessed via the non-cyclic (Parameter) channel are not always the same size as reported from the attribute field. This is so that the data sizes correspond with the way the different data types are handled in the cyclic channel (Registers are always set to 16-bit size and Parameters are cast to 32-bit size, even if they actually use less space).

4. When **writing** mapped data to a VisualMotion Parameter, you must send the size data corresponding to that of the attribute field within the parameter.
   a.) For 32-bit parameters, you must send a data size of 32 bits (otherwise, VM error #07 is returned).
   b.) For 16-bit parameters, you must send a data of size 16-bits. If, for this case, you send data of size 32 bits, one of the following occurs:

i.) For parameters of type 16-bit unsigned, only the Low word is stored, and the High word is ignored.

ii.) For parameters of type 16-bit signed, bits 0-14 of the low word along with the sign bit #31 are used, and the remaining bits are ignored.

c.) For String Parameters (e.g. S-0-0142), you must send the size of the string to write.

d.) All other Parameter Types (list parameters, command parameters, etc), are not supported for mapped data.

When reading mapped data from a VisualMotion Parameter, there are 3 possible cases of sizes returned:

a.) If the parameter type is a string, you receive the number of bytes corresponding to the length of the string.

b.) If the parameter is 32-bit or less, you receive a cast 32-bit value for this parameter. This implies that 16-bit parameters are returned as cast in to 32-bit values.

c.) All other parameter types (e.g. list parameters, command parameters, etc.), are not supported for mapped data.

5. When writing mapped data to a VisualMotion Program Float, Program Integer, Global Float, or Global Integer, the data size must be 32-bits (2 words). Any other size returns a VM error #07 (Invalid Data Format).

   When reading mapped data from a VisualMotion Program Float, Program Integer, Global Float, or Global Integer, the data size returned is always 32-bit (2 words).

6. When writing mapped data to a VisualMotion Register, the data must be 16-bits (1 word). Any other size returns a VM error #07 (Invalid Data Format).

   When reading mapped data from a VisualMotion Register, the data size returned is always 16-bit (1 word).

Selecting Mapped Data

To access a data type over the fieldbus, it has to be specified by an address that consists of a Class, Instance and Attribute. Class, Instance and Attribute for each data type can be calculated by a formula (refer to **Accessing Mapped Data** on page 7-161).

Transmission Sequence for Mapped Data

Note: For mapped data, only one transmission (and one response) is required, to send a read or write message to and receive a response from the PPC-R.

Fieldbus Message

Header	Data Field: Raw Data
key components: • service code: Read = 0x0E Write = 0x10 • mapped data (under ***VisualMotion Mapped Data*** at the end of this section). This is the destination of the data field. Selection of the Class, Instance and Attribute depends on the required VisualMotion. • Data field size (in bytes) minimum = 6 bytes for Class, Instance, Attribute	**key components:** • raw data to be transmitted (write message only; unused for read message)

Important:

The format of the Fieldbus message header and the method of implementation are dependent on the Fieldbus type and the master (PLC) being used. Refer to your Fieldbus master/PLC documentation for proper transport and formatting of the message header.

Non-Cyclic Mapped Data Write

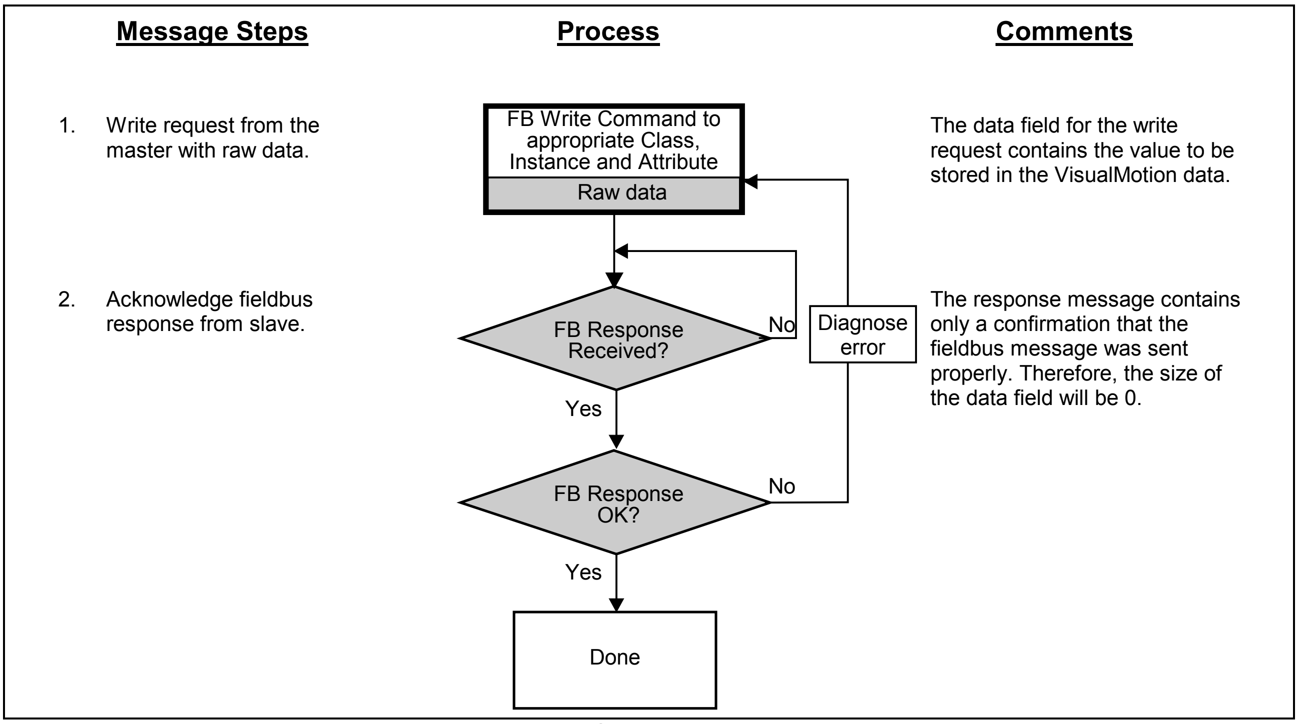


Fig. 7-151: Non-Cyclic Mapped Data Write Process

Example: Write the value -27.43 to Program Float 16 (This is a 32-bit data type, which is mapped to Class 165, Instance 1, and Attribute 16. The Class, Instance and Attribute can be calculated using the formulas under **Accessing Mapped Data** at the end of this chapter.)

1. Write request from the master with raw data.

Header: Write command	**Data Field: VisualMotion raw data**

contains:

- Write service code = (0x10)
- object number (Class 165, Instance 1, Attribute 16)
- data field size (bytes—in this case: 4) + (6 bytes for Class, Inst, Att) = 10 bytes

contains data (here, shown in decimal format):

`-27.43`

2. After the write request from the master, the PPC-R sends a response message.

Header: Response message	**No data field**

contains:

- “Write” message o.k. is denoted by a Response service code of (0x90)
- data field size = 0 bytes

3. If the message response (code in message header) shows o.k., the transaction is complete.

Non-Cyclic Mapped Data Read

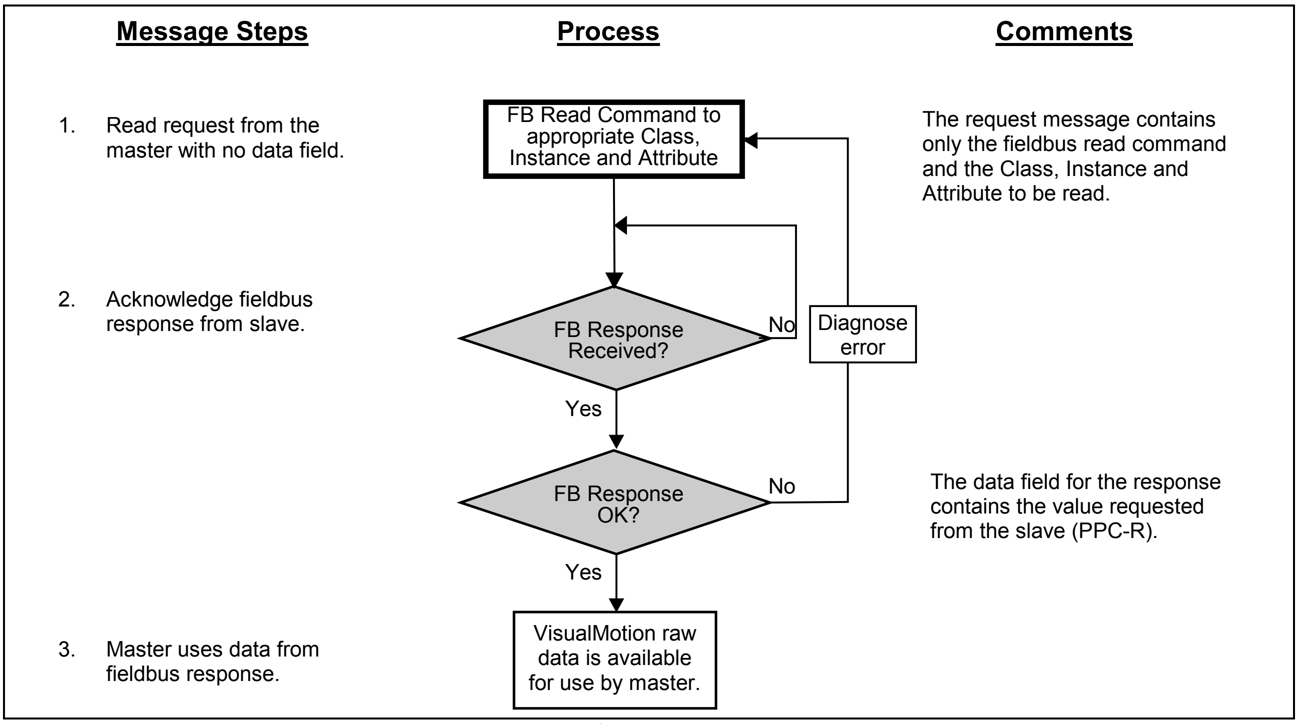


Fig. 7-152: Non-Cyclic Mapped Data Read Process

Example: Read the value contained in Program Integer 8. (This is a 32-bit data type, which is mapped to Class 165, Instance 21, and Attribute 8. The Class, Instance and Attribute can be calculated using the formulas under, "Accessing Mapped Data," at the end of this chapter.)

1. Read request from the master.

Header:
Read command

No data field

contains:
- Read service code = (0x0E)
- Class 165, Instance 21, Attribute 8
- data field size = 6 bytes for Class, Inst, Att

2. After the read request from the master, the PPC-R sends a response message.

Header:
Response message

Data Field:
VisualMotion raw data

contains:
- "Read" message o.k. is denoted by a Response service code of (0x8E)
- data field size = 4 bytes

contains data (here, shown in decimal format):
`12345`

If the message response (code in message header) shows o.k., the requested value is attached to the message in the data field. This value is now available for use by the master (PLC).

Data Exchange Objects

The four data exchange objects Class 100, Instance 1-4, Attribute 100 represent fixed data "containers" of varying lengths that transfer the VisualMotion ASCII Protocol to the PPC-R. These objects serve as an open-ended possibility to access any VisualMotion data (including cams, diagnostic text, etc.), but more work is required in the master to perform a transmission of this type. Both the VisualMotion ASCII message and the fieldbus transfer message must be formulated.

Table 7-71 below lists the available data exchange objects and their sizes.

Data Exchange Object	Data Length (in bytes)
Class 100, Instance 1, Attribute 100	16
Class 100, Instance 2, Attribute 100	32
Class 100, Instance 3, Attribute 100	64
Class 100, Instance 4, Attribute 100	128

Table 7-71: Length of the Data Exchange Objects

Selecting a Data Exchange Object

Depending on the length of a VisualMotion ASCII message, any of these data exchange objects can be selected.

Note: The entire data length of the data exchange object must always be transmitted even if the VisualMotion ASCII message is shorter.
For example, if you want to transmit an ASCII message of 42 bytes, you must use Class 100, Instance 3. To avoid a response error from the Fieldbus slave, you must append 22 "Null" characters to the end of the ASCII message to complete a data size of 64 bytes.

Note: The checksum for the VisualMotion ASCII protocol is NOT used with the data exchange object. If the checksum is sent as part of the string, it will be ignored, and no checksum will be sent in the VisualMotion ASCII response messages. To ensure data integrity, the Fieldbus protocols support a low-level checksum.

Transmission Sequence via a Data Exchange Object

Note: For the data exchange object, two transmissions (and two responses) are required, to send the read or write message to and then receive the response message from the PPC-R.

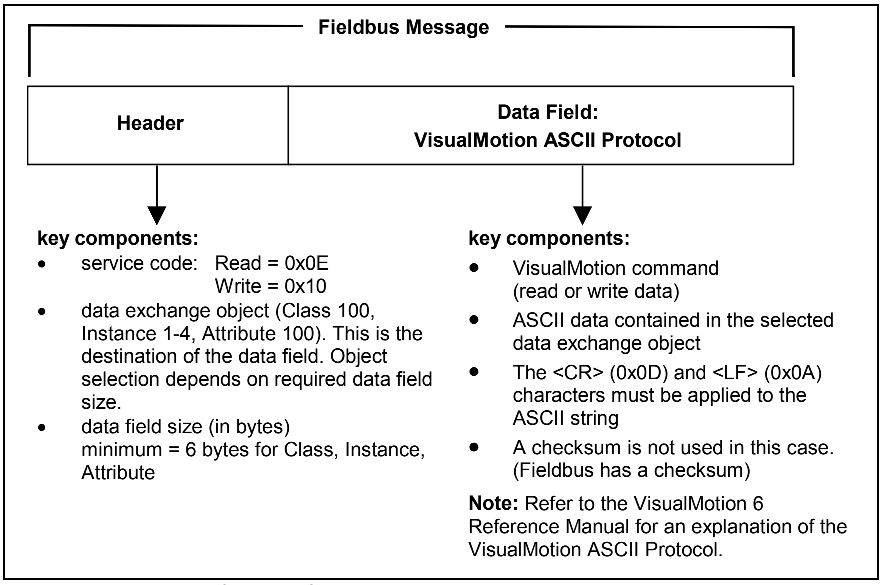


Fig. 7-153: Format of a Non-Cyclic Fieldbus Message using a Data Exchange Object

Important: The format of the fieldbus message header is dependent on the type of master (PLC) being used. Refer to your PLC manufacturer's manual for specific information on this topic.

The following sequence describes the communication between the Fieldbus master (PLC) and the Fieldbus slave (PPC-R):

Message Steps | **Process** | **Comments**

1. Write request from the master with VisualMotion ASCII Protocol.

The data field for the write request can contain a VisualMotion read or write protocol.

2. Acknowledge fieldbus response from slave.

The response message contains only a confirmation that the fieldbus message was sent properly. Therefore, the size of the data field will be 0.

3. Read request from the master to get VisualMotion ASCII response.

The read request message contains only header information (the data field is 0). You must anticipate the approximate size of the data field (VisualMotion ASCII response message) in order to select the appropriate data exchange object. If the selected object is too short, the data will be truncated.

4. Receive fieldbus response from slave.

The response message will contain the VisualMotion ASCII response to the VisualMotion ASCII command text in Step 1.

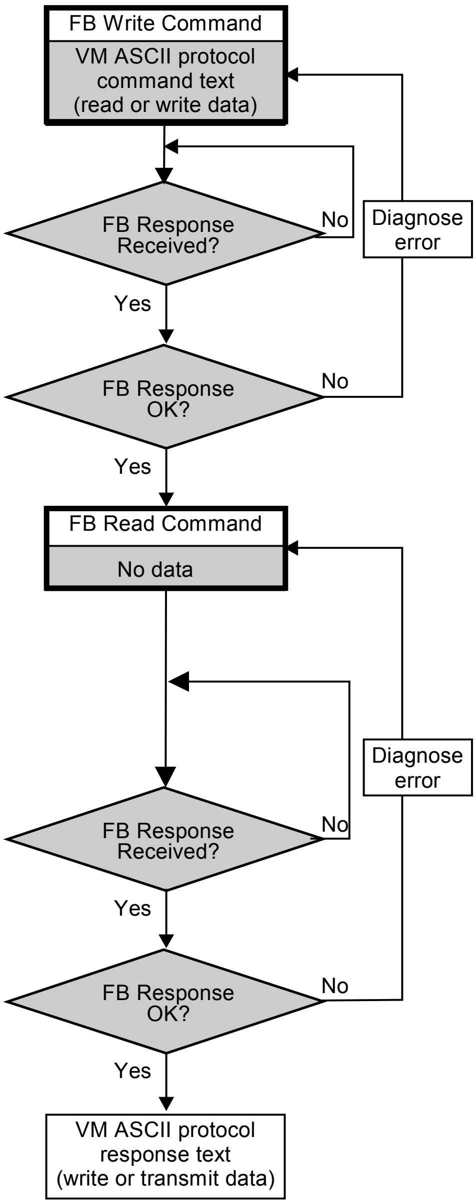


Fig. 7-154: Non-Cyclic (Explicit Messaging) VisualMotion ASCII Communication Process

Example: Read Card Parameter 100 (PPC-R firmware version)

1. Write request from the master with VisualMotion ASCII Protocol.

Header: Write command	**Data Field: VisualMotion ASCII Protocol**
contains: • Write service code = (0x10) • object number (Class 100, Instance 1, Attribute 100) • data field size (bytes—in this case: 16) + (6 bytes for Class, Inst, Att) = 22 bytes	**contains code:** `>0_CP_1.100\CR\LF\00\00\00`

2. After the first read request from the master, the PPC-R sends a response message.

Header: Response message	**No data field**
contains: • “Write” message o.k. is denoted by a Response service code of (0x90) • data field size = 0 bytes	

3. Read request from the master for the VisualMotion ASCII response message.

Header: Read command	**No data field**
contains: • Read service code = (0x0E) • object number (Class 100, Instance 3, Attribute 100 —anticipated return data size between 32 and 64 bytes) • data field size = 6 bytes for Class, Inst, Att	

Note: To ensure that all of the data requested in this step is received in step 4 below, a data exchange object of the appropriate size must be selected.
If the selected data exchange object is too small, the data will be truncated.
If the selected data exchange object is too large, efficiency of transmission will be compromised.

4. The PPC-R sends the final response message.

Header: Response message	**Data Field: VisualMotion ASCII Protocol**
contains: • “Read” message o.k. is denoted by a Response service code of (0x8E) • data field size = 64 bytes	**contains code:** `>0_CP_1.100_PSM01*-GPP-07V11-MS\CR\LF\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00`

DeviceNet, ControlNet, and EtherNet/IP General Error Codes

Error No. (Hex)	Error Name	Error Description
0x09	Invalid attribute value	Invalid attribute data detected.
0x0E	Attribute not settable	A request to modify a non-modifiable attribute was received.
0x13	Not enough data	The service did not supply enough data to perform the specified operation.
0x14	Attribute not supported	The attribute specified does not exist in the device.
0x15	Too much data	The service supplied more data than was expected.
0x16	Object does not exist	The object specified does not exist in the device.
0x1F*	Vendor-specific error	A vendor-specific error has been encountered. The Additional Code Field of the Error Response defines the particular error encountered. Use of this General Error Code should only be performed when none of the Error Codes presented in this table or within an Object Class definition accurately reflects the error. Refer to for information on Low-Byte Error Codes for 0x1F
* Note: This error code is not valid for ControlNet		

Table 7-72: DeviceNet Error Codes

Explicit Messaging Error Codes (Low-Byte) for 0x1F

Error No. (Hex)	Error Description
0xF3	Invalid Attribute – Occurs when an attempt to access an incorrect or undefined location in the mapped data area.
0xF2	Invalid Class and Instance – Occurs when attempting to access an incorrect or undefined location in the mapped data area.
0xF1	Not used
0xF0	ASCII Format Error – occurs when attempting to communicate via the Data Exchange object where VisualMotion ASCII protocol is sent. This error also occurs if the initial characters are incorrect (such as the absence of the “>” start character).
< 0xF0	Error is based on VisualMotion Communication Error codes. Refer to Communication Error Codes for details.

Table 7-73: Parameter Channel Error Codes (Low-Byte)

Accessing Mapped Data

Rexroth has pre-configured a number of VisualMotion data types to DeviceNet, ControlNet, or EtherNet/IP Classes, Instances and Attributes. We call this concept-**mapped data**. These data types can be accessed via DeviceNet/ControlNet/EtherNet/IP Explicit Messaging. The Class, Instance and Attribute for each of these data types can be calculated using the formulas in *Table 7-74* below.

Rexroth
Bosch Group

	Class, Instance	Attribute	Formula
Data Exchange Object	Class 166 Instance 137	0	Note: for backwards compatibility, also listed as
	----	----	Class 100, Instance 1 - 4, Attribute 100
	Class 166 Instance 134	0	
<FREE> (349 objects available)	Class 166, Instance 133	255	
	----	----	
	Class 165 Instance 41	1	
Program Integers (Int 1 – Int 5100)	Class 165 Instance 40	255	Class = 165
	----	----	Instance = 21 + [(Program Integer - 1) \ 255]
	Class 165 Instance 21	1	Attribute = Program Integer - [(Instance - 21) * 255)]
Program Floats (Float 1 – Float 5100)	Class 165 Instance 20	255	Class = 165
	----	----	Instance = 1 + [(Program Float - 1) \ 255]
	Class 165, Instance 1	1	Attribute = Program Float - [(Instance - 1) * 255)]
<FREE> (235 objects available)	Class 164, Instance 255	255	
	----	----	
	Class 164, Instance 21	1	
Global Integers (GInt 1 – GInt 2550*)	Class 164, Instance 20	255	Class = 164
	----	----	Instance = 11 + [(Global Integer - 1) \ 255]
	Class 164, Instance 11	1	Attribute = Global Integer - [(Instance - 11) * 255)]
Global Floats (GFloat 1 – Gfloat 2550*)	Class 164, Instance 10	255	Class = 164
	----	----	Instance = 1 + [(Global Float - 1) \ 255]
	Class 164, Instance 1	1	Attribute = Global Float - [(Instance - 1) * 255)]
<FREE> (245 objects available)	Class 163, Instance 255	255	
	----	----	
	Class 163, Instance 11	1	
Registers (Reg. 1 – Reg. 2550**)	Class 163, Instance 10	255	Class = 163
	----	----	Instance = 1 + [(Register - 1) \ 255]
	Class 163, Instance 1	1	Attribute = Register - [(Instance - 1) * 255)]
T-Parameters (T-0-0001 – T-0-1020)	Class 162, Instance 255	4	Class = 159 + [(T-Parameter - 1) \ 255]
	----	----	Instance = T-Parameter - [(Class - 159) * 255]
	Class 159, Instance 1	1	Attribute = Task Number
<FREE> (GFloat 1 – Gfloat 2550)	Class 158, Instance 255	255	
	----	----	
	Class 158, Instance 14	1	
C-Parameters	Class 158, Instance 13	1	Class = 144 + [(C-Parameter - 1) \ 255]
	----	----	Instance = C-Parameter - [(Class - 144) * 255]

	Class, Instance	Attribute	Formula
(C-0-0001 - C-0-3583)	Class 144, Instance 1	1	Attribute = 1 (data)
A-Parameters (A-0-0001 - A-0-2047)	Class 143, Instance 7	99	Class = 135 + [(A-Parameter - 1) \ 255]
	----	----	Instance = A-Parameter - [(Class - 135) * 255]
	Class 135, Instance 1	1	Attribute = Axis Number
P-Parameters (P-0-0001 - P-0-4095)	Class 134, Instance 15	99	Class = 118 + [(P-Parameter - 1) \ 255]
	----	----	Instance = P-Parameter - [(Class - 118) * 255]
	Class 118, Instance 1	1	Attribute = Drive Number
S-Parameters (S-0-0001 - S-0-4095)	Class 117, Instance 15	99	Class = 101 + [(S-Parameter - 1) \ 255]
	----	----	Instance = S-Parameter - [(Class - 101) * 255]
	Class 101, Instance 1	1	Attribute = Drive Number

* current limitation: C-0-0080/C-0-0081 - Maximum number global integers/floats.
**current limitation: first 1024 registers.

Table 7-74: Formulas for Determining Mapped Objects

Example Lookup Tables for Mapped Data

Card (C) Parameters

The following is an example lookup table for C-Parameters, when using mapped objects.

Example Look-up Chart for: **C-Parameters** CP 0.Y ==> CP = Card Parameter
Y = Parameter Number

	Class 144	Class 144	Class 144	....	Class 144	Class 145	....	Class 158	Class 158
	Instance 1	Instance 2	Instance 3	....	Instance 255	Instance 1	....	Instance 12	Instance 13
Attribute ID 1	CP 0.1	CP 0.2	CP 0.3		CP 0.255	CP 0.256		CP 0.3582	CP 0.3583

Table 7-75: C-Parameters Lookup Table for Mapped Data Types

Axis(A) Parameters

The following is an example lookup table for A-Parameters, when using mapped objects. The same formula also applies to Sercos (S) and Task (T) Parameters.

Rexroth Bosch Group

Example Look-up Chart for: **A-Parameters** AP X.Y ==> AP = Axis Parameter
X = Axis Number
Y = Parameter Number

		Class 135	Class 135	Class 135	....	Class 135	Class 136	....	Class 143	Class 143
		Instance 1	Instance 2	Instance 3	....	Instance 255	Instance 1	....	Instance 6	Instance 7
	1	AP 1.1	AP 1.2	AP 1.3		AP 1.255	AP 1.256		AP 1.2046	AP 1.2047
Attribute ID	2	AP 2.1	AP 2.2	AP 2.3		AP 2.255	AP 2.256		AP 2.2046	AP 2.2047
	3	AP 3.1	AP 3.2	AP 3.3		AP 3.255	AP 3.256		AP 3.2046	AP 3.2047
	:	:	:	:	:	:	:	:	:	:
	:	:	:	:	:	:	:	:	:	:
	40	AP 40.1	AP 40.2	AP 40.3		AP 40.255	AP 40.256		AP 40.2046	AP 40.2047

Table 7-76: A-Parameters Lookup Table for Mapped Data Types

Product-Specific (P) Parameters

The following is an example lookup table for P-Parameters, when using mapped objects.

Example Look-up Chart for: **P-Parameters** PP X.Y ==> PP = SERCOS P-Parameter (set 0 only)
X = Drive Number
Y = Parameter Number

		Class 118	Class 118	Class 118	....	Class 118	Class 119	....	Class 134	Class 134
		Instance 1	Instance 2	Instance 3	....	Instance 255	Instance 1	....	Instance 14	Instance 15
	1	PP 1.1	PP 1.2	PP 1.3		PP 1.255	PP 1.256		PP 1.4094	PP 1.4095
Attribute ID	2	PP 2.1	PP 2.2	PP 2.3		PP 2.255	PP 2.256		PP 2.4094	PP 2.4095
	3	PP 3.1	PP 3.2	PP 3.3		PP 3.255	PP 3.256		PP 3.4094	PP 3.4095
	:	:	:	:	:	:	:	:	:	:
	:	:	:	:	:	:	:	:	:	:
	40	PP 40.1	PP 40.2	PP 40.3		PP 40.255	PP 40.256		PP 40.4094	PP 40.4095

Table 7-77: P-Parameters Lookup Table for Mapped Data Types

Integers

The following is an example lookup table for Integers, when using mapped objects. The same formula also applies to Floats, Global Integers, Global Floats and Registers.

Example Look-up Chart for: **VM Program Integers** PI 0.Y ==> PI = Program Integer
Y = Program Integer Number

		Class 165 Instance 21	Class 165 Instance 22	Class 165 Instance 23	....	Class 165 Instance 40
	1	PI 1	PI 256	PI 511		PI 4846
Attribute ID =	2	PI 2	PI 257	PI 512		PI 4847
	3	PI 3	PI 258	PI 513		PI 4848
	:	:	:	:	:	:
	:	:	:	:	:	:
	255	PI 255	PI 510	PI 765		PI 5100

Table 7-78: Program Integers Lookup Table for Mapped Data Types

7.8 Interbus Fieldbus Slave

General Information

Version Note:

Information in this document is based on VisualMotion Toolkit software version 10VRS and PPC-R firmware version GPP 11VRS. GMP 11VRS firmware does not have a fieldbus slave interface, but can be used with the Rexroth PPC-PCI bus interface to allow cyclic and non-cyclic data transfer.

VisualMotion 11 software is downward compatible with GPP firmware, but, depending on the hardware platform selected, the type of fieldbus communication selection may be limited. The following table lists the fieldbus firmware versions and the available fieldbus interfaces for each version.

Fieldbus Interfaces	**PPC-R GPP07VRS**	**PPC-R GPP08VRS**	**PPC-R GPP09VRS/ GPP 11VRS**	**PPC-P GMP09VRS/ GMP 11VRS**
Interbus		●	●	No Fieldbus Slave Support

Table 7-79: Fieldbus Firmware Version and Interface Type

Note: For fieldbus hardware information, refer to the *VisualMotion 11 Project Planning* manual.

PPC-R System Description with a Fieldbus

The PPC-R can operate on a serial fieldbus interface (network) by means of a fieldbus expansion card that communicates with the PPC-R via dual-port RAM. The function of the fieldbus card is similar to that of a network card in a PC: it allows communication with other devices on the network.

In Fig. 7-102, a commonly described fieldbus interface is pictured:

- **Fieldbus Master** - PLC fieldbus interface
- **Fieldbus Slave** - PPC-R fieldbus interface

In this document, we will refer to the PLC as the **fieldbus master** and the PPC-R as the **fieldbus slave**.

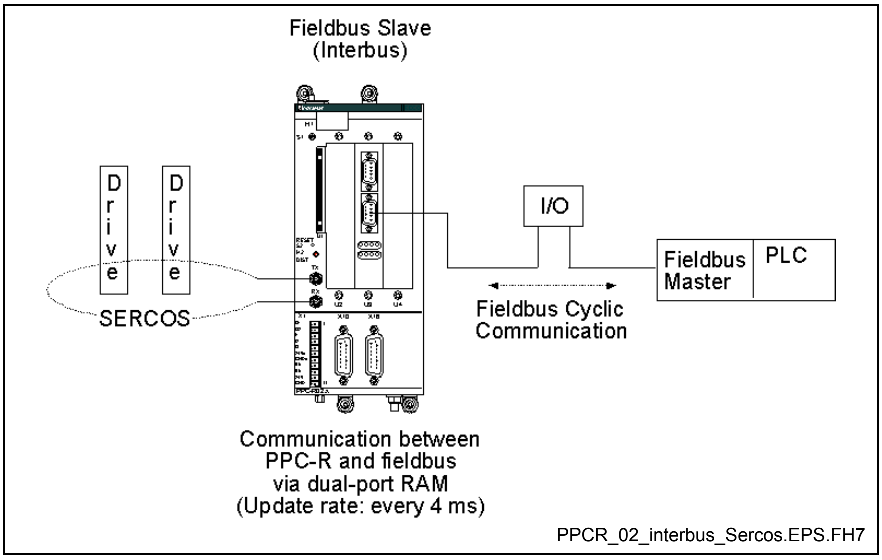


Fig. 7-155: Sample Master/Slave Setup with Fieldbus Card

With the PPC-R, the fieldbus card can be used **only** as a **slave** card in a master/slave setup. When using EtherNet/IP in a VisualMotion 11 system, no other fieldbus interface card (i.e., Profibus, DeviceNet, ControlNet, Interbus) or the MTS-R PLC interface can be used.

The VisualMotion Fieldbus Mapper

In the VisualMotion software package, the Fieldbus Mapper is a tool used to set up fieldbus configuration and data mapping.

Data Transfer Direction (Output vs. Input)

In the VisualMotion Fieldbus Mapper, output and input are always described with respect to the fieldbus master. The definitions for output and input follow:

output: the communication from the PLC to the PPC-R (i.e. from the fieldbus master to the fieldbus slave).
Synonyms for this type of communication: **send** or **write** data.

input: the communication from the PPC-R to the PLC (i.e. from the fieldbus slave to the fieldbus master).
Synonyms for this type of communication: **receive** or **read** data.

Fieldbus Data Channel Descriptions

The Rexroth Interbus fieldbus interface card for the PPC-R supports the following communication channels:

- Cyclic (PD) Channel
- Non-Cyclic (PCP) Channel

Fig. 7-94 shows the possible channel configurations.

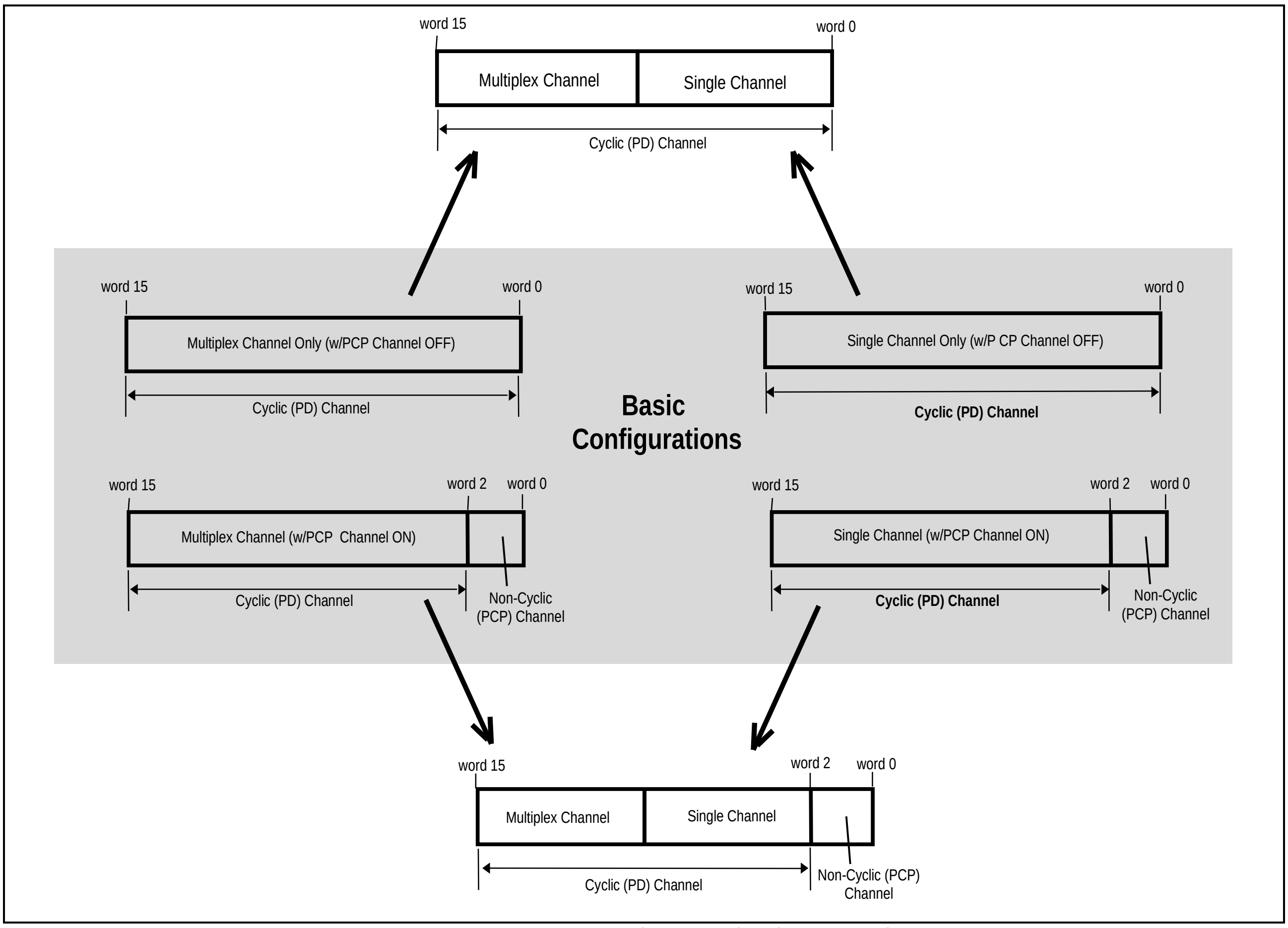


Fig. 7-156: Interbus Channel Configuration Options

Cyclic (PD) Channel

The cyclic (PD) channel, sometimes called the real-time channel, contains user-defined data. This data is stored in two ordered lists (C-0-2600 for input data, C-0-2601 for output data) and transmitted serially over the bus. This data is updated cyclically between the fieldbus master and slave.

The cyclic data channel is limited to 16 input words and 16 output words (provided the non-cyclic channel is turned off). If the non-cyclic (PCP) channel is turned on, it consumes 2 words, thus limiting the cyclic channel to 14 input words and 14 output words. PPC-R data types consume these words in either one-word (16-bit) groups for PPC-R registers or two-word (32-bit) groups for all other data types.

The PPC-R mapping list is scanned every 4 ms and data is sent and received to/from the fieldbus slave board's dual port RAM.

The cyclic data channel can be made up of any combination of the following data types:

- Single Channel
- Multiplex Channel

Cyclic Data: Types and Sizes

The following table outlines the PPC-R data types that can be transmitted via the cyclic channel and the amount of space (in 16-bit data words) that each data type consumes.

Note: The cyclic data mapping lists support only 16- and 32-bit data of the following types for reading and writing:
- Integer
- Float
- Binary (used in control parameters)
- Hex (used in control parameters)
For all other data types (e.g. diagnostic messages - "strings"), use the non-cyclic Channel.

PPC-R Data Type	Data Size (in 16-Bit Words)
Register	1
Program Integer (currently active program ONLY *)	2
Program Float (currently active program ONLY *)	2
Global Integer	2
Global Float	2
Card Parameter	2
Axis Parameter	2
Task Parameter	2
Note: Drive parameters "S" or "P" cannot be transmitted cyclically because of the inherent delay of parameter access over the SERCOS service channel. See ***"Non-Cyclic (PCP) Channel."*** However, if a drive parameter is mapped to an Axis Parameter, that Axis parameter could be used in cyclic data (see description of Axis Parameters 180-196 in the **VisualMotion Reference Manual**).	
*** Important Note:** Integers and floats are shown only for the currently active program. Each time you activate a new progam, the fieldbus reads/writes to the newly-activated program.	

Table 7-80: PPC-R Cyclic Data Types and Sizes

Single Data Types

Single data types are mapped directly in the cyclic mapping ordered lists (C-0-2600, C-0-2601).

Multiplex Data Types (Cyclic Data Channel)

Important: You should use multiplexing **only** if your Interbus master is consistent over the entire cyclic channel!

In some multi-axis applications, 14 or 16 words of cyclic data transfer are not sufficient to meet the requirement of the application.

When insufficient data transfer space is available, multiplex data can be set up within the cyclic channel. One multiplex container acts as a placeholder for multiple possible PPC-R data types (all of the same word size). The currently transmitted PPC-R data type is based on an index value placed in a multiplex control or status word attached to the end of the cyclic list. Depending on the index specified by the master, the multiplex channel permits a different set of data within the cyclic channel to be transferred as current real-time data. Multiplex containers can be added to the input and output lists separately and the input and output indexes can be designated separately (in the control and status words).

Note: Using the multiplex channel reduces the maximum number of usable words for storing control data to 15. The 16th word (or last used word, if fewer than 15 words) is used as the multiplex entry control/status word.

Rexroth
Bosch Group

Note: When using VisualMotion 11 with GPP 7 firmware, a maximum of 15 multiplex containers and a maximum of 180 mapping items can be transmitted in the input or output list. This limitation of mapping objects means that you cannot multiplex all 15 containers with all 32 available indexes (=480 items).
For VisualMotion 11 with GPP 8 or 9 firmware, there is no limitation for multiplexing (each of the first 31 words may be multiplexed with up to 32 indexes).

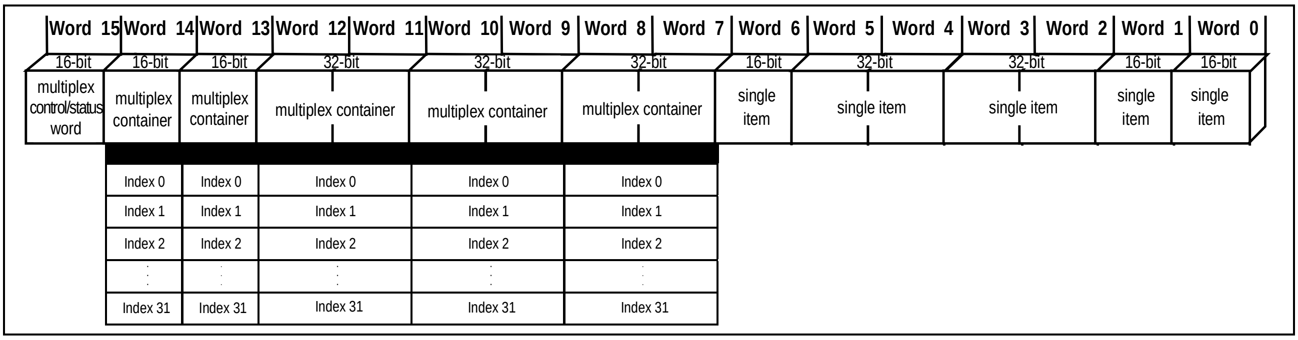


Fig. 7-157: Sample Command (PLC⇒PPC-R) or Response (PPC-R ⇒PLC)

The multiplex control and status words serve to command and acknowledge multiplex data transferred between the fieldbus master and the fieldbus slave. The **control** word is associated with **output** communication (PLC⇒PPC-R). The **status** word is associated with **input** communication (PPC-R⇒PLC). Single data items are not affected by the multiplex control and status words.

Note: For specific information about how the fieldbus master uses the multiplex control and status words, refer to *Multiplexing* on page *7-110*.

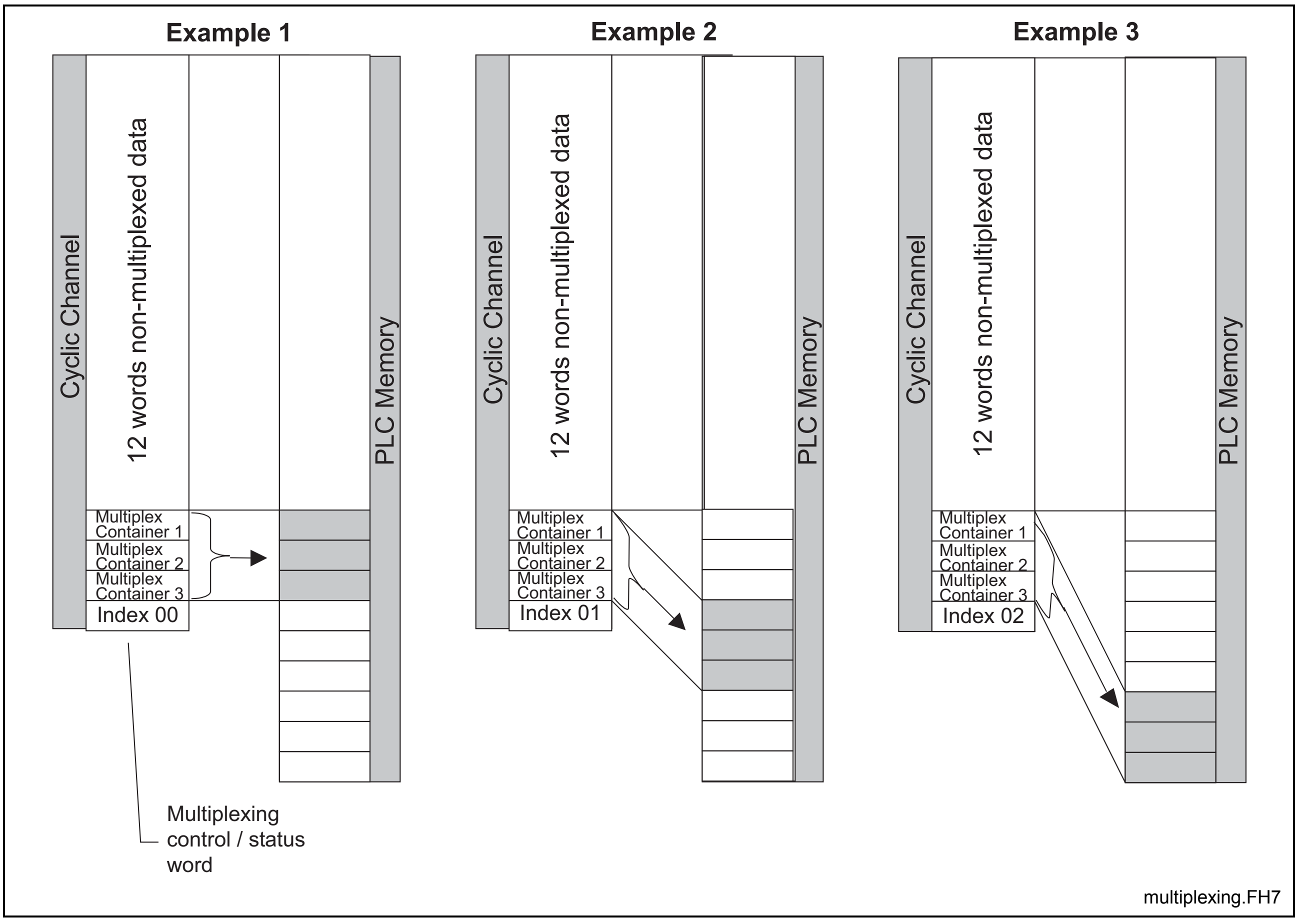


Fig. 7-158: Examples for Reading Data via the Multiplex Channel

Non-Cyclic (PCP) Channel

For Interbus systems using the PPC-R/VisualMotion hardware configuration, the non-cyclic (PCP) channel can be used for parameterization, extended diagnostic information and other “non-urgent” communication.

When enabled, the PCP channel is always fixed at a length of 2 words. If it is not needed, the PCP channel can be disabled, allowing use of those two words for the cyclic channel.

Note: For further explanation of the features supported in the PCP channel, refer to *Non-Cyclic Data Access via the Non-Cyclic (PCP) Channel* on page *7-188*.

Fieldbus Mapper Functionality

Initializing the Fieldbus Mapper from VisualMotion 11

1. Open an existing program or create a new program. You must be using PPC-R hardware with GPP firmware to use the Fieldbus Mapper described in this document.
2. Select ***Commission*** ⇒ ***Fieldbus Mapper***. The main Fieldbus Mapper window appears (refer to Fig. 7-106).

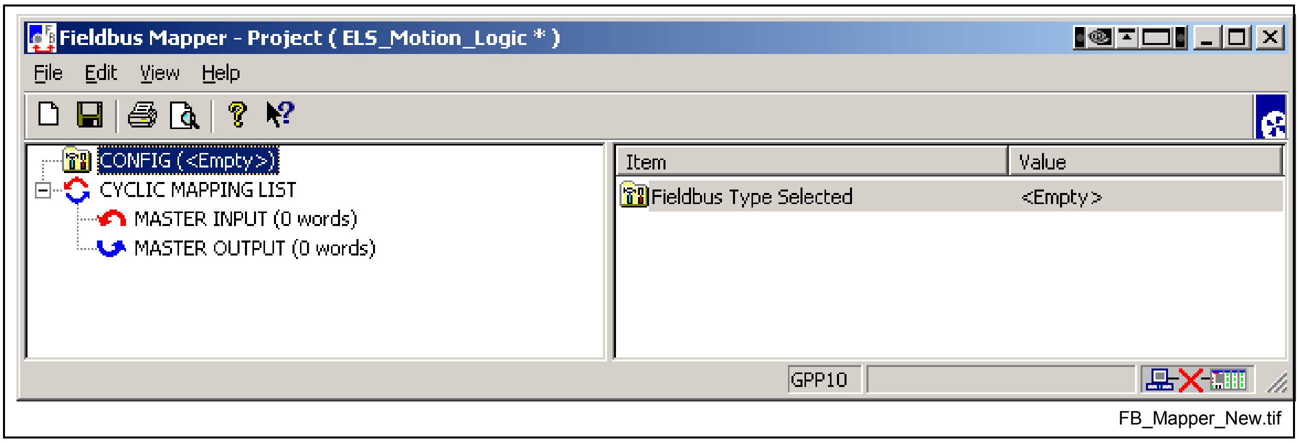


Fig. 7-159: FBMapper – Project Window (Empty)

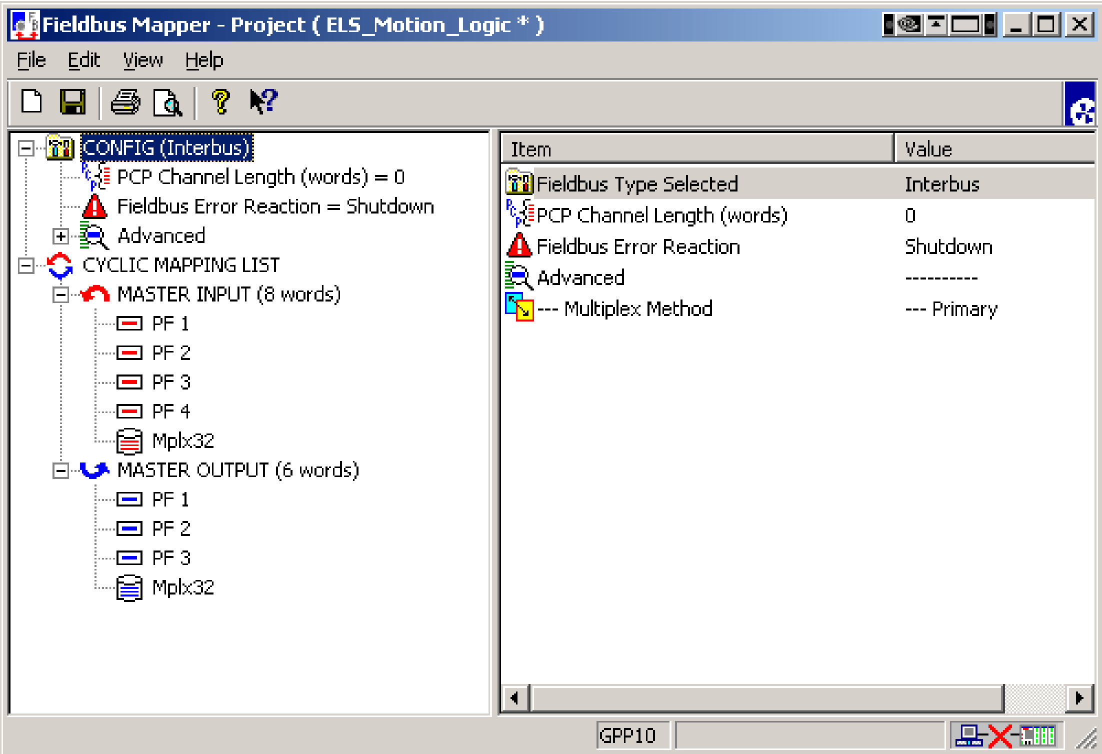
Creating a New Fieldbus Mapper File

1. Click **or select *File* ⇒ *New*.**
   A “setup wizard” goes through three steps:
   – Fieldbus Slave Definition
   – Fieldbus Slave Configuration
   – Cyclic Data Configuration
2. Enter the information requested in the setup windows. For more details on each step, refer to *Fieldbus Slave Definition*, *Fieldbus Slave Configuration*, and *Cyclic Data Configuration* for detailed information about each configuration step.
3. Save the file (automatically has a ***.prm** extension).

Importing a Fieldbus Mapper File

A Fieldbus Mapper file can be imported from another project. To import the file:

1. Select ***File* ⇒ *Import***.
2. Browse to find the desired file (***.prm** extension).
3. Click ***Open***. The main Fieldbus Mapper window appears, which lists the configuration information. Refer to Fig. 7-160.



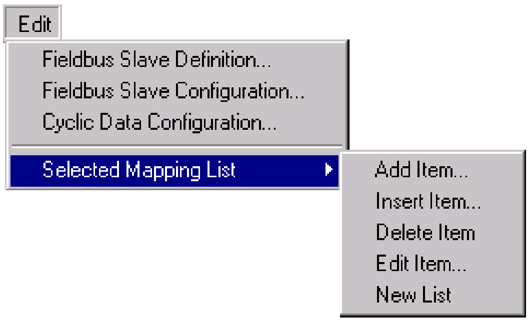
FB_Mapper_Main_IB.tif

Fig. 7-160: FBMapper – Project Window (Complete)

4. From the Fieldbus Mapper main window, double-click on the specific item to be edited. The corresponding setup window appears.

- Or -

Select the item to edit from the ***Edit*** menu (refer to Fig. 7-161). For more information about each step, refer to *Fieldbus Slave Definition*, *Fieldbus Slave Configuration*, and *Cyclic Data Configuration* for detailed information about each configuration step.



FB_Mapper_Edit_Menu.tif

Fig. 7-161: Fieldbus Mapper Edit Menu

Note: You can also directly add, insert, delete, edit an item, or create a new list by:

- clicking on the item to be edited in the main Fieldbus Mapper window and selecting the desired function under ***Edit ⇒ Selected Mapping List***

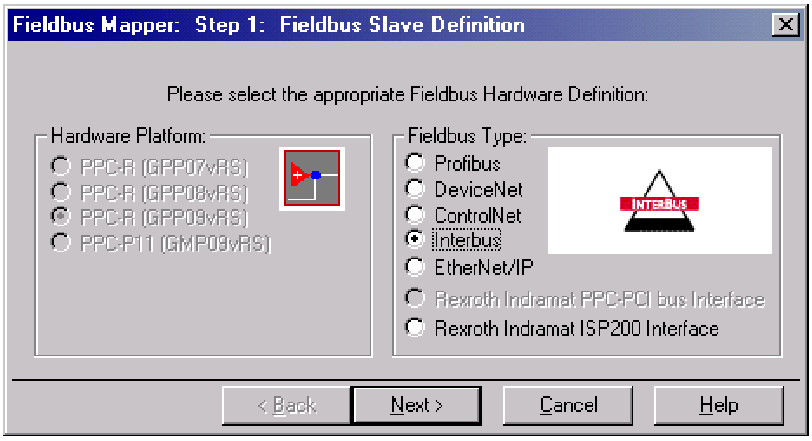
OR

- right-clicking on an item to display a menu of functions

Rexroth Bosch Group

Fieldbus Slave Definition

From the Fieldbus Slave Definition window, select **Interbus** as the Fieldbus Type (refer to Fig. 7-162). Refer to Table 7-79 for a list of the available hardware platforms for Interbus. The hardware platform can only be selected when the project is in Service mode.

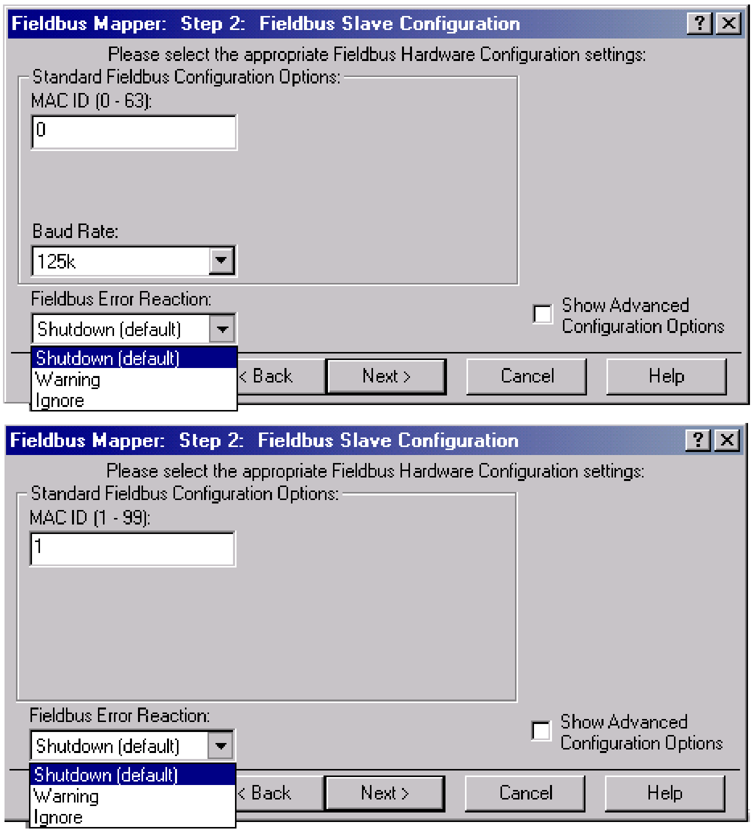


FB_Hardware_IB.tif

Fig. 7-162: Fieldbus Slave Definition Window

Fieldbus Slave Configuration

The Interbus Fieldbus Slave Configuration window is shown in Fig. 7-163 below.



FB_Slave_Config.tif

Fig. 7-163: Fieldbus Slave Configuration

Fieldbus Error Reaction

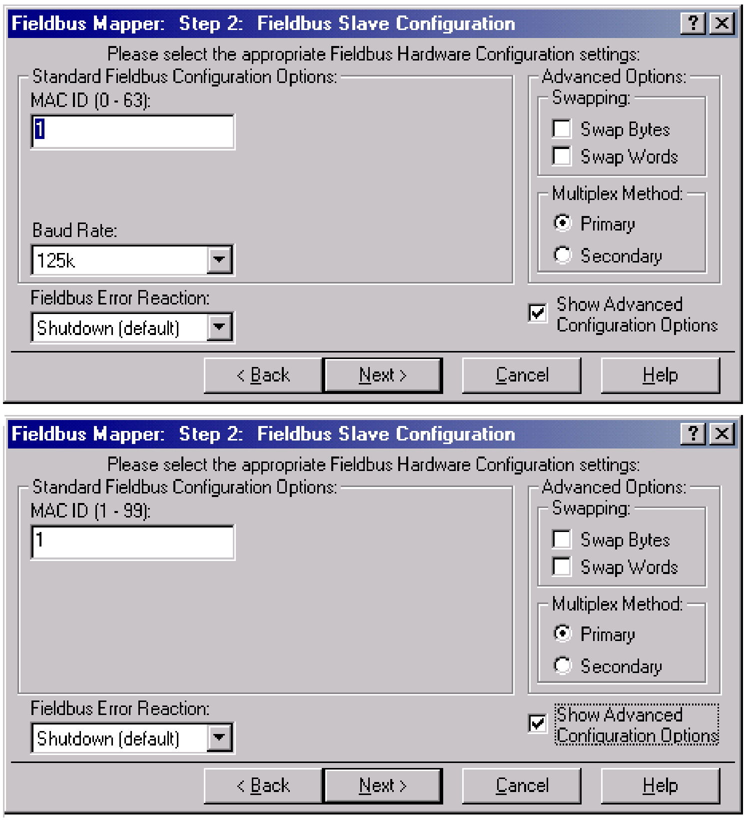
Set the Error Reaction to Shutdown (default), Warning or Ignore. Refer to *Fieldbus/PLC Cyclic Read/Write Monitoring* on page 7-108 for detailed information about each setting.

PCP Channel Length

The PCP (non-cyclic) channel can be set to 0 words (Off) or 2 words (On).

Advanced Configuration Options

The *Advanced Options* field is displayed if the checkbox next to *Show Advanced Configuration Options* is checked (refer to Fig. 7-164 below). In most cases, the default options should apply.



FB_Slave_Config_Adv.tif

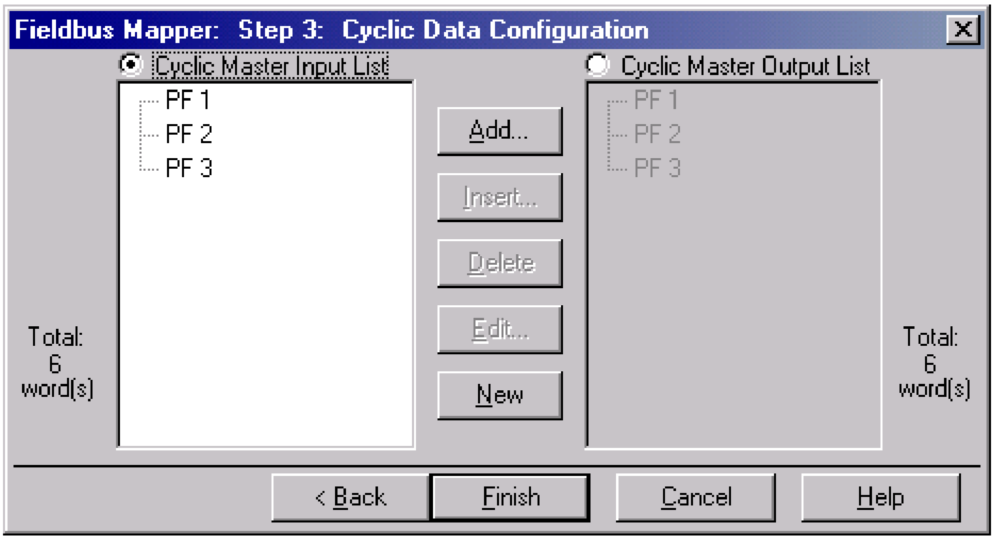
Fig. 7-164: Fieldbus Slave Configuration: Advanced

- **Multiplex Method:** select Primary or Secondary (Primary is the default). Select Secondary only if you have a consistent fieldbus master. Refer to *Multiplexing* on page 7-110 for detailed information about each method.

Cyclic Data Configuration

An example of the Cyclic Data Configuration window is shown in Fig. 7-165 below. If you are editing an existing Fieldbus Mapper file, the list will probably contain more items.

First, you must select the Cyclic Input List (from PPC-R to PLC) or the Cyclic Output List (from PLC to PPC-R).

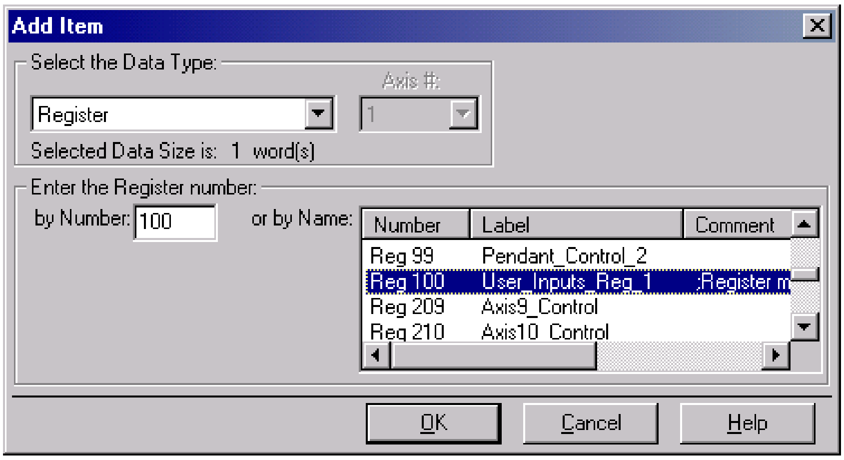


Cyclic_Data_Config.tif

Fig. 7-165: Cyclic Data Configuration

Adding an Item to the List

1. Select the Cyclic Input List or the Cyclic Output List.
2. Click ***Add***. The window in Fig. 7-166 below appears. Select the Data Type (for example, Register).



Cyclic_Data_Add.tif

Fig. 7-166: Add Item to Cyclic Data

Note: Registers and 16-bit Multiplex Containers (used only for Registers) require one data word (16 bits), and all other data types require two data words (32 bits) of space.

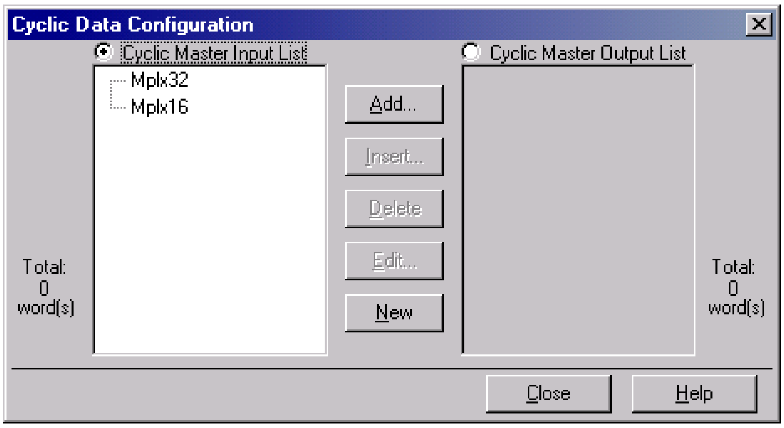
3. Enter the required information (for example Register Number) or select it from the list below. Only the available data types for your designated VisualMotion hardware setup and fieldbus type are listed.

Note: If your project is in Service mode and you check the box next to “Get Latest (Online),” the data type label list is updated based on your firmware version and the currently active program.

4. Click ***OK*** to add the selected item to the list.

Adding Multiplex Containers to the List

1. Select the Cyclic Input List or the Cyclic Output List.
2. Click ***Add***.
3. In the *Add Item* window under *Select the Data Type*, select Multiplex Container 16-bit (for Registers) or Multiplex Container 32-bit (for all other data types).
4. Click ***OK*** to add the Multiplex Container to the List. The window in Fig. 7-167 below is an example where a 16-bit Multiplex Container and a 32-Bit Multiplex Container have been added.



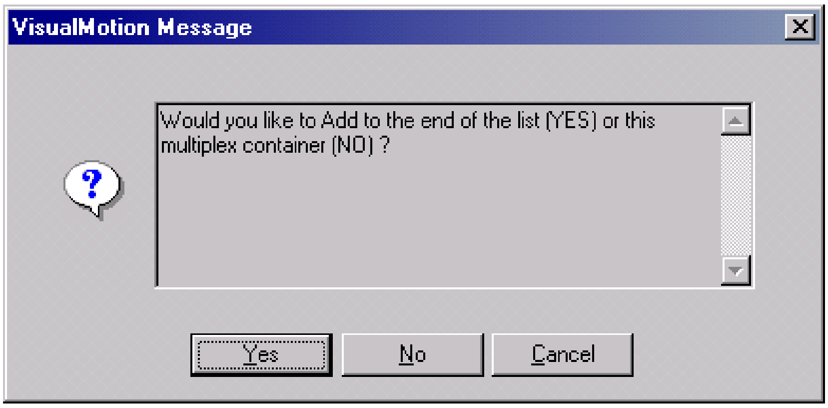
Cyclic_Data_Config_mltplx.tif

Fig. 7-167: Cyclic Data Configuration, Multiplex Containers

Note: At this point, the Multiplex Containers do not yet contain any items. To add multiplex items, refer to steps below.

Adding Items to an Empty Multiplex Container

1. In the Cyclic Data Configuration window, select the multiplex container to which you want to add items.
2. Click ***Add***. The window in Fig. 7-168 below appears. Because it is unclear whether you would like to add to the list or to the multiplex container, the Fieldbus Mapper is requesting clarification.



Add_Multiplex.tif

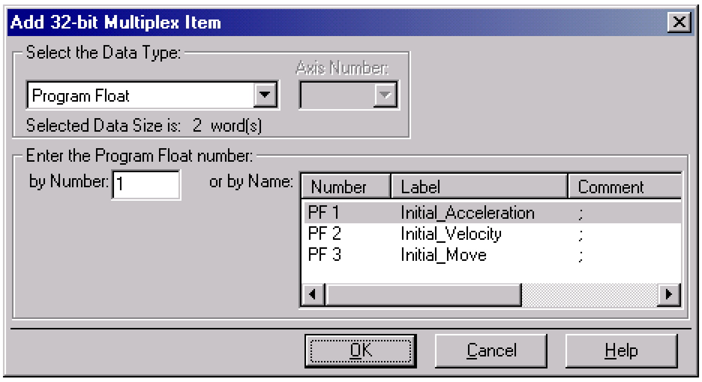
Fig. 7-168: Add Item or Multiplex Item

Note: For subsequent items, highlight any of the indexes within the multiplex container before clicking *Add*, and the Fieldbus Mapper will know you want to add to that container.

3. To add to the selected multiplex container, click ***No***. The window in Fig. 7-169 below is an example for adding a 32-bit multiplex item.
4. Select the desired item to be added to the multiplex container.

Note: In addition to the data types that can be added to the multiplex list, an empty item called **Multiplex Empty Item** is available to fill a space within the multiplex container, if nothing is to be mapped to a particular index.

5. Click ***OK***. The item is automatically placed in the multiplex container as the next unassigned index item (e.g. the first item is index 00, the last is index 31).
6. Repeat for as many items as you want to add to the multiplex container, up to 32 items.



Add_Multiplex_Item.tif

Fig. 7-169: Adding a Multiplex Item to the Container (32-bit example)

Editing the Cyclic Data Lists

To make changes to an existing list, use the following buttons:

Button	Function
Add...	Inserts a new item at the end of the list.
Insert...	Inserts a new item into the list directly before the selected item.
Delete	Removes the selected item from the list.
Edit...	Allows editing of the selected item. (To edit a list item, you may also double-click on it.)
New	Clears up the current list.

Table 7-81: Button Functions in the Cyclic Data Configuration Window

Additional Functions

Several additional functions are available in the Fieldbus Mapper:

Menu Item	Function
Print	Print the current fieldbus configuration data.
Print Preview	Preview the printout of the current fieldbus configuration data
Print Setup	Configure printer settings

Table 7-82: Additional Functions

Getting the Fieldbus Configuration from the PPC

After getting the fieldbus configuration from the PPC, the following information is detected by the system and appears in the configuration list:

- Fieldbus Type Found
- Fieldbus FW (Firmware) Version
- GPP Control FW (Firmware) Version

An example is shown in Fig. 7-170 below.

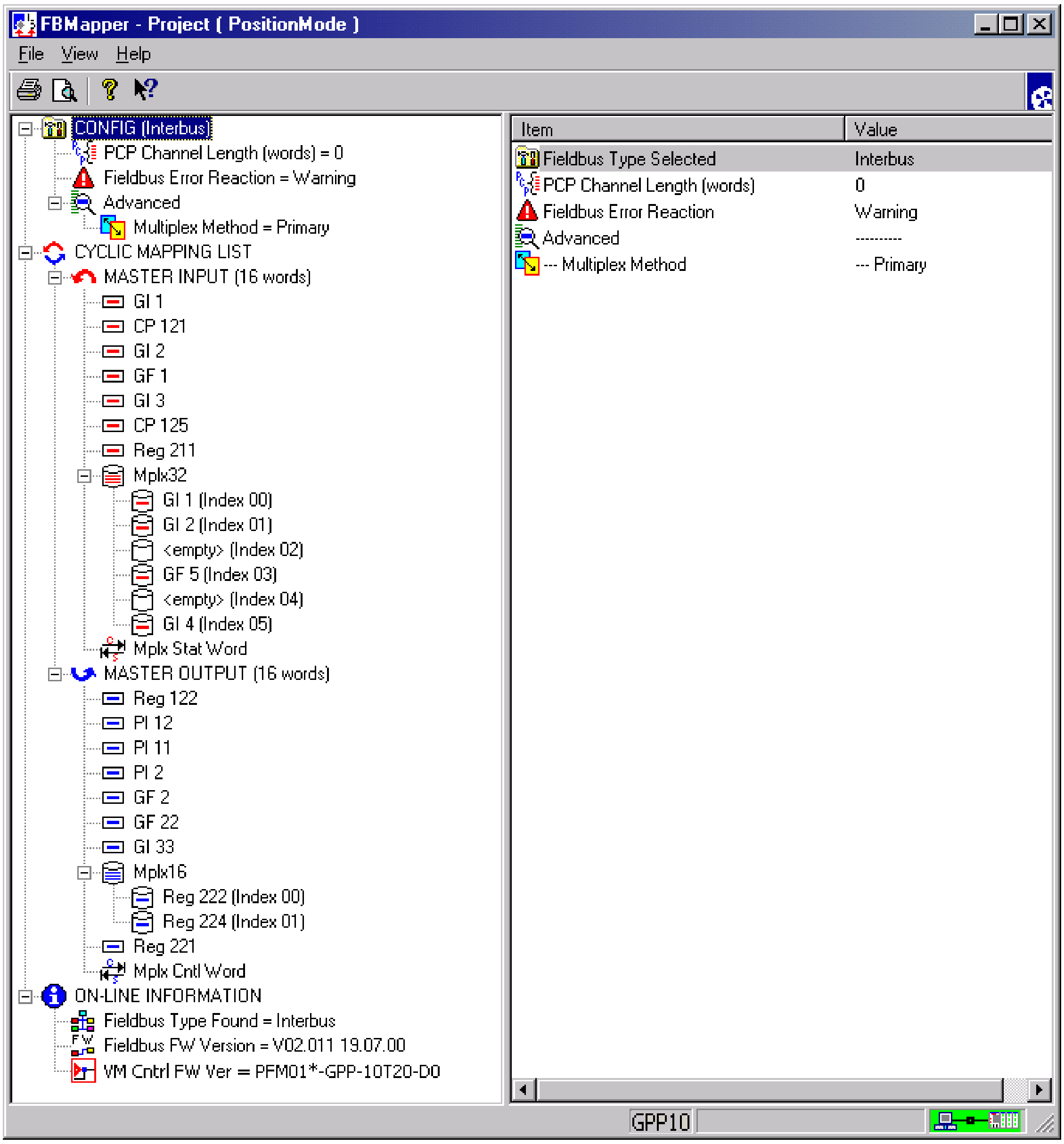


Fig. 7-170: Online Fieldbus Configuration Information

Information for the GPP Programmer

Fieldbus Status

VisualMotion Register 19 holds the information for "Fieldbus Status." The register information can be referenced in a VisualMotion application program to respond to the status of each bit. The use of these bits is application-dependent.

Table 7-48 below contains the bit assignment for the fieldbus status. The assigned bits are labeled with "x" and the bit number in the second row. Unassigned bits are labeled with "---."

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
---	x15	---	---	---	---	---	---	---	---	---	x5	x4	---	x2	x1

Table 7-83: Bit Assignment for VisualMotion Register 19

Bit Definitions

x1, x2 Status bits for the internal DPR (Dual-Port RAM) communication between the fieldbus slave and the PPC-R:

x1: FB Init. OK , LSB (least significant bit)

x2: FB Init. OK, MSB (most significant bit)

The bit combinations for x1 and x2 are as follows:

Bit 2 (PPC-R)	Bit 1 (Fieldbus)	Description
0	0	A reset has been executed on the DPR, **or** neither the PPC-R nor the fieldbus card have initialized the DPR.
0	1	The DPR is initialized by the fieldbus card, but **not** yet by the PPC-R.
1	0	The DPR initialization is complete. DPR has been initialized by the fieldbus card and PPC-R. Fieldbus to PPC-R communications system is ready.
1	1	Fieldbus to PPC-R communications system is ready.

Table 7-84: Possible Settings for Bits 1 and 2, Status Bits for DPR Communication

x4 Status bit for the active bus capabilities of the fieldbus slaves (FB Slave Ready)

0--> The fieldbus slave is not (yet) ready for data exchange.

1--> The fieldbus slave can actively participate on the bus.

x5 Status bit for the non-cyclic (PCP) channel (Non-Cyc Ready)

0--> The non-cyclic channel cannot (yet) be used.

1--> The non-cyclic channel is ready for use by the fieldbus master.

x15 Status bit for the cyclic data output (Cyclic Data Valid):

0--> The cyclic data outputs (coming in to the PPC-R) are INVALID.

1--> The cyclic data outputs (coming in to the PPC-R) are VALID. The system looks for this bit to be 1 before allowing data transfer.

This bit is monitored for the Fieldbus Error Reaction. Whenever this bit goes to 0 after a fieldbus card was initially found by the PPC-R, the selected Error Reaction (system shutdown, error message, or ignore) is initiated. Refer to *Fieldbus/PLC Cyclic Read/Write Monitoring* on page *7-108* for an explanation of the Fieldbus Error Reaction setting.

Fieldbus Diagnostics

VisualMotion Register 20 holds the information for "Fieldbus Diagnostics."

Table 7-50 below contains the bit assignment for the diagnostics. The assigned bits are labeled with "x" and the bit number in the second row. Unassigned bits are labeled with "---."

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
X16	x15	x14	x13	---	---	---	---	---	---	---	---	---	---	---	---

Table 7-85: Bit Assignment for VisualMotion Register 20

Bit Definitions

x13 - x16 Identification of the fieldbus interface card (FB Card Found)

The bit combinations for x13, x14 and x15 are as follows:

Bit 16	Bit 15	Bit 14	Bit 13	Fieldbus Type
0	0	0	0	<NO CARD>
0	0	0	1	<Not Defined>
0	0	1	0	Interbus
0	0	1	1	DeviceNet
0	1	0	0	Profibus
0	1	0	1	ControlNet
0	1	1	0	<Not Defined>
0	1	1	1	EtherNet/IP (10 MB)
1	1	1	1	Indramat PLC Interface

Table 7-86: Identification of the Fieldbus Interface

Fieldbus/PLC Cyclic Read/Write Monitoring

Monitoring of Fieldbus read/write capabilities to the cyclic channel are associated with three parameters:

- C-0-2611 Fieldbus/PLC Cyclic Channel: Current Number of Misses– displays the current number of transfers to/from the cyclic channel.
- C-0-2612 Fieldbus/PLC Cyclic Channel: Peak Number of Misses – displays the maximum number of missed transfers to/from the cyclic channel.
- C-0-2613 Fieldbus/PLC Cyclic Channel: Timeout Counter – displays the number of timeouts in the cyclic channel. If after 4 ms, the Cyclic Mapping Lists are not successfully transmitted, a "miss" is noted.

For more information about these parameters, refer to the *VisualMotion 11 Functional Description* manual.

Fieldbus Error Reaction

Note: The Fieldbus Error Reaction setting is active only in Sercos Phase 4. In all other Sercos phases, it will be inactive.

You can select how you would like the PPC-R system to react in case of a fieldbus error. This reaction can be set in the "Fieldbus Slave Configuration" window, using the combo box labeled "Fieldbus Error Reaction."

Three options are available for the Error Reaction setting. Depending on the selected setting, the value 0, 1, or 2 is stored in Parameter C-0-2635:

Setting	Value in Parameter C-0-2635
Shutdown	0 (default)
Warning Only	1
Ignore	2

Table 7-87: Parameter C-0-2635 Values for Error Reaction Settings

Fieldbus Mapper Timeout

The Fieldbus Mapper continually scans the system for sufficient resources to process the cyclic data mapping lists (2600 and 2601 lists). If 10 out of 10 consecutive attempts of the mapping list updates are incomplete, the system is considered to have insufficient resources and the selected error reaction is evoked, as follows:

If "Shutdown" (0) is set in Parameter C-0-2635, the following error is generated from the PPC-R card: **520 Fieldbus Mapper Timeout**

If "Warning Only" (1) is set in Parameter C-0-2635, the following error is generated: **209 Fieldbus Mapper Timeout**

If "Ignore" (2) is set in Parameter C-0-2635, the system will update as resources become available, but there is no way to monitor whether or not updates actually occur.

Lost Fieldbus Connection

Register 19, bit 4 indicates the status of the fieldbus. Refer to *Fieldbus Status* on page *7-107* for more specific bit information. The system monitors this bit and evokes the selected error reaction if the bit is low (0), after a fieldbus card is found. A typical situation that will cause this condition is the disconnection of the fieldbus cable from the fieldbus card.

If "Shutdown Control" (0) is set in Parameter C-0-2635, the following error is generated from the PPC-R (active in Sercos Phase 4 only): **519 Lost Fieldbus Connection**

If "Warning Only" (1) is set in Parameter C-0-2635, the following error is generated (active in Sercos Phase 4 only): **208 Lost Fieldbus Connection**

If "Ignore" (2) is set in Parameter C-0-2635, there is no noticeable reaction when Register 19 status bits go low, unless the GPP application program is customized to evoke a special reaction.

Troubleshooting Tip:

If a fieldbus card is not found on the system, the Error Reaction setting will be ignored. If you have a fieldbus card and the Error Reaction is not responding as expected, the system may not "see" your fieldbus card.

Information for the PLC Programmer

Important: The fieldbus master's access of the cyclic channel must be consistent over the entire channel length in order to establish reliable multiplexing communications.

Multiplexing

Primary Multiplex Method (for Inconsistent Masters)

Explanation of the Master Consistency Problem

The PPC-R fieldbus slave interfaces can guarantee consistency, however, some fieldbus masters can only guarantee byte, word or double word consistency. If the master is only word-consistent, it is possible that the master cannot transfer the data and the control word of one multiplex index consistently from the PLC to the fieldbus. Therefore, it is necessary to have a second multiplex method where both input data and output data require the handshake bits to update via the fieldbus.

Fig. 7-112 below illustrates the control word definition for the Primary Multiplex Method.

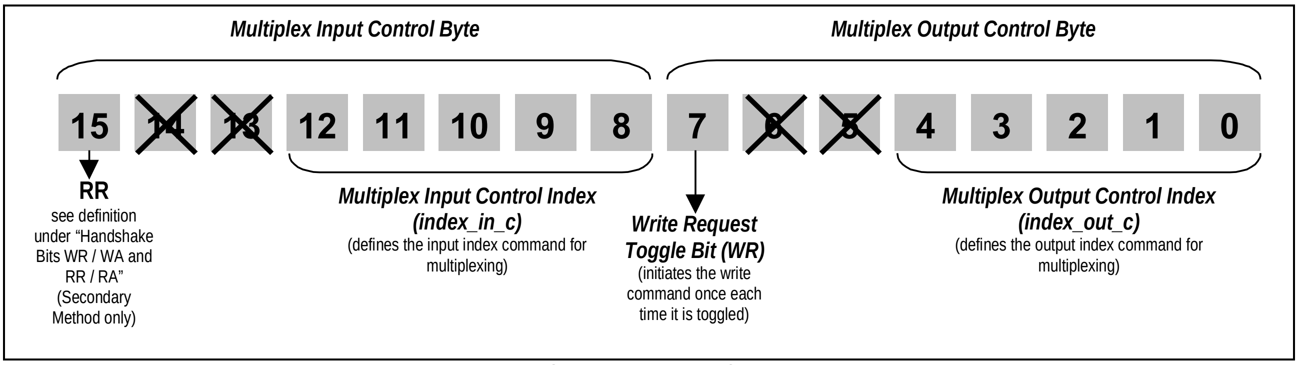


Fig. 7-171: Control Word Definition, Primary Multiplex Method

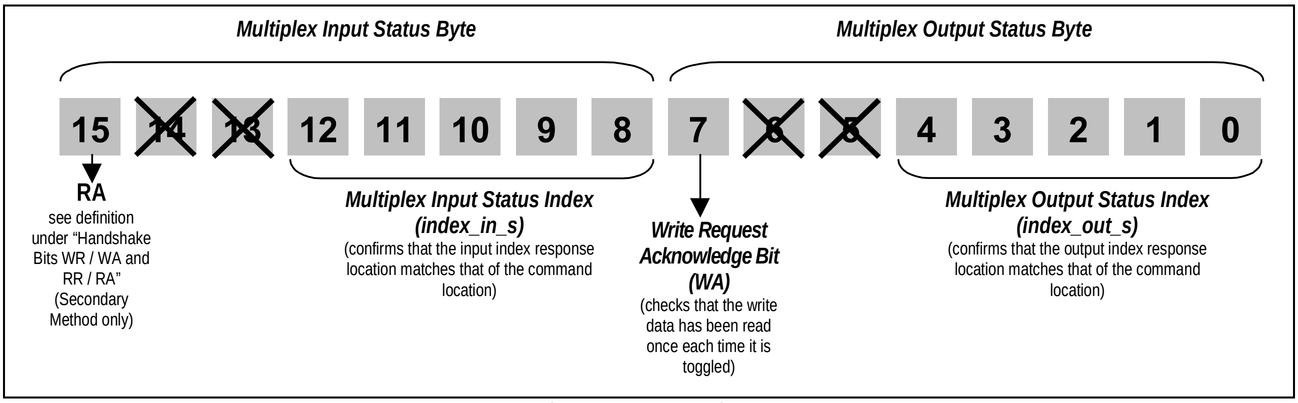


Fig. 7-172: Status Word Definition, Primary Multiplex Method

The Primary Multiplex Method has the following features:

- You can transfer a different index from master to slave as from slave to master.
- The handshake bits for both reading and writing of this multiplex channel make the multiplexing possible on inconsistent systems (masters).

Handshake Bits WR and WA

WR and WA are handshake bits that allow the controlled writing of data via the multiplex channel. WR and WA control the data transfer for writing data_out (data send from master to slave).

WR == WA:

- tells the master that the slave has received the last multiplex data_out. The master can now send new data_out.
- tells the slave to do nothing, because the master has not yet put new consistent data_out on the bus.

WR! = WA:

- tells the slave to do something, because the master has now put consistent new data_out on bus.
- tells the master to do nothing, because the slave has not yet received the latest multiplex data_out.

Handshake Bits RR and RA

RR (Read Request) and RA (Read Acknowledge) are handshake bits that allow a controlled data transfer and use of the multiplex channel on inconsistent masters. RR and RA control the data transfer for reading data_in (data send from slave to master).

RR == RA:

- tells the master that the slave has sent the requested data_in. The master can now read the data_in and request new data_in.
- tells the slave to do nothing, because the master has not yet put new consistent data on the bus.

RR != RA:

- tells the slave to put new data_in on the bus, because the master requests new data_in.
- tells the master to do nothing, because the slave has not yet put the latest requested multiplex data_in on the bus.

Master Communications (Primary Multiplex Method)

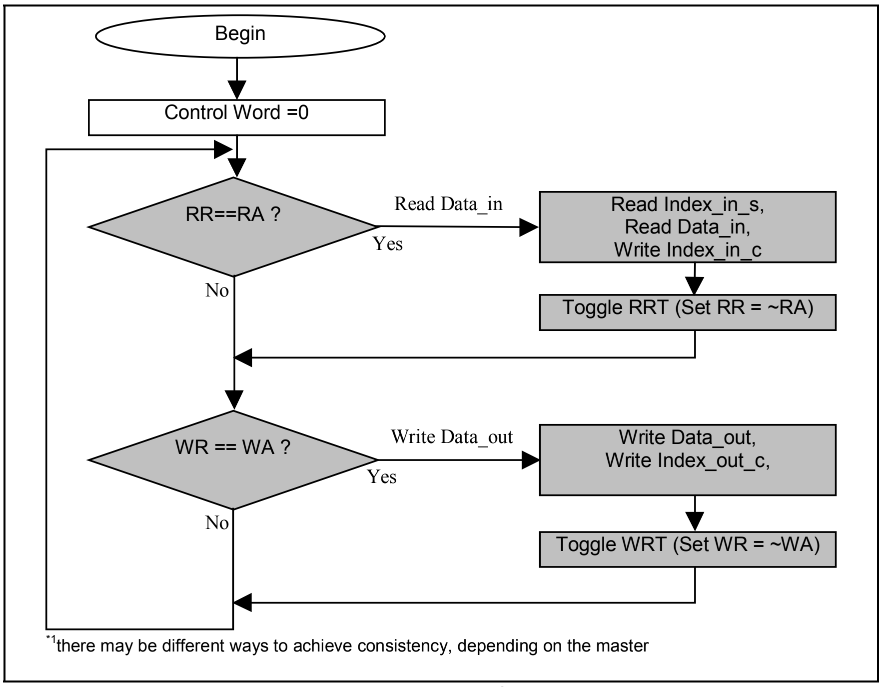


Fig. 7-173: Primary Multiplex Method, Master Communications

For some masters, it could be enough to first write data and then the control word. For other masters, you may have to implement a delay time (this time could be different from master to master) before writing WR = ~WA.

Secondary Multiplex Method (for Consistent Masters only)

Important: You should use the Secondary Multiplex Method **only** for a master that is consistent over the entire cyclic channel. The Primary Multiplex Method is available for inconsistent masters. Refer to *Secondary Multiplex Method (for Inconsistent Masters)* on page *7-113*.

The advantage of the Secondary Method is easier handling of input data for consistent masters.

Note: The meanings of the control and status words are the same as for the Primary Multiplex Method. The only difference is the toggle bits RR and RA, which are used only in the Primary Method.

Control Word and Status Word

Control Word

The control word is transferred in the multiplex channel from master to slave. It tells the slave in which index the data is being transferred from master to slave and in which index the data is requested from slave to master.

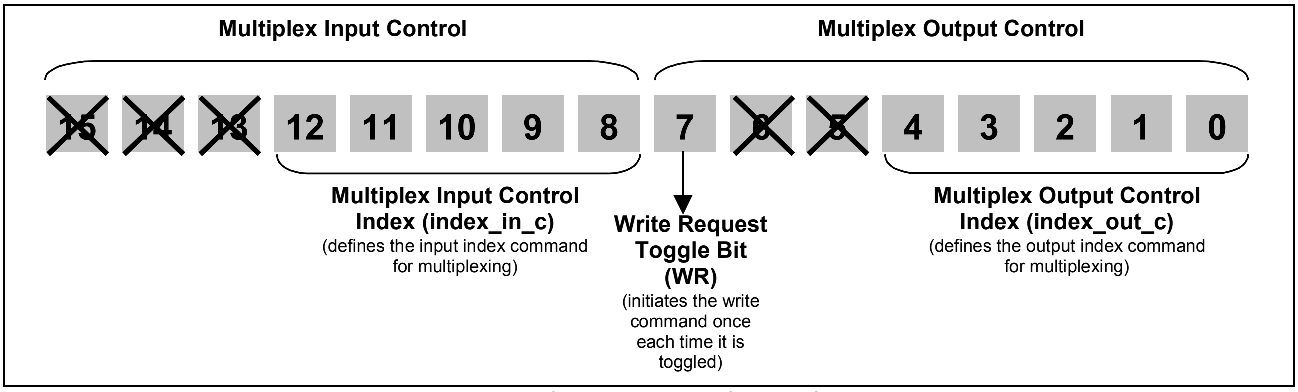


Fig. 7-174: Control Word Definition, Secondary Multiplex Method

Index_out_c: tells the slave in which index the data are transferred from master to slave (out = master -> slave, _c = element of control word).

Index_in_c: tells the slave in which index the data is requested from slave to master (in = slave -> master, _c = element of control word).

WR (Write Request): handshake bit (refer to meaning of WR and WA).

Note: Input data via the Multiplex Channel is continually being updated.

Status Word

The status word is transferred in the multiplex channel from slave to master. It acknowledges the written index and the requested index.

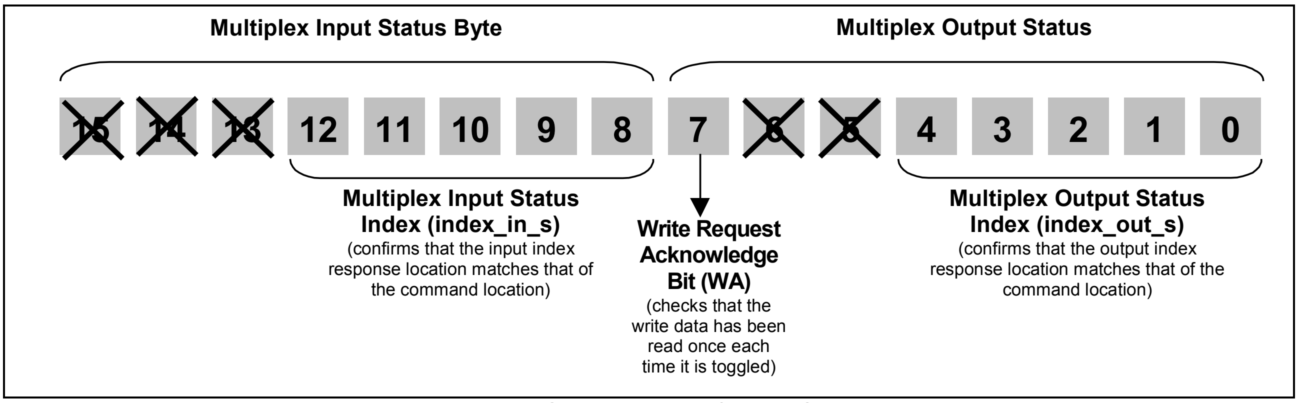


Fig. 7-175: Status Word Definition, Secondary Multiplex Method

- **Index_out_s:** acknowledges index written by the master (out = master -> slave, _s = element of status word).
- **Index_in_s:** tells the master which index is transferred from slave to master in the actual process data cycle (in = slave -> master, _s = element of status word).
- **WA (Write Acknowledge):** Handshake bit (refer to meaning of WR and WA under the Primary Multiplex Method).

Master Communications (Secondary Multiplex Method)

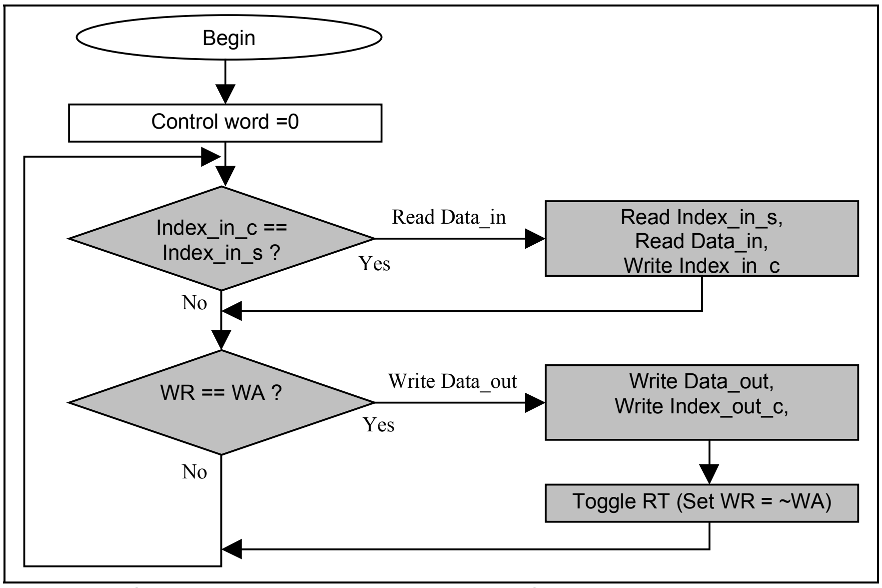


Fig. 7-176: Secondary Multiplex Method, Master Communications

Programming Example

To aid in implementing the multiplex function in a PLC program, the following flow chart shows two ways of reading and writing data. Reading and writing can be executed separately, which allows the input data to be updated about 30% faster. The “Read Data” example would be placed at the beginning of a PLC program the “Write Data” example at the end.

Combined reading and writing makes the PLC program simpler, especially when using the same index for both transfer actions.

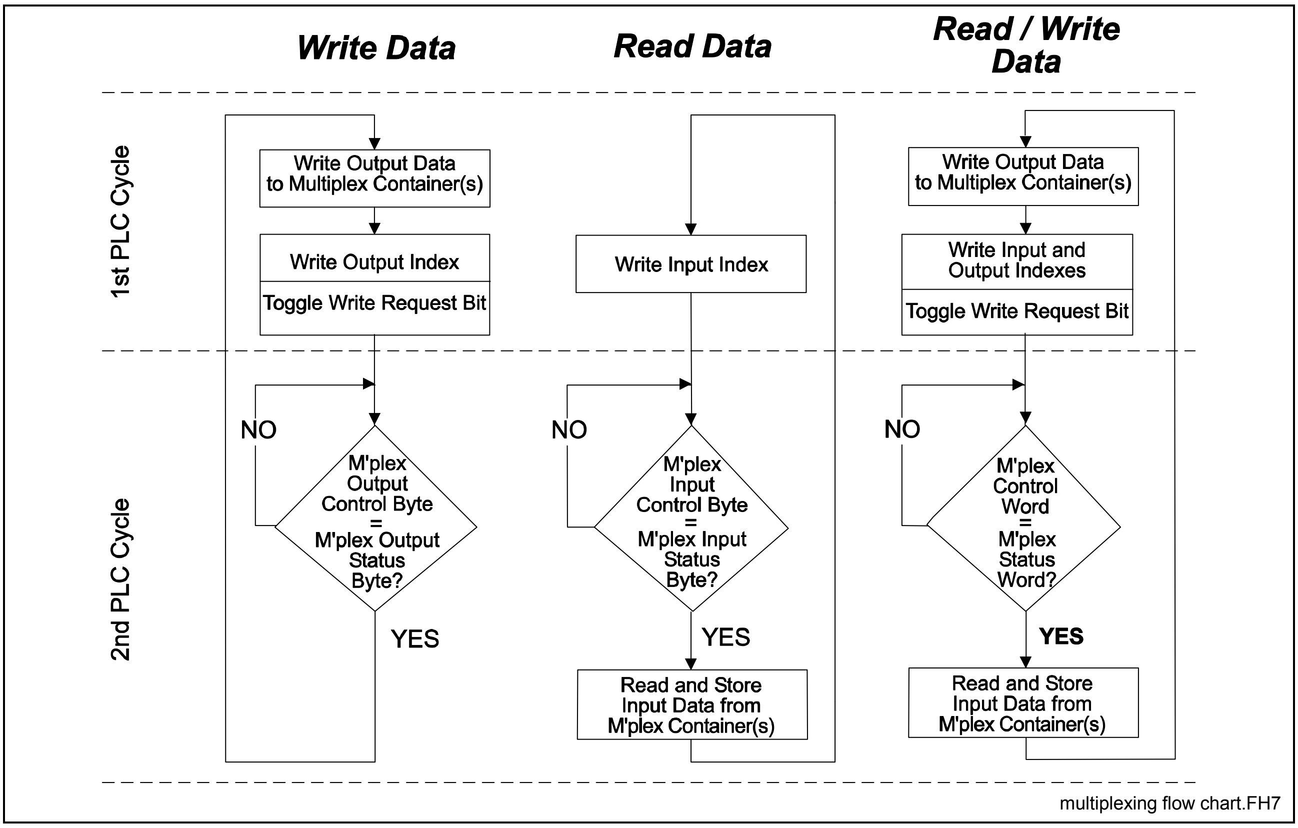


Fig. 7-177: Flow Chart of Multiplex Programming Examples (Secondary Method)

Non-Cyclic Data Access via the Non-Cyclic (PCP) Channel

To support the configuration of drives and the access to parameters via Interbus, Rexroth supports the PCP channel.

Note: The PCP Channel is fixed at a length of 2 words when enabled.

To read or write a VisualMotion data type non-cyclically, a special set of pre-defined objects is used in the PCP channel. Refer to the documentation provided by the fieldbus manufacturer for access and support of the PCP channel.

The following methods for transferring data are available in the PCP channel:

- Mapped Data
- Data Exchange Objects

Mapped Data

Mapped data is the most powerful feature of the PPC-R non-cyclic fieldbus interface. Through mapped data, the user has access to virtually every PPC-R data type over the fieldbus. It is easy to implement from the PLC side and requires no setup on the PPC-R side.

To access a data type over the fieldbus, it has to be specified by an address that consists of an index and a subindex. The index and subindex for each data type can be calculated by a formula (refer to *Accessing Mapped Data* on page *7-125*).

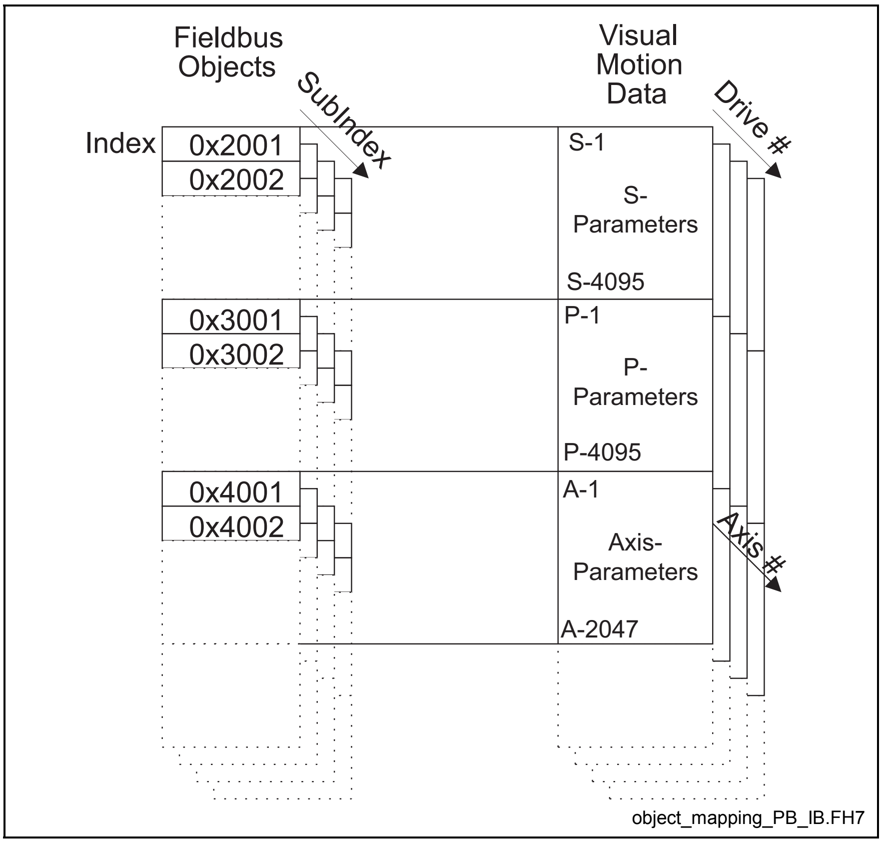


Fig. 7-178: Mapped Data

Mapped data can be used with the following parameters and values:

- S-Parameters (SERCOS Drive S-Parameters)
- P-Parameters (SERCOS Drive P-Parameters)
- A-Parameters (PPC Axis Parameters)
- C-Parameters (PPC C System parameters)
- T-Parameters (PPC Task parameters)

size and format depend on parameter *,1

PF-Values (PPC Program Float data, 32 bit – 2 words, IEEE format) *,2

GI-Values (PPC Global Integer data, 32 bit – 2 words) *,2

GF-Values (PPC Global Float data, 32 bit – 2 words, IEEE format) *,2

PI-Values (PPC Program Integer data, 32 bit – 2 words) *,2

Reg.-Values (PPC Register data, 16 bit – 1 word) *,3

Data Exchange Objects (0x5E70 – 0x5E73) (embedded ASCII Protocol)

*You may notice that parameters accessed via the non-cyclic (Parameter) channel are not always the same size as reported from the attribute field. This is so that the data sizes correspond with the way the different data types are handled in the cyclic channel (Registers are always set to 16-bit size and Parameters are cast to 32-bit size, even if they actually use less space).

7. When **writing** mapped data to a VisualMotion Parameter, you must send the size data corresponding to that of the attribute field within the parameter.

    a.) For 32-bit parameters, you must send a data size of 32 bits (otherwise, VM error #07 is returned).

    b.) For 16-bit parameters, you must send a data of size 16-bits. If, for this case, you send data of size 32 bits, one of the following occurs:

        i.) For parameters of type 16-bit unsigned, only the Low word is stored, and the High word is ignored.

        ii.) For parameters of type 16-bit signed, bits 0-14 of the low word along with the sign bit #31 are used, and the remaining bits are ignored.

    c.) For String Parameters (e.g. S-0-0142), you must send the size of the string to write.

d.) All other Parameter Types (list parameters, command parameters, etc), are not supported for mapped data.

When reading mapped data from a VisualMotion Parameter, there are 3 possible cases of sizes returned:

a.) If the parameter type is a string, you receive the number of bytes corresponding to the length of the string.

b.) If the parameter is 32-bit or less, you receive a cast 32-bit value for this parameter. This implies that 16-bit parameters are returned as cast in to 32-bit values.

c.) All other parameter types (e.g. list parameters, command parameters, etc.), are not supported for mapped data.

8. When writing mapped data to a VisualMotion Program Float, Program Integer, Global Float, or Global Integer, the data size must be 32-bits (2 words). Any other size returns a VM error #07 (Invalid Data Format).

   When reading mapped data from a VisualMotion Program Float, Program Integer, Global Float, or Global Integer, the data size returned is always 32-bit (2 words).

9. When writing mapped data to a VisualMotion Register, the data must be 16-bits (1 word). Any other size returns a VM error #07 (Invalid Data Format).

   When reading mapped data from a VisualMotion Register, the data size returned is always 16-bit (1 word).

Object Index

The index refers to the particular fieldbus slave object that a VisualMotion data type is (automatically) mapped. This object allows for simple, indirect access to VisualMotion data types, and it is combined with the subindex to create a direct relationship to the VisualMotion data types. The available objects can be calculated using the formulas in *Accessing Mapped Data* on page *7-125*.

Object SubIndex

The subindex refers to an additional piece of information necessary to obtain direct access to VisualMotion data types. The reference of the subindex depends on the data type in question. For example, the SubIndex refers to the drive number when accessing S and P parameters. However, the subindex refers to the task number when referring to task parameters. The available subindex ranges can be calculated using the formulas in *Accessing Mapped Data* on page *7-125*.

Data Exchange Objects

The four data exchange objects 5E70 to 5E73 represent fixed data "containers" of varying lengths that transfer the VisualMotion ASCII Protocol to the PPC-R card. These objects serve as an open-ended possibility to access any VisualMotion data (including cams, diagnostic text, etc.), but more work is required in the master to perform a transmission of this type. Both the VisualMotion ASCII message and the fieldbus transfer message must be formulated.

Table 7-53 lists the available data exchange objects and their sizes.

Data Exchange Object	Data Length (in bytes)
5E70	16
5E71	32
5E72	64
5E73	128

Table 7-88: Length of the Data Exchange Objects

Handling a Data Exchange Object

When mapped objects are not capable of transferring the desired data, a Data Exchange Object can be used.

The same procedures for writing and reading data apply to the Data Exchange Object.

Selecting a Data Exchange Object

Depending on the length of a VisualMotion ASCII message, any of these data exchange objects can be selected.

Note: The entire data length of the data exchange object must always be transmitted even if the VisualMotion ASCII message is shorter.
For example, if you want to transmit an ASCII message of 42 bytes, you must use object 5E72. To avoid a response error from the fieldbus slave, you must append 22 "Null" characters to the end of the ASCII message to complete a data size of 64 bytes.

Note: The checksum for the VisualMotion ASCII protocol is NOT used with the data exchange object. If the checksum is sent as part of the string, it will be ignored, and no checksum will be sent in the VisualMotion ASCII response messages. To ensure data integrity, the fieldbus protocols support a low-level checksum.

Transmission Sequence via a Data Exchange Object

Note: For the data exchange object, two transmission sequences (and two response sequences) are required, to send the read or write message to and then receive the response message from the PPC-R card.

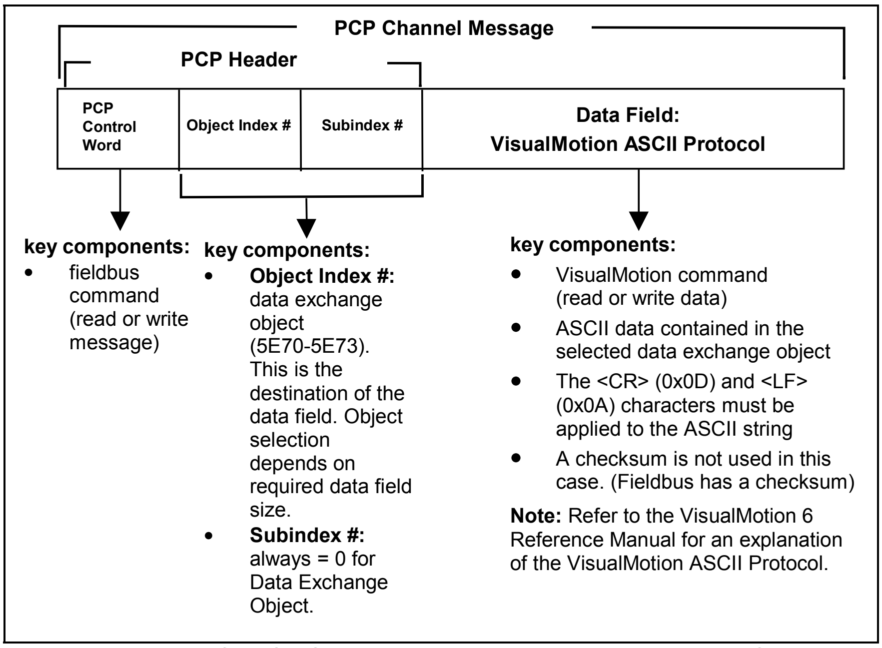


Fig. 7-179: Format of a PCP Channel Message using a Data Exchange Object

For information about reading and writing data using the PCP channel, consult the documentation provided by the fieldbus manufacturer.

Rexroth Bosch Group

Message Steps	Process	Comments
1. Write request from the master with VisualMotion ASCII Protocol.		The data field for the write command can contain a VisualMotion read or write protocol.
2. Acknowledge PCP Channel response from slave.		The response message contains only a confirmation that the fieldbus message was sent properly. Therefore, the size of the data field will be 0.
3. Read request from the master to get VisualMotion ASCII response.		The read request message contains only header information (the data field is 0). You must anticipate the approximate size of the data field (VisualMotion ASCII response message) in order to select the appropriate data exchange object. If the selected object is too short, the data will be truncated.
4. Receive PCP Channel response from slave.		The response message will contain the VisualMotion ASCII response to the VisualMotion ASCII command text in Step 1.

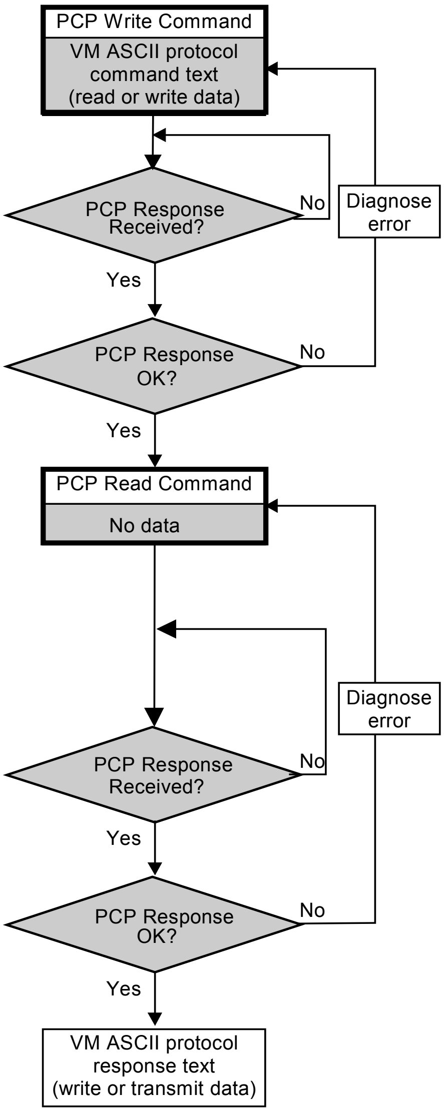


Fig. 7-180: PCP Channel Procedure, Using Data Exchange Object

Accessing Mapped Data (via the PCP Channel)

Rexroth has pre-configured a number of VisualMotion data types to Interbus indexes and subindexes. We call this concept **mapped data**. These data types can be accessed via the Interbus PCP Channel. The index and subindex for each of these data types can be calculated using the formulas in *Table 7-56* below.

	Object Index #	SubIndex #	Formula
	0x5E73	0x00	
Data Exchange Object	----	----	
	0x5E70	0x00	
	0x5E65	0xFF	
<FREE>	----	----	(with SubIndex)
(349 objects available)	0x5D14	0x01	
	0x5D13	0xFF	Index = 0x5D00 + [(Program Integer –1) \ 255]
Program Integers	----	----	
(Int 1 – Int 5100)	0x5D00	0x01	SubIndex = Program Integer – [(Index – 0x5D00) * 255]
	0x5CFF	0xFF	Index = 0x5CEC + [(Program Float –1)] \ 255]
Program Floats	----	----	
(Float 1 – Float 5100)	0x5CEC	0x01	SubIndex = Program Float – [(Index – 0x5CEC) * 255]
	0x5CEB	0xFF	
<FREE>	----	----	(with SubIndex)
(235 objects available)	0x5C01	0x01	
	0x5C00	0xFF	Index = 0x5BF7 + [(Global Integer – 1) \ 255]
Global Integers	----	----	
(GInt 1 – GInt 2550*)	0x5BF7	0x01	SubIndex = Global Integer – [(Index – 0x5BF7) * 255]
	0x5BF6	0xFF	Index = 0x5BED + [(Global Float – 1) \ 255]
Global Floats	----	----	
(GFloat 1 – Gfloat 2550*)	0x5BED	0x01	SubIndex = Global Float – [(Index – 0x5BED) * 255]
	0x5BEC	0xFF	
<FREE>	----	----	(with SubIndex)
(245 objects available)	0x5AF8	0x01	
	0x5AF7	0xFF	Index = 0x5AEE + [(Register – 1) \ 255]
Registers	----	----	
(Reg. 1 – Reg. 2550**)	0x5AEE	0x01	SubIndex = Register – [(Index – 0x5AEE) * 255]
	0x5AED	0x04	Index = 0x56F1 + T-Parameter
T-Parameters	----	----	
(T-0-0001 – T-0-1020)	0x56F1	0x01	SubIndex = Task Number
	0x56F0	0xFF	
<FREE>	----	----	(with SubIndex)
(241 objects available)	0x5600	0x01	
	0x55FF	0x01	Index = 0x4800 + C-Parameter
C-Parameters	----	----	
(C-0-0001 - C-0-3583)	0x4801	0x01	SubIndex = 1
	0x47FF	0x63	Index = 0x4000 + A-Parameter
A-Parameters	----	----	
(A-0-0001 - A-0-2047)	0x4001	0x01	SubIndex = Axis Number
	0x3FFF	0x63	Index = 0x3000 + P-Parameter
P-Parameters	----	----	
(P-0-0001 - P-0-4095)	0x3001	0x01	SubIndex = Drive Number

	Object Index #	SubIndex #	Formula
S-Parameters	0x2FFF	0x63	Index = 0x2000 + S-Parameter
	----	----	
(S-0-0001 - S-0-4095)	0x2001	0x01	SubIndex = Drive Number
<Reserved>	0x1FFF	----	
	----	----	
	0x0000	----	

* current limitation: C-0-0080/C-0-0081 - Maximum number global integers/floats..

**current limitation: first 1024 registers.

Table 7-89: Formulas for Determining Mapped Objects

Example Lookup Tables for Mapped Objects

Card (C) Parameters

The following is an example lookup table for C-Parameters, when using mapped objects.

Example Look-up Chart for: **C-Parameters** CP 0.Y ==> CP = Card Parameter
Y = Parameter Number

Index

	0x4801	0x4802	0x4803	……………	0x48FF	0x4900	……………	0x55FE	0x55FF
SubIndex = 0x01	CP 0.1	CP 0.2	CP 0.3		CP 0.255	CP 0.256		CP 0.3582	CP 0.3583

Table 7-90: Mapped Object Lookup Table for C-Parameters

Axis(A) Parameters

The following is an example lookup table for A-Parameters, when using mapped objects. The same formula also applies to Sercos (S) and Task (T) Parameters.

Example Look-up Chart for: **A-Parameters** AP X.Y ==> AP = Axis Parameter
X = Axis Number
Y = Parameter Number

Index

	0x4001	0x4002	0x4003	………… …	0x40FF	0x4100	……………	0x47FE	0x47FF
SubIndex = 0x01	AP 1.1	AP 1.2	AP 1.3		AP 1.255	AP 1.256		AP 1.2046	AP 1.2047
0x02	AP 2.1	AP 2.2	AP 2.3		AP 2.255	AP 2.256		AP 2.2046	AP 2.2047
0x03	AP 3.1	AP 3.2	AP 3.3		AP 3.255	AP 3.256		AP 3.2046	AP 3.2047
:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:
0x28	AP 40.1	AP 40.2	AP 40.3		AP 40.255	AP 40.256		AP 40.2046	AP 40.2047

Table 7-91: Mapped Object Lookup Table for A-Parameters

Product-Specific (P) Parameters

The following is an example lookup table for P-Parameters, when using mapped objects.

Example Look-up Chart for: **P-Parameters** PP X.Y ==> PP = SERCOS P-Parameter (set 0 only)
X = Drive Number
Y = Parameter Number

Index = (Class ID && Instance ID for DeviceNet)

SubIndex = (Attribute ID for DNet)

	C118, In1	C118, In2	C118, In3	....	C118, In255	C119, In1	....	C134, In14	C134, In15
	0x3001	0x3002	0x3003	....	0x30FF	0x3100	....	0x3FFE	0x3FFF
0x01	PP 1.1	PP 1.2	PP 1.3		PP 1.255	PP 1.256		PP 1.4094	PP 1.4095
0x02	PP 2.1	PP 2.2	PP 2.3		PP 2.255	PP 2.256		PP 2.4094	PP 2.4095
0x03	PP 3.1	PP 3.2	PP 3.3		PP 3.255	PP 3.256		PP 3.4094	PP 3.4095
:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:
0x28	PP 40.1	PP 40.2	PP 40.3		PP 40.255	PP 40.256		PP 40.4094	PP 40.4095

Table 7-92: Mapped Object Lookup Table for P-Parameters

Integers

The following is an example lookup table for Integers, when using mapped objects. The same formula also applies to Floats, Global Integers, Global Floats and Registers.

Example Look-up Chart for: **VM Program Integers** PI 0.Y ==> PI = Program Integer
Y = Program Integer Number

Index

SubIndex =

	0x5D00	0x5D01	0x5D02	....	0x5D13
0x01	PI 1	PI 256	PI 511		PI 4846
0x02	PI 2	PI 257	PI 512		PI 4847
0x03	PI 3	PI 258	PI 513		PI 4848
:	:	:	:	:	:
:	:	:	:	:	:
0xFF	PI 255	PI 510	PI 765		PI 5100

Table 7-93: Mapped Object Lookup Table for Integers

8 Data Editing and Monitoring Tools

8.1 Parameter Overview

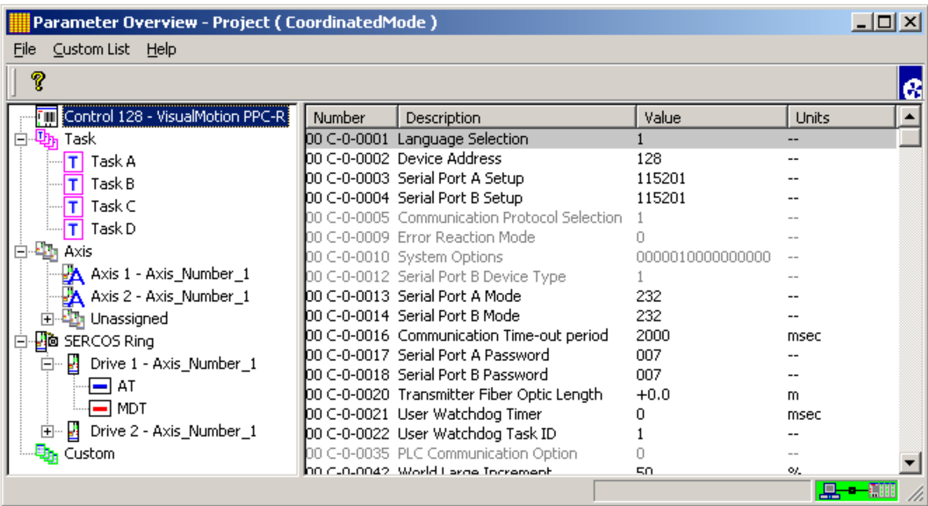
The Parameter Overview tool is used to view, modify, and compare existing Control, Task, Axis and Sercos device parameters. The user can also create and edit a Custom List containing user-selected parameter. The following parameter types are displayed in an expandable tree structure, similar to the folder (directory) structure found in Windows™:

Parameter Type	Description
Control	All control specific parameters are displayed when Control is selected.
Task	All parameters for the selected task are displayed when Task (A, B, C or D) is selected.
Axis	All parameters for the selected axis are displayed when Axis # (up to 64) is selected.
Sercos	All parameters for the selected Sercos digital drive, up to a maximum of 64, and Sercos I/O devices are listed.
Custom	Any existing custom list created by the user.

Table 8-1: Parameter Types

Parameter Overview Edit Mode

The Parameter Overview tool is launched by selecting ***Data ⇒ Parameters ⇒ Edit***.



Data_Param.tif

Fig. 8-1: Parameter Overview Window

Editing Parameters in Offline Mode

When the *Parameter Overview* tool is launched in offline mode, the parameters are read directly from the project folder and not from the control's memory. All parameters that are not read-only can be modified and saved to the project.

Note: When editing parameter values in offline mode, the system does not validate the entered value with the parameter's valid range until the project is switched to online mode.

Editing Parameters in Online Mode

When the *Parameter Overview* tool is launched in online mode, the parameters are read directly from the control's memory. Some parameters can be edited in Sercos phase 4 while others are only editable in Sercos phase 2. The system will notify the user of the valid Sercos phase for editing the selected parameter. All parameters that are not read-only can be modified and saved to the project and control.

Parameter Access

Parameter "read" or "write" access is identified by the text color displayed in the parameter overview window. The following table indicates the color code / access combination.

Color Code	Description	Access
grayed out text	read-only parameter or not editable in current phase	read-only
black text	parameter that can be edited	read/write
red text	used to indicate an error	read/write
blue text parameter list	parameter list, denoted by Xs, can be edited	read/write
grayed out parameter list	read-only parameter list, denoted by Xs, or not editable in current phase	read-only

Table 8-2: Parameter Access

Configuring Sercos Drive AT and MDT

The *Parameter Overview* tool contains a Sercos Telegram Tool used to configure each drive's Sercos Drive Telegram (AT) and Master Data Telegram (MDT). The Sercos Telegram Tool is only available in online mode. VisualMotion's Telegram Tool provides the user with a convenient and comprehensive interface for viewing the AT and MDT and modifying selected portions of each telegram.

The AT and MDT are used to cyclically exchange data between drives and control every Sercos cycle. The AT is sent from each drive to the control and the MDT is sent from the control to each drive on the system. The AT and MDT are comprised of various parameters stored in each drive. Some parameters displayed in the AT and MDT are automatically configured based on the system's primary and secondary modes of operation. These parameters appear as gray text. User configurable areas are displayed in black text.

Refer to section 7.3*, Sercos Drive Telegram Utility,* for details.

Edit a Parameter

To edit a parameter, select the parameter type (Control, Task, etc.), and then double click on the desired parameter number from the parameter overview window.

Parameter Help System not Found

VisualMotion issues an error message when help is requested for a help system that is not installed on your PC. Some reasons why this may occur are as follows:

- The help files were moved to a different folder.
- The help files were not installed or deleted from the computer.
- The system language was changed and the help files do not exist for the selected language.

Fig. 8-2: Help System Error Message

Note: This error is typically encountered when help for Sercos drive parameter is requested. Each drive firmware has its own help system that can be installed. Drive help is available on the following DriveHelp CD with material number 282411:

DOK-GENERL-DRIVEHELP**-GN**-MS-D0600

When the **Yes** button is pressed, the following window will assist the user in locating a suitable help system.

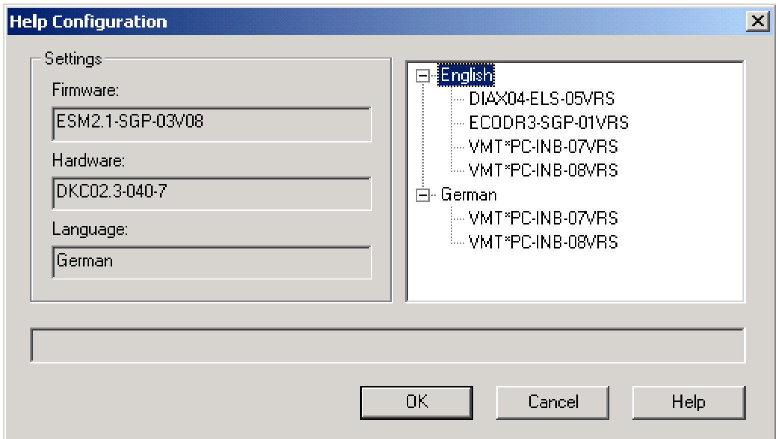
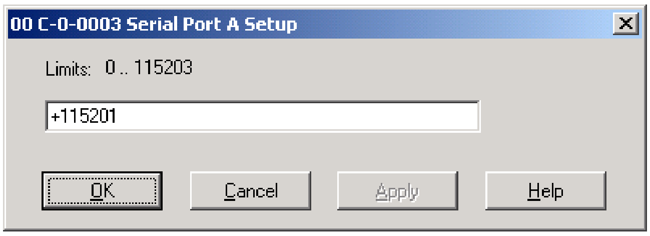


Fig. 8-3: Help System Location Window

Edit a Standard Parameter

A standard parameter can be an Integer, Float, String or Hex value. The standard parameter edit window in the figure below is displayed when editing a standard parameter. The current data **Limits** for a parameter are displayed above the input field, from minimum to maximum values.

Note: When editing parameter values in offline mode, the system cannot check the entered value with the parameter's valid range until the project is switched to online mode.



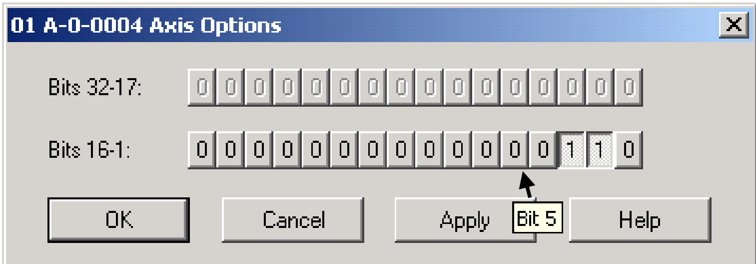
On_Line_Edit_Stnd_Param.tif

Fig. 8-4: Parameter Edit Window

Edit a Binary Parameter

The binary parameter edit window in the figure below is modified by clicking on the desired bit(s). Holding the mouse cursor over a bit will display a tool tip containing the bit number.

Note: 16 bit parameters will only have the first 16 bits accessible. Bits 17-32 are grayed out.

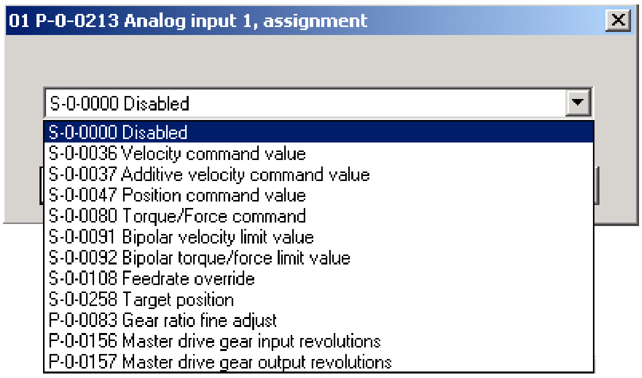


On_Line_Edit_Binary.tif

Fig. 8-5: Binary Parameter Edit Window

Edit a Parameter from a Predefined List

This parameter edit window uses a drop down list to selected parameters that have been predefined as valid selections. To edit this parameter, the user selects the desired parameter from the drop down list.

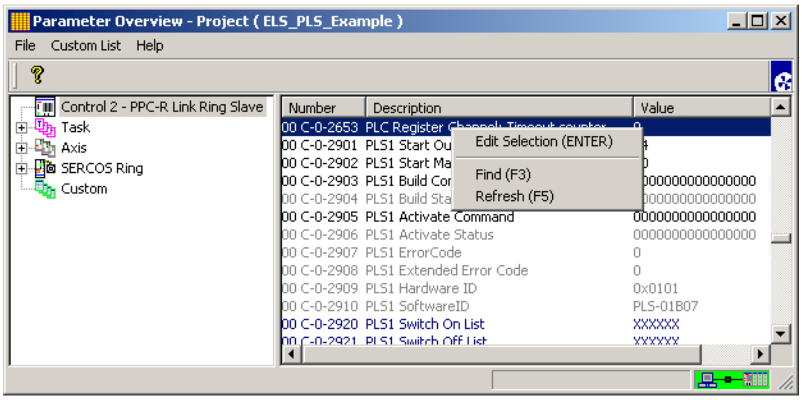


On_Line_Param_Predfnd.tif

Fig. 8-6: Predefined Parameter List Edit Window

Edit, Refresh or Find a Parameter or List

Right clicking on a selected parameter opens a popup window where the user can ***Edit Selection (ENTER)***. Selecting the ***Refresh (F5)*** option updates all the parameters visibly displayed in the main window. The ***Find (F3)*** option allows the user to locate a parameter by using a partial description.



On_Line_Param_Options.tif

Fig. 8-7: Parameter Options

Note: Parameters can also be edited by double clicking the left mouse button.

Display a Parameter List

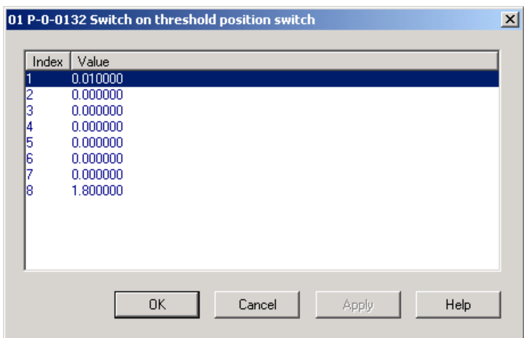
The information contained in Parameter lists can be displayed in one of the two following formats:

- Data Format – list of data
- IDN Format – list of parameter numbers

Parameter lists are displayed in either blue text (read/write access) or gray text (read only) with the value column displaying six Xs. To display a parameter list, double click on the desired parameter and VisualMotion will open a new window. The new window will display the parameter number and description in the window's header and display the contents in either Data format or IDN format.

Data Format

Parameter lists displayed in data format will contain an Index and a Value column as shown in the figure below.



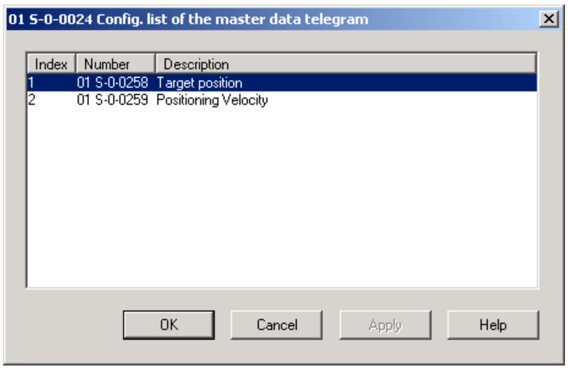
On_Line_Param_Data.tif

Fig. 8-8: Parameter List Data Format

IDN Format

Parameter lists displayed in IDN (IDentification Number) format is a list of other parameters containing an index, parameter number and description as shown in the figure below.

Rexroth
Bosch Group

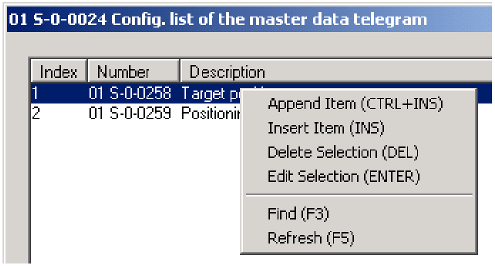


On_Line_Param_IDN.tif

Fig. 8-9: Parameter List IDN Format

Append, Insert and Delete within a Parameter List

Once a parameter list is opened, right clicking anywhere in the window opens a popup window where the user can perform the following functions.



On_Line_Param_IDN_Apnd.tif

Fig. 8-10: Parameter List Options

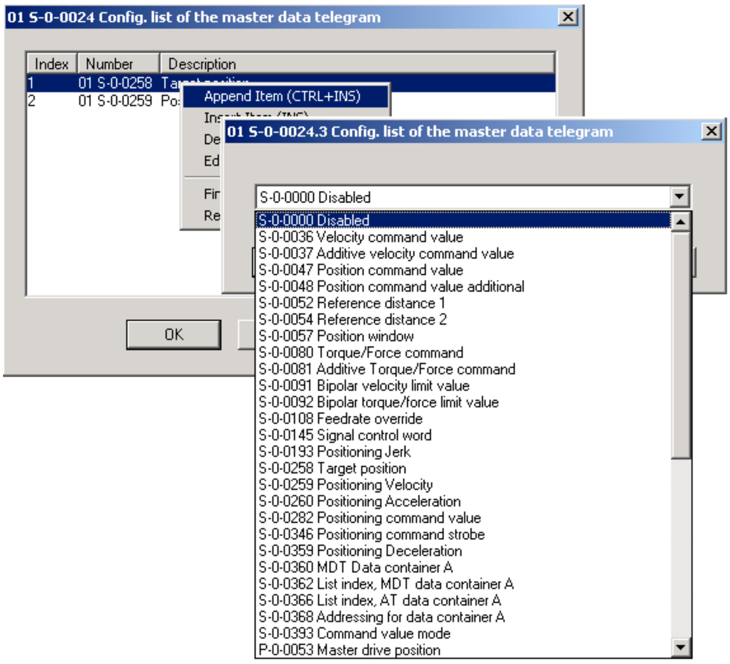
Note: Not all parameter lists allow the addition and removal of parameters.

- **Append Item (CTRL+INS)** – adds a new item to the end of the current list.
- **Insert Item (INS)** – inserts an item above the selected value or parameter.
- **Delete Selection (DEL)** – deletes the selected data or parameter from the list.
- **Edit Selection (ENTER)** – opens an edit window where the data or parameter value can be edited.
- **Refresh (F5)** – refreshes all values displayed in the parameter list.
- **Find (F3)** – locates a parameter using the number or description.

Adding Predefined Parameters to a List

The Append and Insert functions within a list can be used to add predefined parameters to certain lists. To add a parameter from a list, right click in the window and select Append or Insert. A parameter selection edit window will open. Next, select the desired parameter from the drop down list. Repeat the process as necessary.

Note: Only certain drive parameters contain a listing of valid parameters. In other cases, the data is manually entered.



On_Line_Param_Edit.tif

Fig. 8-11: Parameter Selection Edit Window

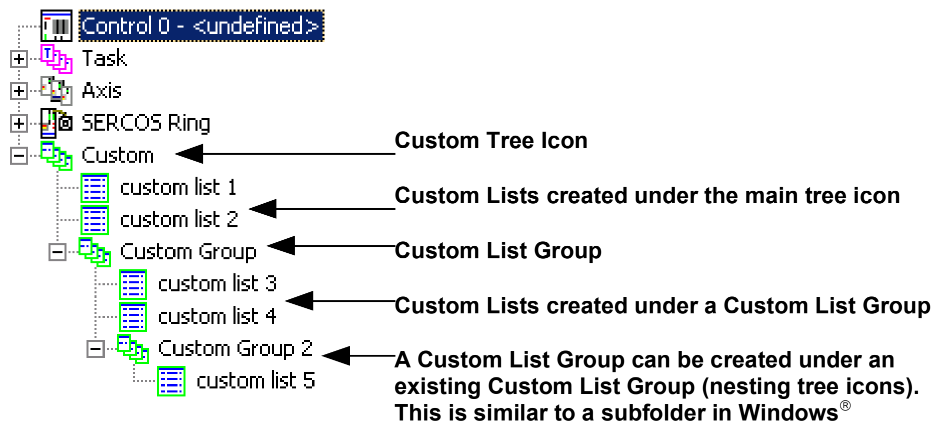
Custom

The **Custom** tree icon provides the user with easy to create and manage parameter lists that are specific to their application. From this tree icon selection, the user can create, modify and delete custom lists and custom list groups.

A *Custom List* is a grouping of parameters that are user selected.

A *Custom List Group* is a tree icon that is created under the **Custom** tree icon and used to group multiple custom lists together.

Note: All custom groups and custom list are stored under the ***\Rexroth\VisualMotion11\param*** directory. Custom list are saved with a "*.custom" file extension and custom list groups appear as subfolders under ***param***.



On_Line_Param_Custom_List.tif

Fig. 8-12: Custom List File Structure

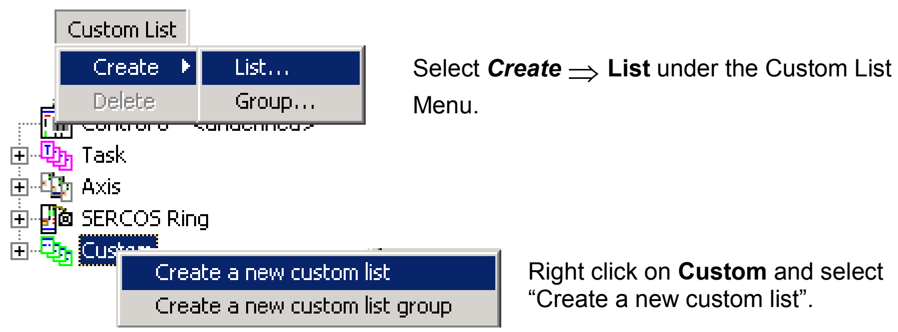
Create a Custom List

Any combination of parameters from the four types (Control, Task, Axis or Sercos Ring) can be added to a custom list. By creating a custom list, the user minimizes the number of displayed parameters, making navigating and searching for a parameter easier.

To create a new custom list, use the following steps:

1. Select ***Custom List*** ⇒ ***Create*** ⇒ ***List*** from the main menu or right click on the **Custom** tree icon and select "Create a new custom list".

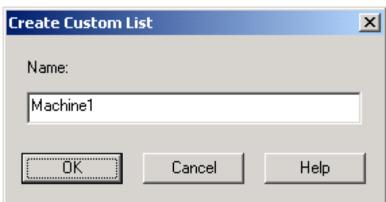
Note: In both cases, the **Custom** tree icon must first be highlighted before the selections are made available.



On_Line_Param_Create_List.tif

Fig. 8-13: Custom List Menu

2. Enter a name in the *Create Custom List* window, up to a maximum of 80 characters, that identifies the list and press the **OK** button.



Create_Custom_List.tif

Fig. 8-14: Create Custom List

3. The following sequence is how parameters are added to a custom list within the *Custom List Editor* window.
   a) Select one of the following parameter sets from the drop down list:
      - Control Parameter Set
      - Task Parameter Set
      - Axis Parameter Set
      - Sercos Parameter Set (Drive or I/O)

   b) If applicable, select a task (A, B, C or D) for Task Parameter Set and an address number for Axis or Sercos Parameter Sets.
   c) Parameters are added to a list by double clicking on the desired parameter or highlighting the parameter from the left window and pressing on the >> insert button. Repeat the process for the another parameter set until all the desired parameters are added to the list.
   d) Press the **OK** button to save the custom list.

Note: To remove a parameter from the custom list, double click on the parameter in the right window or highlight it and press the << remove button.

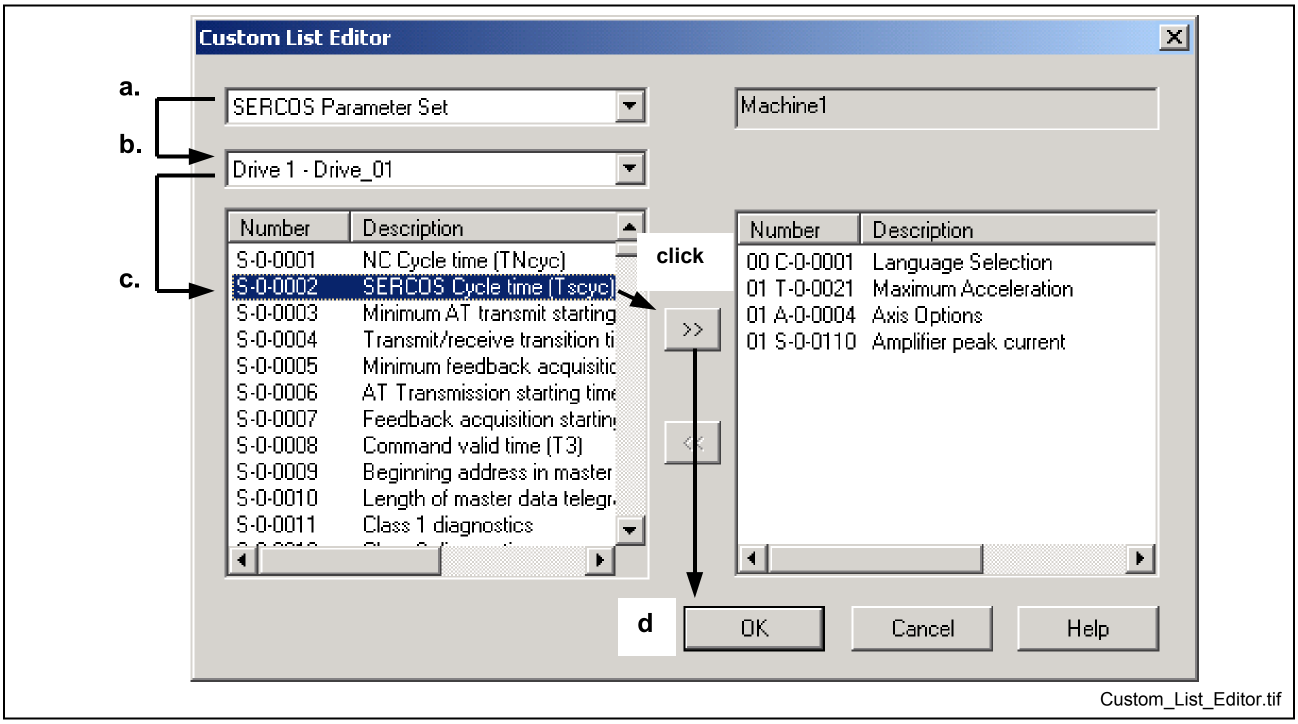
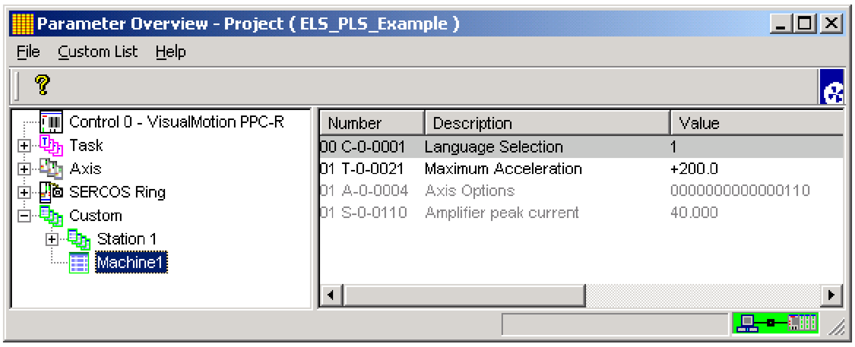


Fig. 8-15: Custom List Editor

The newly created custom list name will appear below the **Custom** tree selection, as shown in the figure below.



Custom_Tree.tif

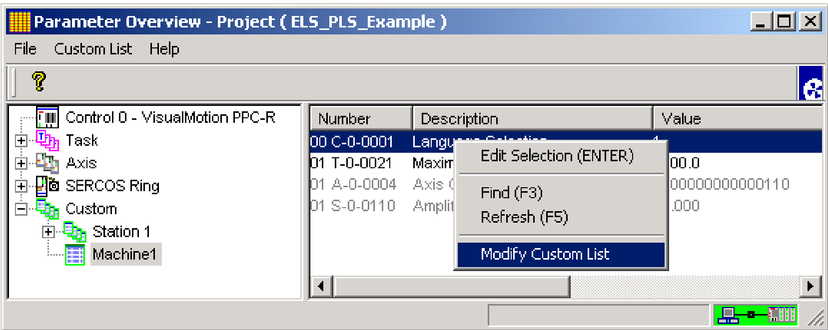
Fig. 8-16: Custom Tree Selection

Note: To delete a custom list, right click on the name and select "Delete the selected custom list item" from the popup window.

Modify a Custom List

To modify a custom list, use the following steps:

1. Select the custom list from the tree structure. The contents of the custom list are displayed in the right window.



Custom_List_Modify.tif

Fig. 8-17: Modify Custom List

2. Right click anywhere in the right window and select "Modify Custom List".

The addition and removal of parameters is described in section, "Create a Custom List".

Create a Custom List Group

A custom list group is a tree icon created under **Custom** where multiple custom lists can be grouped. This option allows the user to group or categorize different custom lists to a particular project or machine. Multiple custom list groups can be created following the previous group (nesting).

To create a custom list group, use the following steps:

1. Select ***Custom List*** ⇒ ***Create*** ⇒ ***Group*** from the main menu or right click on the **Custom** tree icon and select "Create a new custom list group".

Note: In both cases, the **Custom** tree icon must first be highlighted before the selections are made available.

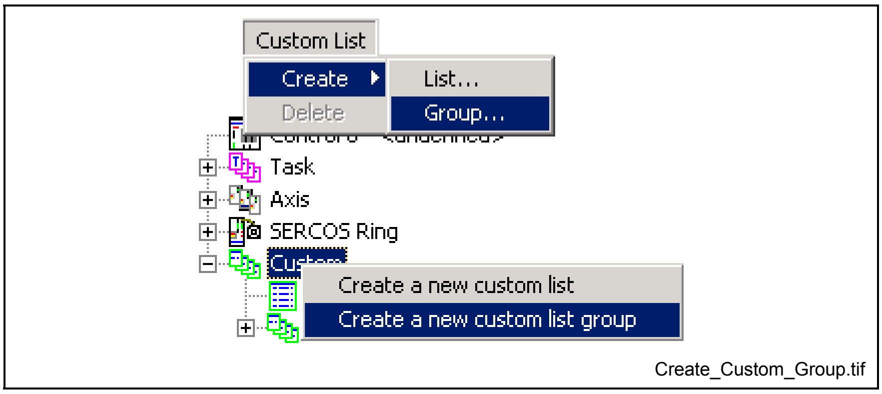


Fig. 8-18: Create Custom Group

2. Enter a name in the *Create Custom List Group* window for the custom, up to a maximum of 20 characters, that identifies the group and press the **OK** button.

The newly created custom list group name will appear below the **Custom** tree icon, as shown in the figure below.

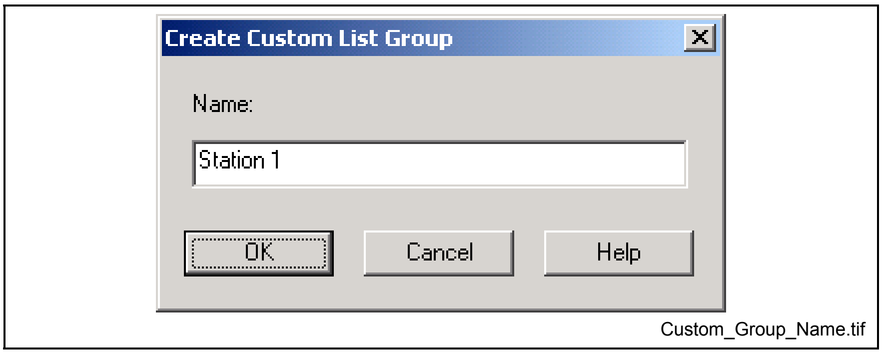


Fig. 8-19: Name Custom Group

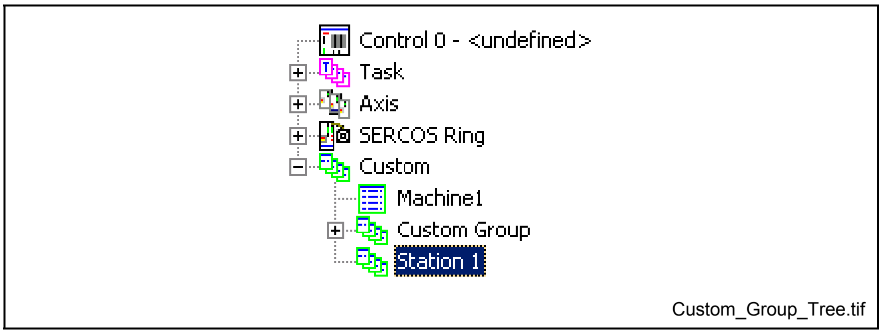


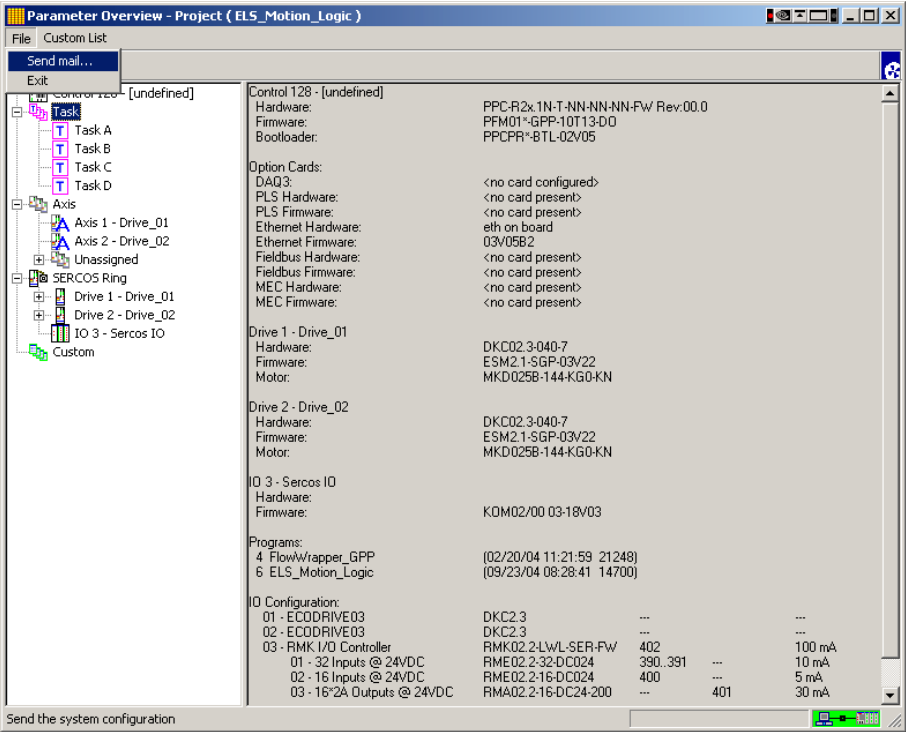
Fig. 8-20: Custom Group Tree

To add a custom list under the group name, select the group and follow the steps in section *Create a Custom List*.

Note: To delete a custom list group, right click on the group name and select "Delete the selected custom list item". This option is not available if a custom list exists under the group name.

System Configuration

The system configuration overview for the connected system components can be viewed from the Parameter Overview tool by selecting the top-level selection for Task, Axis or Sercos Ring.



Param_System_Config.tif

Fig. 8-21: System Configuration

This system configuration can be e-mailed to a recipient by selecting ***File*** ⇒ ***Send mail***. This option launches the e-mail system on the PC and attaches a text file containing the system configuration as displayed in the figure above.

Note: The *Send mail…* option will be visible only if the PC has a configured e-mail client.

Compare Parameters

Selecting ***Data*** ⇒ ***Parameters*** ⇒ ***Compare…*** launches the *Compare Parameters* tool. This tool is supported in all programming modes (offline, online and service). It allows the user to compare system parameter sets (C, T, A, S, and P), of the same type, saved in the project and on the control's memory with each other or with the system default values. Parameter values can also be edited, copied or deleted, if allowed. The modification to parameter values are not valid until saved.

The parameter selections and tabs that appear under them vary based on the current programming mode. The following table lists the four tab selections, functions and their relevant programming mode:

Tab Selection	Function	Programming Mode
Components	Compares functionality based parameter sets on the Control and the Default values against the Project.	Offline, Online, and Service
Parameters	Compares individual Control, Task, Axis, and Drive parameter sets on the Control and the Default values against the Project.	Offline, Online, and Service
Project	Compares one or more Task, Axis, or Drive parameter sets stored in the project against the system Default values.	Offline and Online
Control	Compares one or more Task, Axis, or Drive parameter sets stored on the control against the system Default values.	Online and Service

Table 8-3: Compare Parameters Selections

Note: Drive parameter sets and the Control column are only supported in online and service mode.

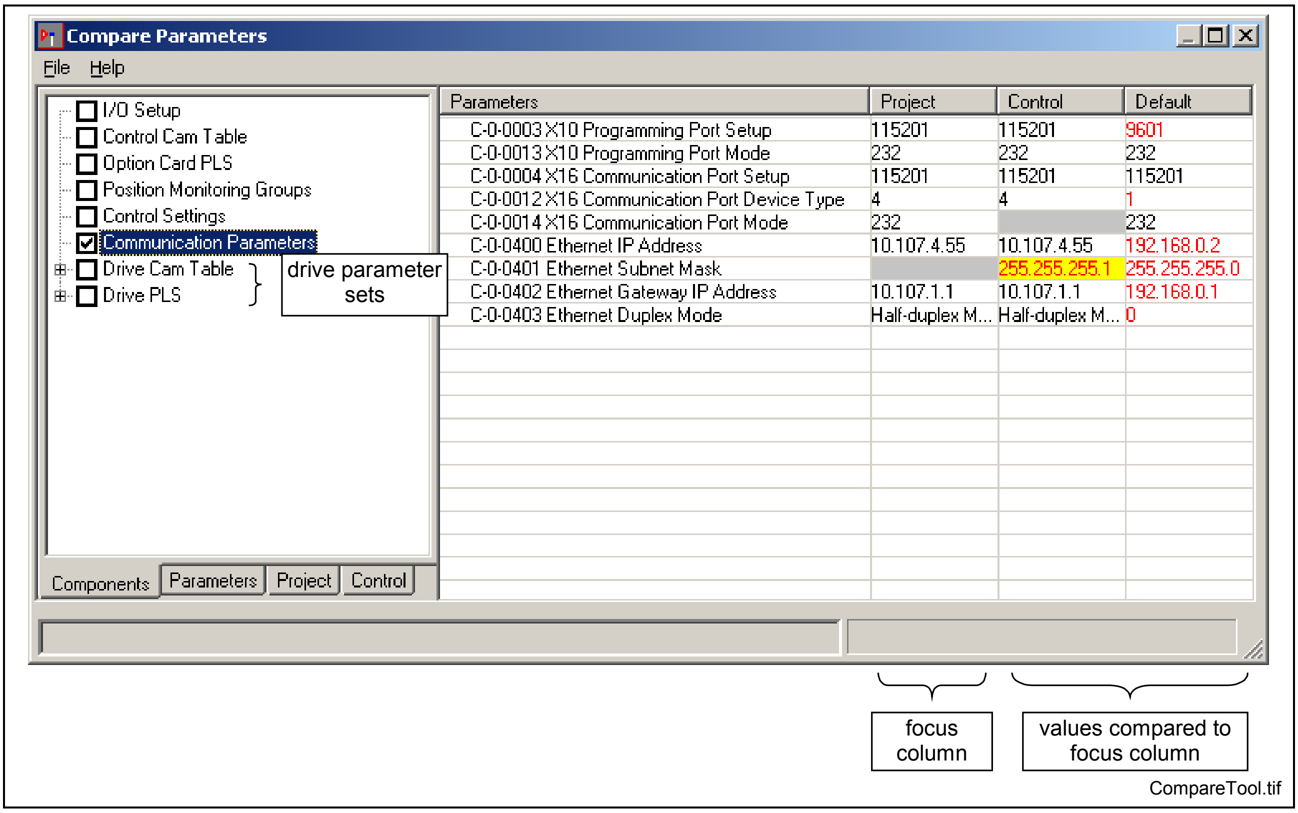


Fig. 8-22: Compare Parameters Tool in Online Mode

Comparison Focus

When comparing parameter sets, the focus of the comparison is on the first column of parameter values. All columns to the right are being compared to the first column of values. For example, Fig. 8-22 is comparing the values on the Control and Default to those in the Project.

Display Indications

The following table lists the visual display indications used in the Compare Parameter tool.

Indicator	Meaning
black text / white background	no change in value as compared to the focus column
no text / gray background	parameter is not stored as part of the parameter set
red text	compared value is different from the value in the focus column
yellow background	value was modified but not saved
light gray text	read-only parameter / cannot be modified
blue text	list parameter

Table 8-4: Display Indications

Menu and Right Click Options

Selecting ***File ⇒ Options…*** opens the window in Fig. 8-23. The following table describes the available menu options:

Option	Description
Hide read-only parameters	When checked, read-only parameters (light gray text) are not displayed.
Hide parameters not in the project	When checked, all parameters not stored in the project (gray background) are not displayed.
Hide parameter which are equal	When checked, all parameters with equal values in all columns are not displayed.

Table 8-5: Compare Parameter Options

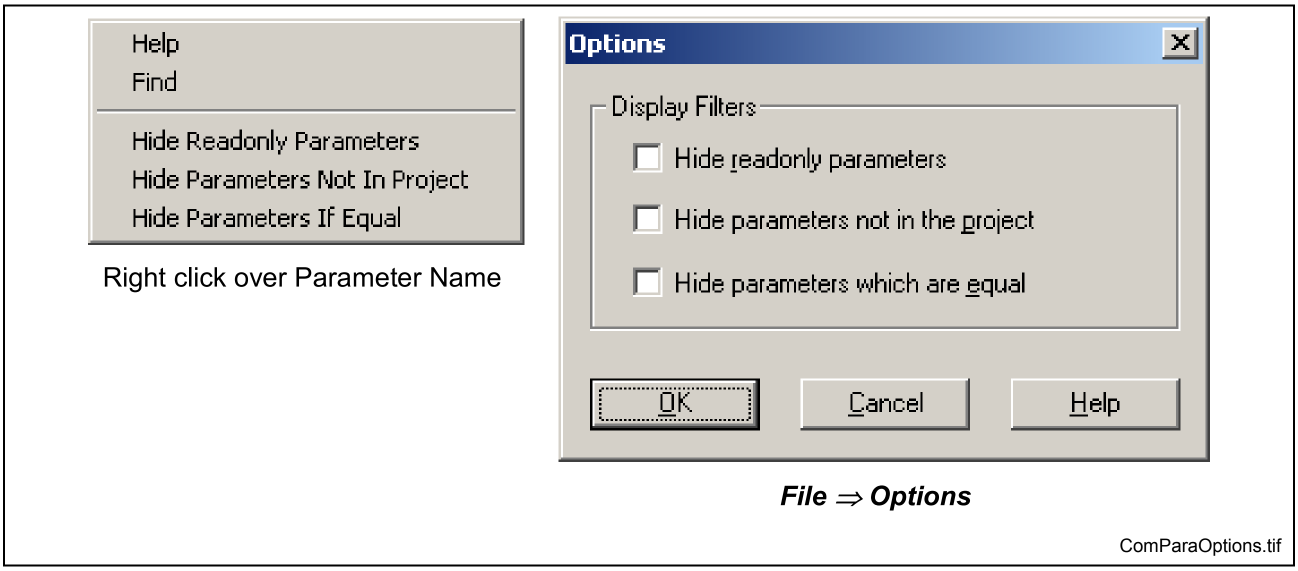
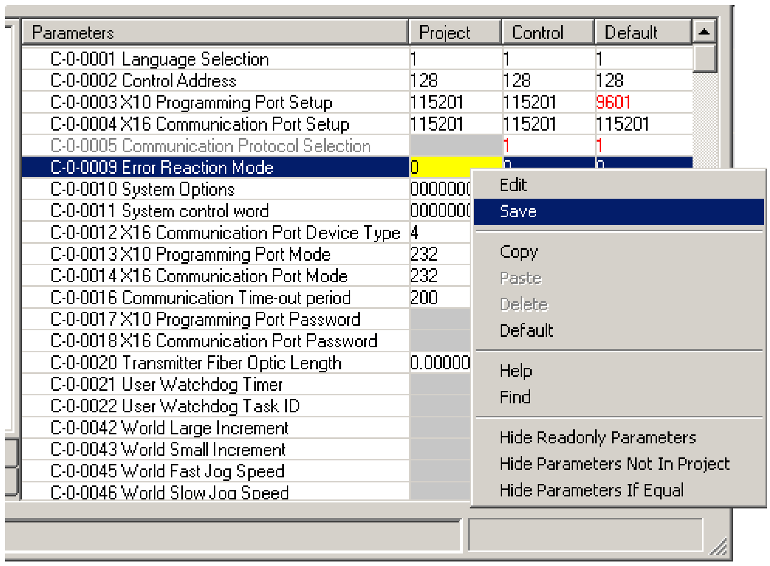


Fig. 8-23: Compare Parameter Options

Edit Parameter Values

Parameter values can be edited by double clicking over an existing value or gray background and entering the new value. Once the new value is entered, the background color changes to yellow, indicating that the value was changed. If the value entered is different from the focus column, the new value will be displayed in red. Once the parameter set is saved (either by selecting *File ⇒ Save*, right clicking over the new value and selecting Save, or changing to a different tab) the new value will appear as black text over a white background.

Note: Default values cannot be changed from their original value.



ComParaEdit.tif

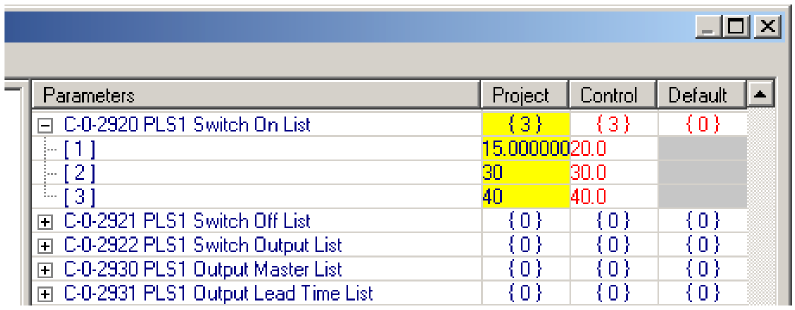
Fig. 8-24: Edit a Parameter Value

The following table describes the available option when right clicking over a parameter value:

Option	Description
Edit	Selects a writable parameter value for input
Save	Right clicking over an individual value saves only that modified value while others remain with a yellow background. Right clicking over the column heading saves all modified value for the entire set. When selecting *File ⇒ Save*, all modified parameters values, regardless of column, are saved.
Copy	Copies the contents to the Windows clipboard.
Paste	Copies the valid contents of the Windows clipboard to a new location, if value type is allowed.
Delete	Deletes the parameter value and replaces it with a gray background. Not supported for Control or Default values.
Default	Replaces the current value for a project or control with the Default value.
Help	Opens content sensitive help for the current selected parameter.
Find	Searches parameter names for a match.

Table 8-6: Compare Parameter Value Options

List parameter values, displayed as blue text, can only be expanded and modified if they contain elements. An list parameter with zero elements is display as { 0 }. Double clicking on the list parameter name, expands the list parameter into its individual element values.



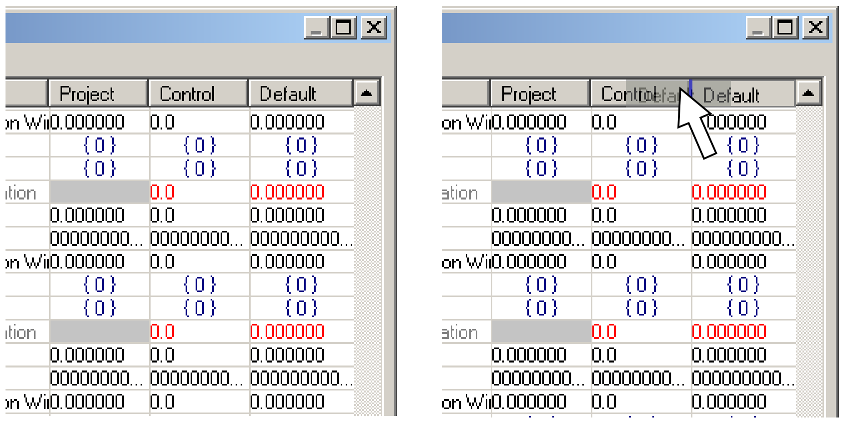
ComParaList.tif

Fig. 8-25: List Parameter Values

Copy Parameter Values

Parameter values can be individually copied by double clicking on an existing value and selecting the contents (using standard Windows right click or keyboard shortcuts). Next, double click on an adjacent parameter and paste the new value.

To copy an entire parameter set (for example, Task A to Task B) to a different parameter set of similar type, click, hold, and drag the column heading of the first set to a different column heading.



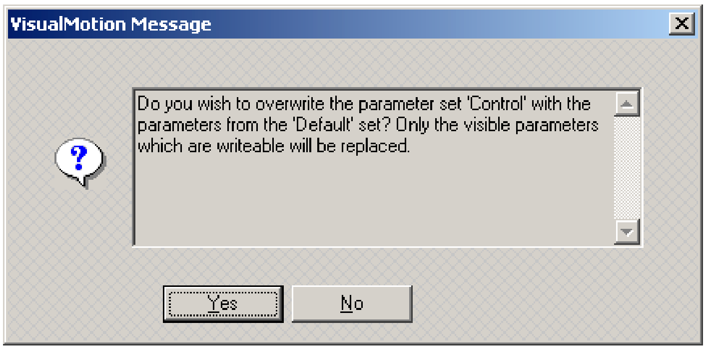
Columns before Copying

Columns during Copying
Notice the gray shadow between the Default and Control Columns

ComParaCopyColumn.tif

Fig. 8-26: Copy a Column to a Different Column

After the copy, the following VisualMotion message warning requires the user to accept the copy command.



CopyMessage.tif

Fig. 8-27: Copy Warning Message

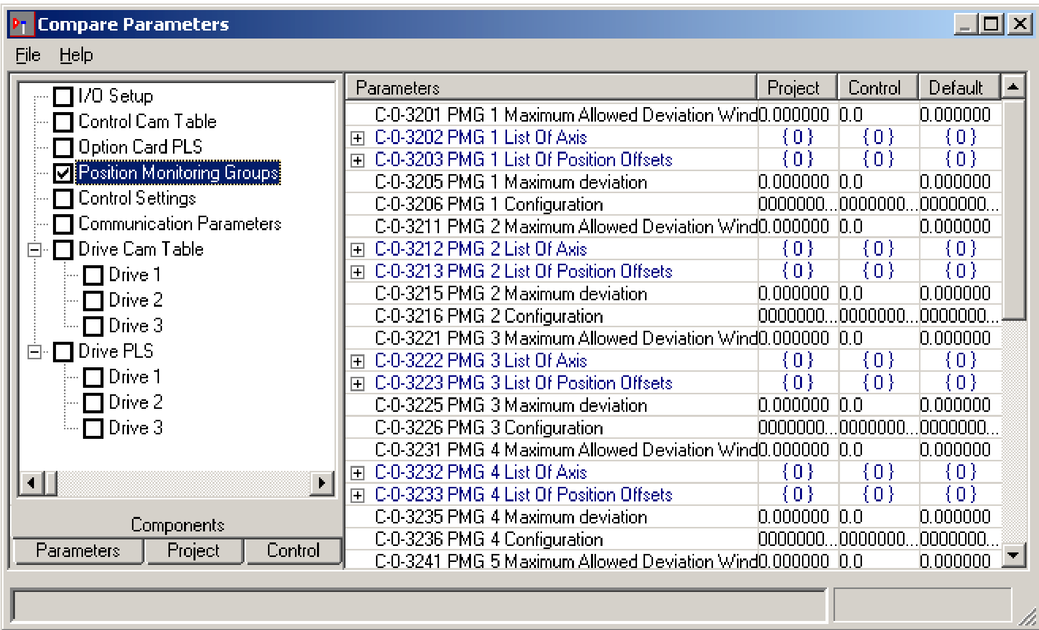
Compare Components

The Components tab displays a collection of control and drive parameter sets grouped by functionality. Drive specific parameter sets such as, Drive CAM Table and Drive PLS are only accessible in online and service modes.

The following table lists the available columns that can be compared based on the current programming mode of VisualMotion Toolkit:

Programming Mode	Project	Control	Default
Offline Mode	X		X
Online Mode	X	X	X
Service Mode		X	X

Table 8-7: Compare Component Columns



ComComp.tif

Fig. 8-28: Compare Components in Online Mode

Rexroth Bosch Group

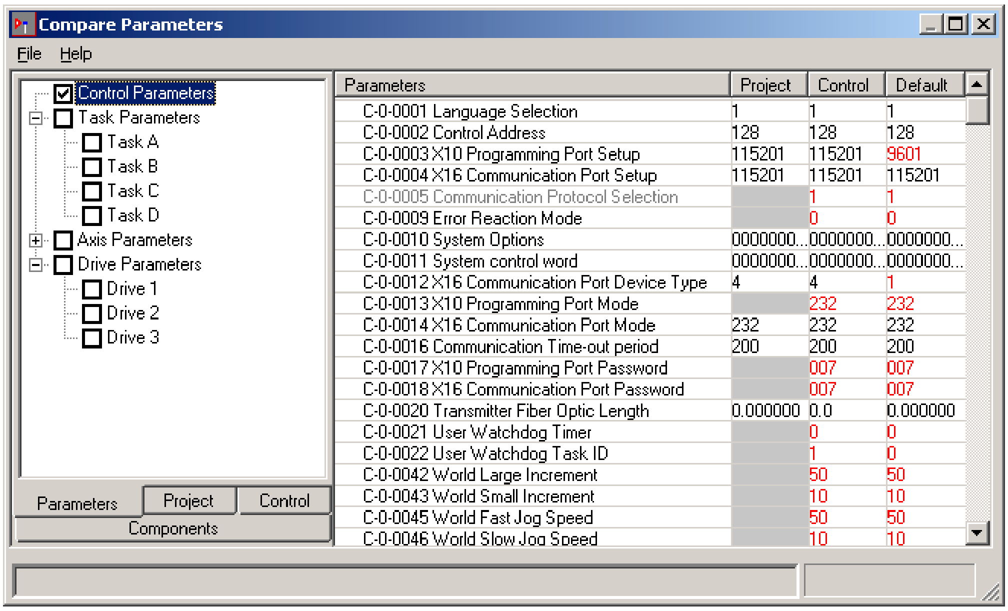
Compare Parameters

The Parameters tab displays the complete set of Control, Task, Axis, and Drive parameters. Drive parameters (S and P) are only accessible in online and service mode.

The following table lists the available columns that can be compared based on the current programming mode of VisualMotion Toolkit:

Programming Mode	Project	Control	Default
Offline Mode	X		X
Online Mode	X	X	X
Service Mode		X	X

Table 8-8: Compare Parameter Columns



ComParam.tif

Fig. 8-29: Compare Parameters in Online Mode

Compare Project

The Project tab allows a user to compare multiple sets of similar parameter sets, stored in the project, for Task, Axis, and Drive among each other or against the Default values. Since only one set of control parameters exists in a system, they are not accessible under the Project tab. To compare control parameters against the default values, select the Parameters tab. Drive parameters (S and P) are only accessible in online and service mode.

Note: The Project tab is only supported in offline and online modes. In service mode, there is no connection to the project.

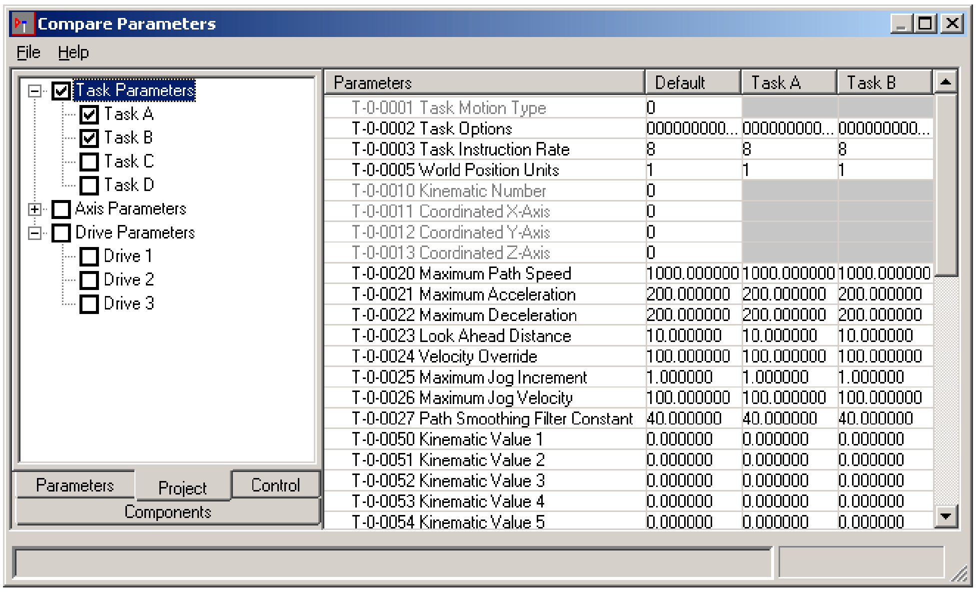


Fig. 8-30: Compare Project in Online Mode

Compare Control

The Control tab allows a user to compare multiple sets of similar parameter sets, stored in the control's memory, for Task, Axis, and Drive among each other or against the Default values. Since only one set of control parameters exists in a system, they are not accessible under the Control tab. To compare control parameters against the default values, select the Parameters tab. Drive parameters (S and P) are only accessible in online and service mode.

Note: The Control tab is only supported in online and service modes. In offline mode, there is no connection to the control.

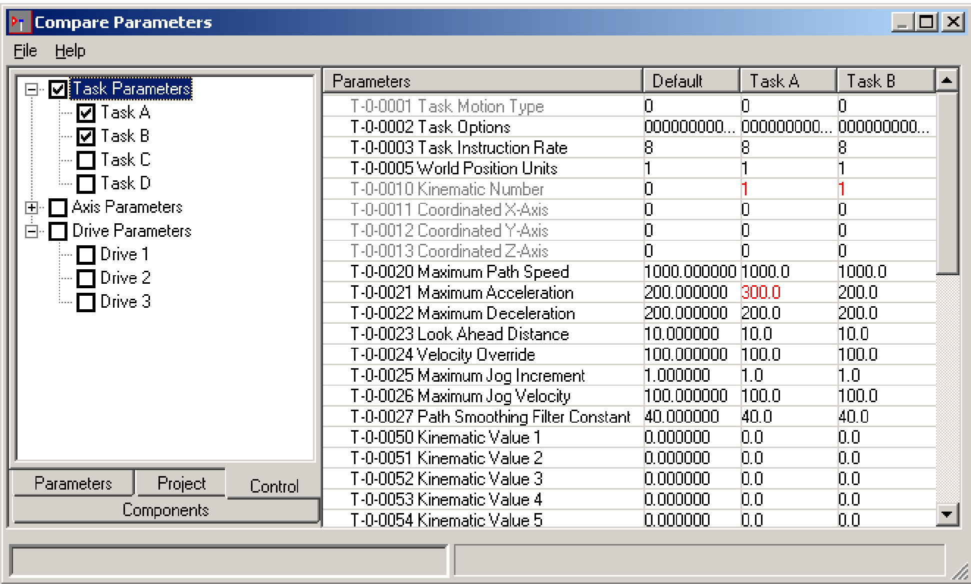


Fig. 8-31: Compare Control in Online Mode

8.2 Data Editor

The Data Editor is used to display and edit registers, program variables, global variables, event tables, points, zones, and custom lists. From the data type drop-down list, the user can select the desired data type. In addition, a user defined custom list can be created containing a combination of the mentioned data types. The available display formats are binary, hexadecimal, and decimal.

Registers

VisualMotion supports 1024 (16-bit) registers. Registers can be modified as a complete binary register or by their individual bits.

Sorting Data

Any of the three data type columns (Number, Label and Value) can be sorted in ascending or descending order by clicking on the column heading.

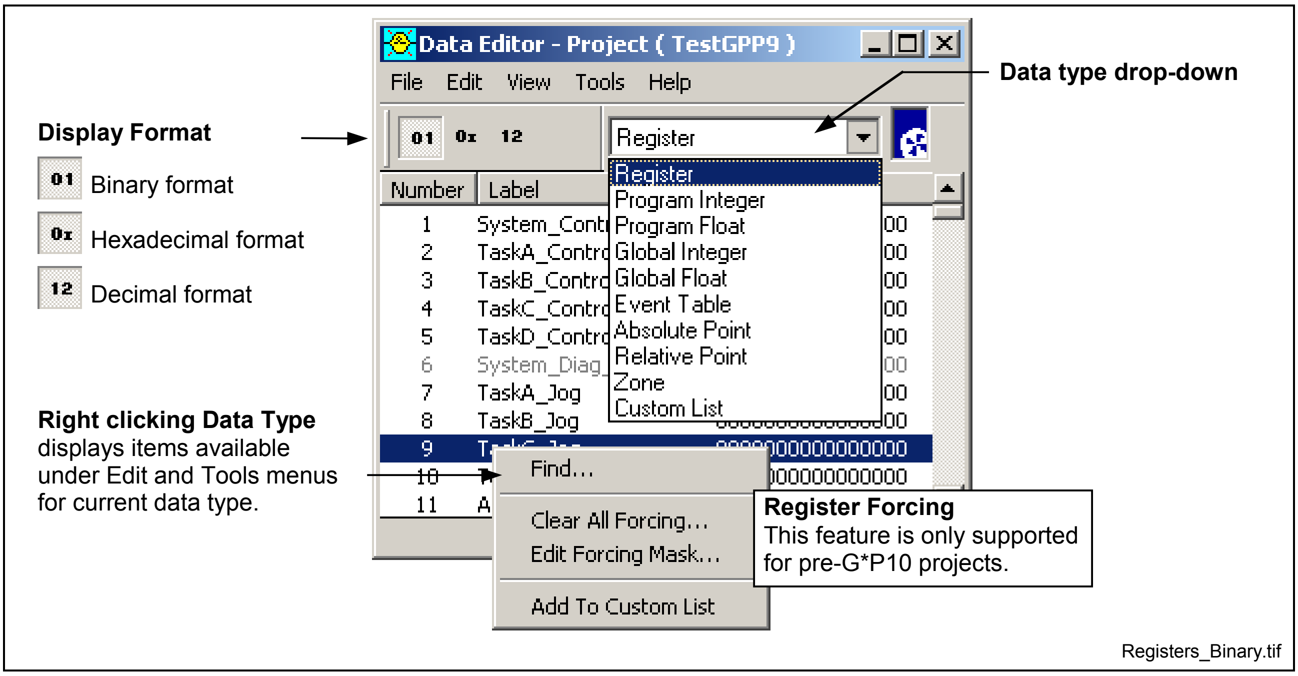


Fig. 8-32: Registers

Note: VisualMotion register forcing options are available only for pre-G*P 10 projects. Refer to VisualMotion 9 documentation for available register forcing options. I/O forcing for G*P 10 or 11 projects are controlled by the PLC through IndraLogic.

Register Update Priority

Register update priority is based on its usage in the system. Register update priority is listed as follows, from highest-to-lowest:

Highest

- I/O Task (I/O subsystem)
  - I/O Setup Inputs
  - I/O Mapping
  - I/O Setup Outputs
- Events
- User Program (Tasks A-D)

- I/O Bit icon
- Register Transfer icon

Lowest

- Serial Port
  - any access to registers over the serial port

Menu Items

File – *Program* allows the user to choose which program to display and *Exit* closes the window.

Note: *Program* selection is only available in service mode. Project mode will only read the current project data.

Edit – Allows the user the search for a specific data type Number, Label or Value by using the ***Find…*** feature.

View - Allows the user to change the displayed format of the register data between Hex, Binary, and Decimal.

Tools – The items available are dependent on the current data type selected and displayed. The following menu items are available:

- ***Control Selection…*** (service mode only)
- ***Force A Register…***
- ***Clear All Forcing…***
- ***Edit Forcing Options…***

Register Forcing
This feature is only supported for pre-G*P10 projects.

- ***Add to Custom List***
- ***Delete From Custom List*** (available in Custom List)
- ***Save Global Variables*** (available for Global Integers and Floats)

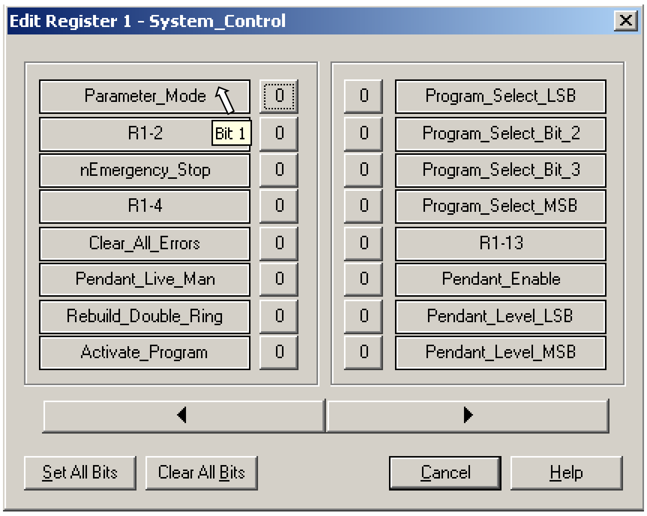
Help - allows the user to launch the help system and provide information about current VisualMotion software installed.

Edit a Register

To edit a register double click on the desired register to open the Edit Register window. The window displayed is dependent on the display format of the selected register.

Binary Register

Binary display registers are edited using the following *Edit Register* window. The register bit label is displayed (if configured) with the current state of the bit (0 or 1). To change the state of a bit click on the label or bit state (0 or 1).



Register_Edit_Bits.tif

Fig. 8-33: Bits of Registers

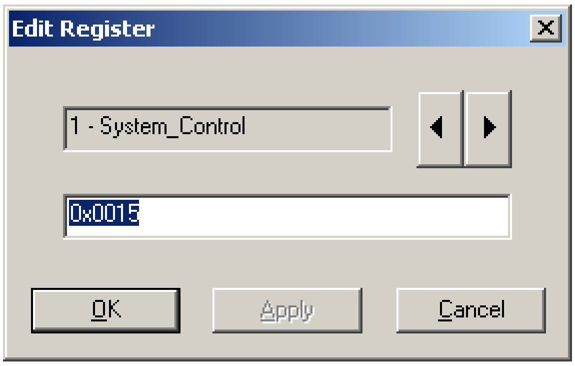
Register bit numbers are displayed as a tool tip when the mouse pointer is held over a register bit label or bit state, as illustrated in Fig. 8-33.

The ***Set All Bits*** and ***Clear All Bits*** buttons can be used to set or clear all register bits at one time.

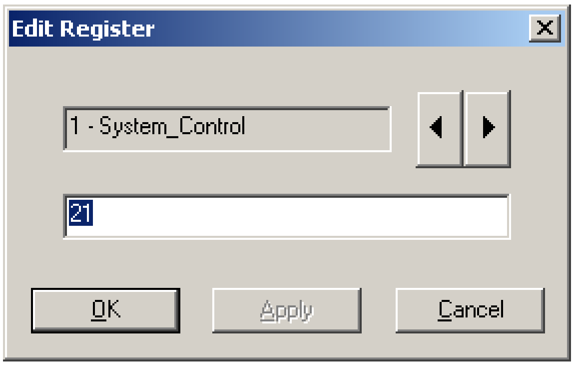
The ***left*** (previous) and ***right*** (next) arrow buttons allows the user to modify the bits of additional register without the need to exit back to the main register window. These buttons operate using a spin-control feature. This means the user can either advance to the next register or go back to the previous register

Hexadecimal and Decimal Registers

Hexadecimal and Decimal displayed registers are edited using the following *Edit Register* windows.



Hexadecimal Format

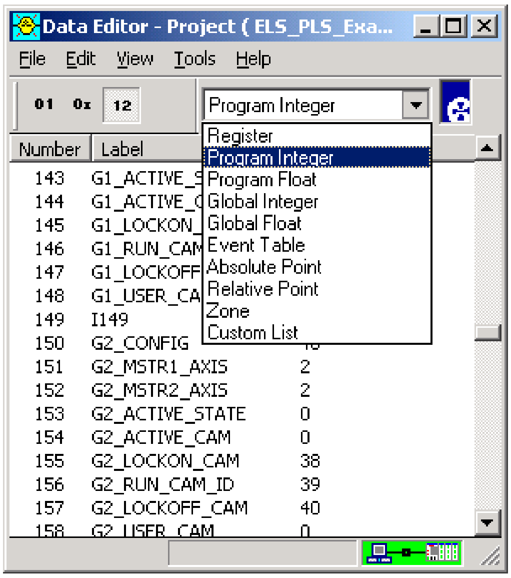


Decimal Format

Fig. 8-34 Hexadecimal and Decimal Registers

Variables

Selecting ***Data ⇒ Variables*** opens the *Data Editor Runtime Tool* window. The window automatically uploads and displays variables of the active project on the control. In service mode, a program is selected with ***File ⇒ Program Select***.



Variables.tif

Fig. 8-35: Variable

Saving Global Variables to Flash

Global variables are initialized to zero on power up and do not retain programmed values on power down unless they are saved to flash memory. The number of allowable global variables in a project is determined by control parameters C-0-0080 (Global Integers) and C-0-0081 (Global Floats). The default number of global integers and global floats in a project is 512 and 256, respectively.

Global variables can be saved to flash memory using one of the following methods:

- Data Editor Menu Selection
- Editing Command Parameter C-0-0082

Flash Global Variables from Data Editor

1. Set Rexroth VisualMotion Toolkit to service or online mode.
2. Switch the control to parameter mode.
3. Select ***Data ⇒ Variables*** and select either Global Integer or Global Float from the drop-down list.
4. Select ***Tools ⇒ Save Global Variables to Flash…***

Rexroth
Bosch Group

Flash Global Variables via Control Parameters

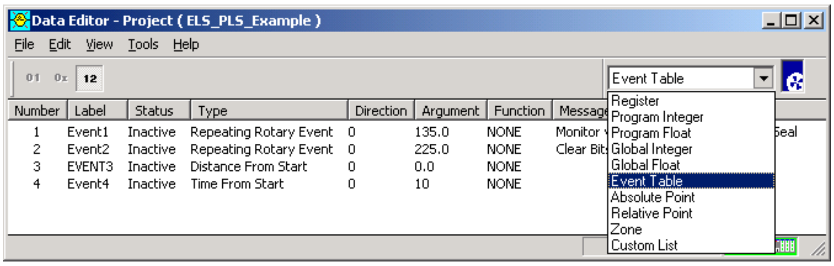
1. Switch the control to parameter mode.
2. Edit C-0-0082 and transition bits 1 and 2 from (0 to 1).
   Status parameter C-0-0083, bits 1-3 display a (1) when command is set.
3. Switch the control to manual or automatic mode.
   Status parameter C-0-0083 indicates success or error as follows:
   - Bits 1 and 2 set to (1) indicating a successful flash
   - Bits 1-4 all set to (1) indicating an error.

Global variables are now stored in flash memory and can be reinitialized with saved values.

Note: To save global variables again, edit C-0-0082 and transition bits 1 and 2 from (1 to 0) and repeat steps 1 - 3.

Event Tables

Events are used to execute an event function on time, position, angle or I/O State conditions. Each event has status, type, direction, distance or time, event function, and message fields. The Data Editor window permits viewing and editing of the event table of a program that has already been downloaded to the control. The window automatically uploads and displays the contents of the event table for the currently active program on the control.



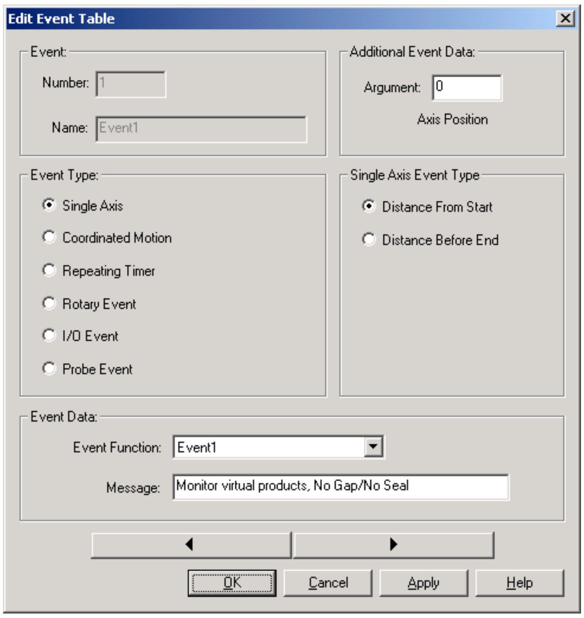
Event_List.tif

Fig. 8-36: Event Runtime Utility

Edit an Event

To setup or edit an event:

1. Double-click on the event to open the following Edit Event window opens.
2. Select an Event Type and options and click on the **Apply** or the **OK** button to send to the control.



Event_Edit.tif

Fig. 8-37: Edit Event Window

The following event types are supported:

- **Percentage of Coordinated Path**

  Events for Coordinated path are related to the **start** or **end** of a path segment as a percentage of the total path distance on the path segment.

- **Time in Coordinated from Start/before End**

  Coordinated motion can provide time-based events that are related to travel time along a specified geometry segment and are initiated by the path planner. These events execute at a fixed time after motion starts or before motion ends in the specified segment.

- **Single Axis Distance**

  Events for single axis motion may be set to take place at an absolute distance from the **start** or **end** of the axis move.

- **Repeating Timer**

  A cyclic event triggered by a continuously Repeating Timer.

- **Rotary (Repeating Axis Position)**

  An event triggered each time the configured axis encounters an absolute position. The axis motion type can be single-axis, ELS, ratio or velocity mode and configured for modulo or non-modulo positioning.

- **Task Input Transition**

  This event type triggers events on a low to high transition of a Task Control Register interrupt event bit. IndraLogic is used to map a specified I/O register condition from the I/O register to bit 9 in the appropriate Task Control Register. The approximate latency is <= 2 milliseconds.

- **Feedback Capture**

  This event type uses the Probe capability of Rexroth drives to trigger a Rexroth VisualMotion 11 event based on a positive or negative transition of the drive's Probe 1 or Probe 2 input.

- **I/O Register Event**

  This event type uses the bits of input Register 88 (USER_X1_REG) to trigger up to 16 events **only** in Task A. Register 89 (USER_X0_REG) is used to monitor the status of events triggered by Register 88.

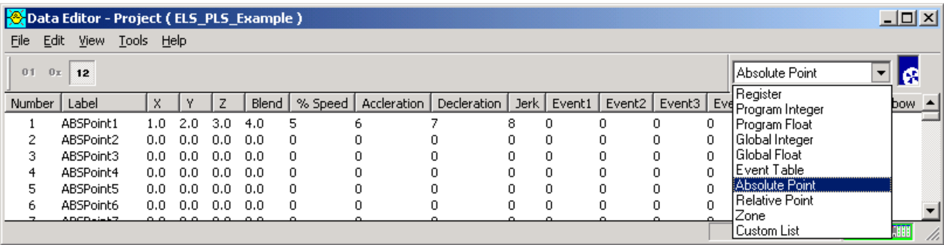
- **PPC-R X1 Input**

  This event type uses the PPC-R’s X1 digital inputs (I1, I2 and I3) to trigger an event based on a positive or negative transition.

Point Tables

From the Data Editor, the user can view and edit absolute and relative point tables in the currently active program.

Note: Relative and Absolute points are added to a project by selecting the VM Data Table, selecting the REL or ABS Points tab and clicking on the Add button. Refer to section 13.3, *VM Data Table*, in volume 2 of the *VisualMotion Functional Description* for details.



Points.tif

Fig. 8-38: Points Window

The window automatically uploads and displays the contents of the point table in the current project on the control. Points are used in coordinated motion programs to describe a location in Cartesian coordinates, tool orientation and associated events.

Note: The point table can also be referred to as a display of raw information for building CAMs using the Rexroth VisualMotion CamBuild Icon. Refer to the CamBuild Icon in chapter 14, volume 2 of the *VisualMotion 11 Functional Description* for details.

File

The file menu has the following selections:

- **Program Select** allows the user to select an active program. (available only in service mode)
- **Exit** closes the Points windows.

View

The ***View*** menu presents a list of the point elements and allows the user to select or deselect any of them to determine which ones appear in the display.

Table

The ***Table*** menu allows the user to switch between the absolute and relative point tables.

Update

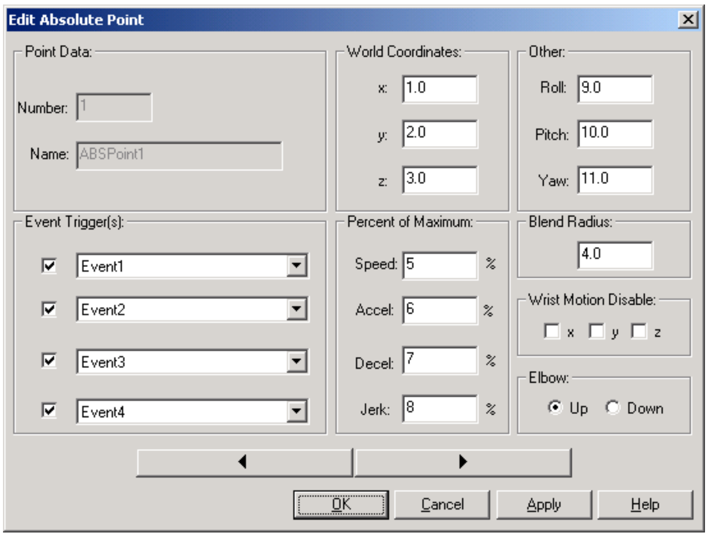
Clicking on ***Update*** refreshes the window and updates any changes that have been made.

Settings

Clicking on ***Settings*** opens the *Control Selection* window allowing the user to select a different method of communication.

Edit

The user can highlight a point and click Edit to make changes to existing point values. Another option is to double-click the desired point in the list. Editing a point table entry opens an absolute or relative Edit Point Values window, depending on which type was selected with the Table menu.



Points_Edit.tif

Fig. 8-39: Edit Points

The **Edit Point Values** window permit individually changing the values for each point table entry on the control. Clicking on the **Apply** button immediately downloads the new values to the program on the control. The changed values are displayed at the next automatic update.

Note: Coordinated moves using an ABS/REL table point with zeroes in any of the fields for Speed, Acceleration, Deceleration, and Jerk will default to 1%.

The first time a point is taught (and has zeroes in the Speed, Acceleration, Deceleration and Jerk fields) default values are loaded. The defaults are 10% for speed, 100% for acceleration, 100% for deceleration and 100% for jerk.

In the *Coordinates/wrist motion disable* section of this screen, the user can disable coordinated motion on one or more axes for a particular point.

In the **Elbow Properties** section of this screen, the user can choose the Up or Down radio button to determine the elbow direction for a particular point. To include this feature in the VisualMotion program.

Each point has: {x, y, z, blend, speed, acceleration, deceleration, jerk vent 1, event 2, event 3, event 4, roll, pitch, yaw, and elbow} fields.

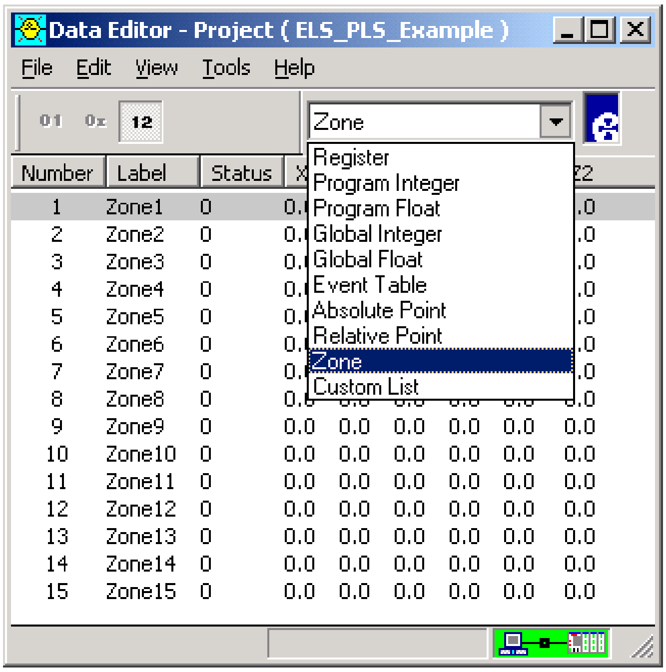
To better navigate long tables, the user may click and drag the vertical scroll bar button while looking at the point number indicator that appears at the right of the window title bar. The point number corresponds to the top of the list when the scroll bar button is released. The user must expand the viewing window to view the additional roll, pitch, yaw and elbow properties used with a six-axis control system.

Zone Tables

Zones can be used in coordinated motion programs to describe a volume of space where motion of any kind is prevented.

Programs on control - program selection is through a menu selection containing a list of programs on the control plus "Currently active".

To edit an event, select it in the list box, and press the **Edit** menu selection, or double click on the list item. Modify the necessary fields and press the ***Apply*** button to save the changes and keep the edit window open or click the ***OK*** button to save the changes and close the edit window.



Zones.tif

Fig. 8-40: Zones List

Each zone is defined by the { x, y, z } coordinates of two opposite corners (Point 1 and Point 2) of a cube in space. For each defined zone, the status can be ACTIVE (checked) or INACTIVE (unchecked) for selected tasks, no task or all tasks. For example, the zone defined in the “Edit Zone 2 Values” screen below is active for zones A and D.

Custom List

The Data Editor allows the user to create a custom list of the data types available from the data type drop-down list.

Create a Custom List

A custom list containing registers and program variables can be created to simplify the view of registers and program variables in a project.

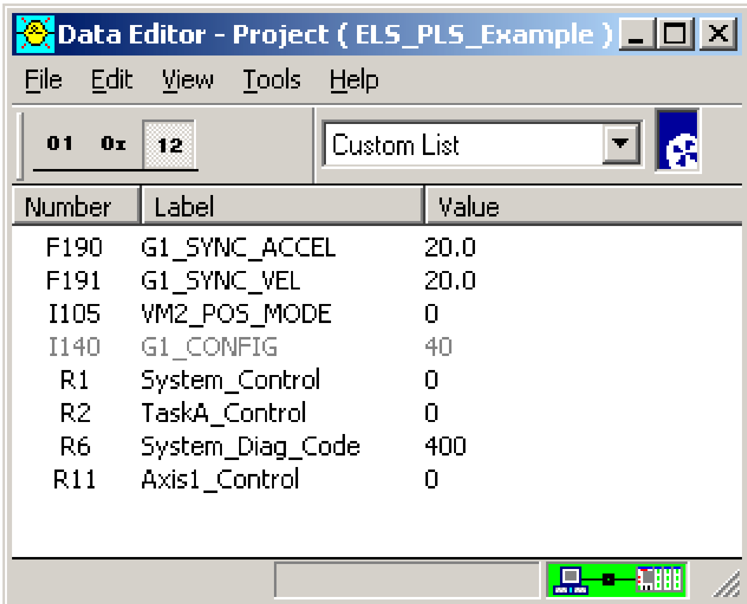
To create a custom list…

1. Display either registers or program variables from the drop-down list.
2. Using the mouse, right click or the desired data item and select Add to Custom List.

Note: Items can also be selected as a group by selecting the first item, holding the Shift key down, and selecting the last item and then right clicking to select the ***Add to Custom List*** button.

---

3. When complete, select *Custom List* from the data item drop-down list to display your custom list. The *Number* column identifies the data type added to the custom list.



Note: The data types displayed can be sorted in ascending and descending order by clicking on the column heading (Number, Label or Value).

---

4. To remove a data item from your custom list, right click over the data item and select *Delete From Custom List*.

8.3 ELS Runtime Tool

This tool is designed for modifying the default program variables used for parametrizing ELS data. These are variables that were initialized from the ELS process setup and any ELS icons that were used in the Initialization task. These variables are used to control the moving, stopping and jogging of the following multiple master components:

- ELS Masters
- ELS Group Masters
- Virtual Masters

Note: An active ELS program is necessary for the ELS Runtime Tool to open.

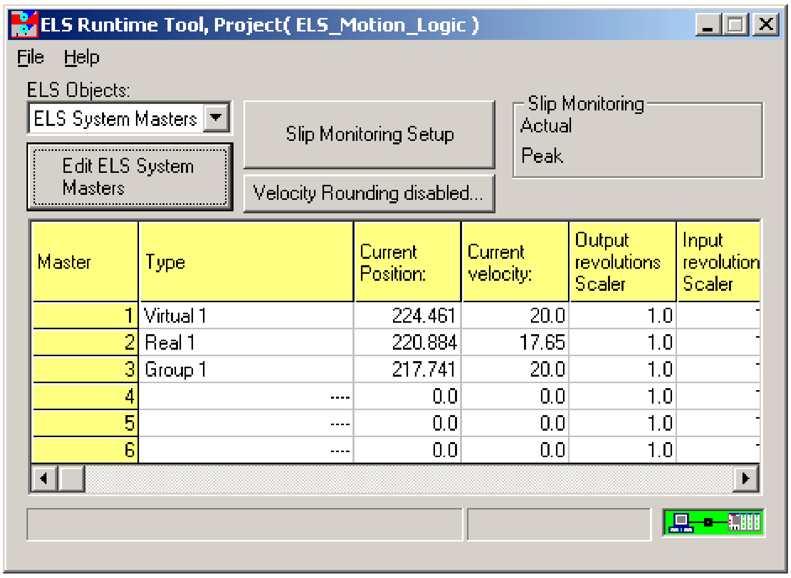


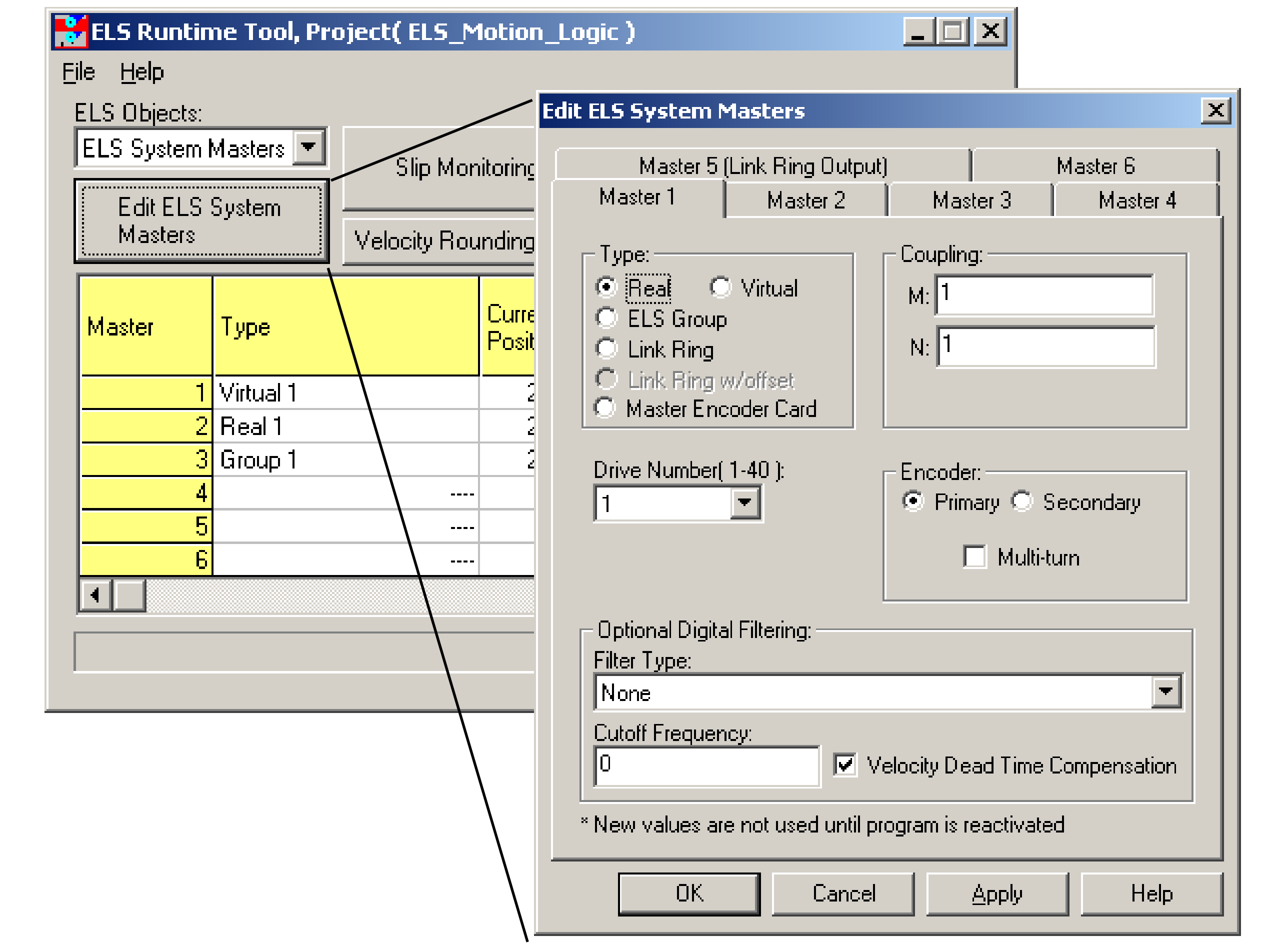
Fig. 8-41: ELS Runtime Utility

Modifications to these values overwrites the initially compiled and downloaded values on the control. The values remain active until they are overwritten (e.g. by more changes using the runtime utility or by compiling and downloading a new program to the control).

The **ELS Objects** list box at the top left allows selection of any ELS Masters, ELS Groups or Virtual Masters that have been setup. The button below the list box changes, depending on the ELS object selected.

ELS System Masters

Selecting **ELS System Masters** and clicking the button labeled "Edit ELS System Masters" opens the **Edit ELS System Masters** window below.



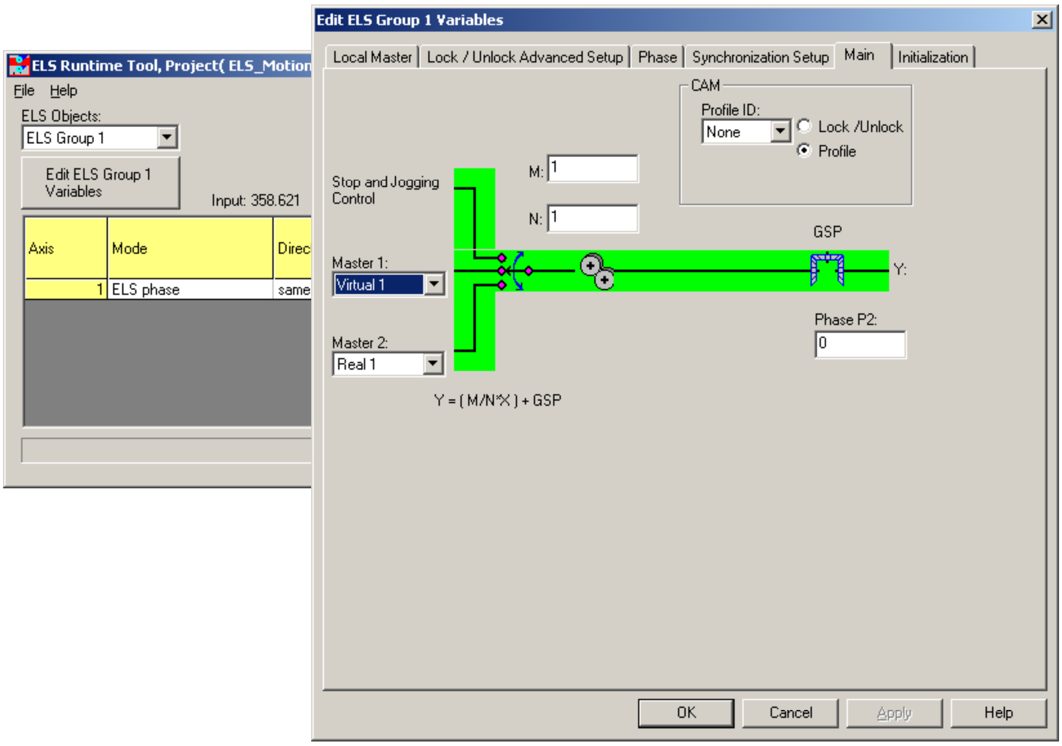
ELS_Masters.tif

Fig. 8-42: Edit ELS System Masters

For more information about the initial setup of system masters and setting the values in this window, refer to the section 4.5, *Setup ELS Process*, for details.

ELS Group

Selecting **ELS Group x** in the list box and clicking the button labeled "Edit ELS Group x Variables" opens the **Edit ELS Group Variables** tab array below.



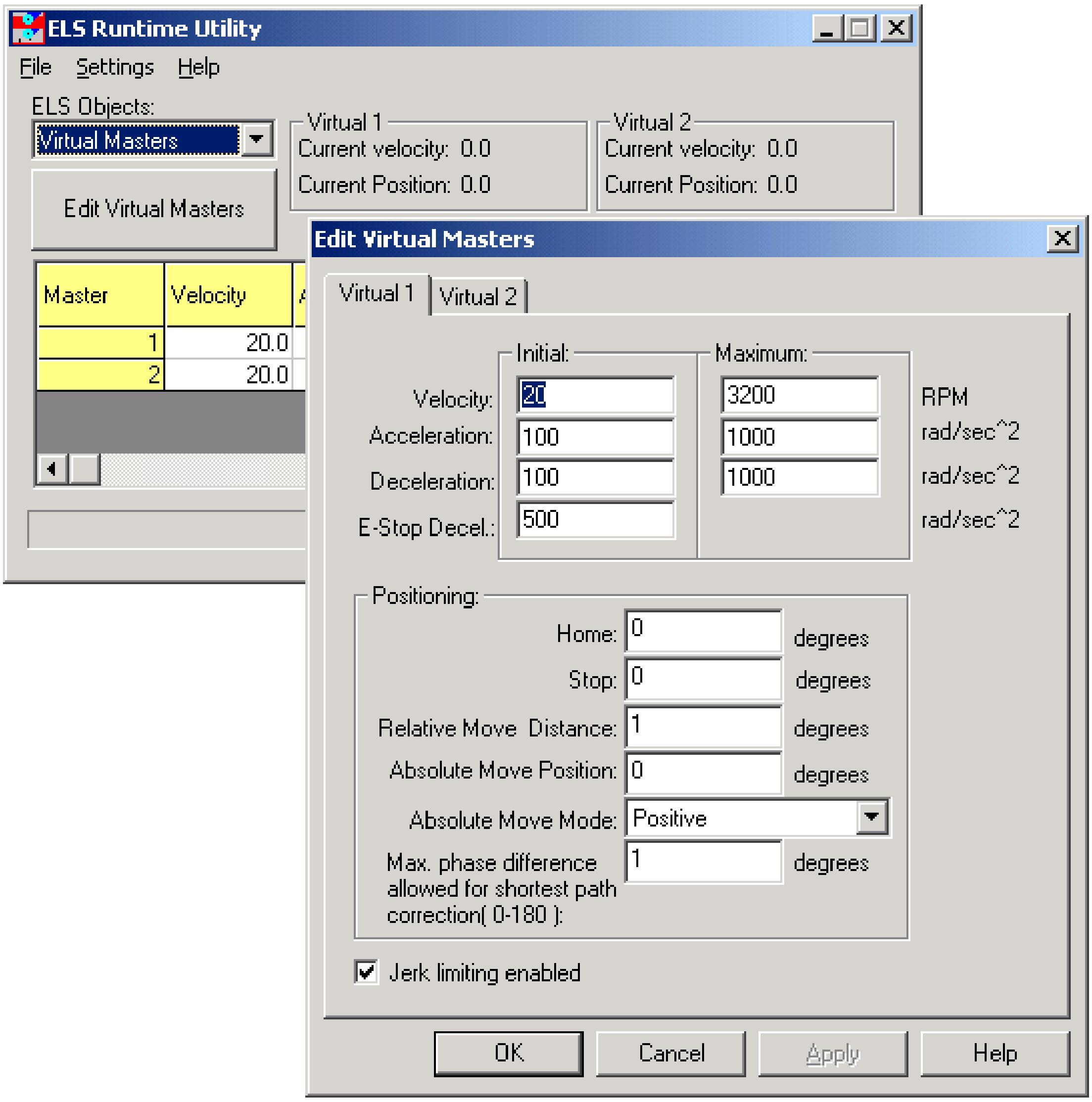
ELS_Group_Variables.tif

Fig. 8-43: Edit ELS Group x Variables

The tabs open windows that are accessible in the initial setup of ELS elements, using the ***ELSGrp*** icon.

Virtual Masters

Selecting **Virtual Masters** and clicking the button labeled "Edit Virtual Masters" opens the **Edit Virtual Masters** window below.



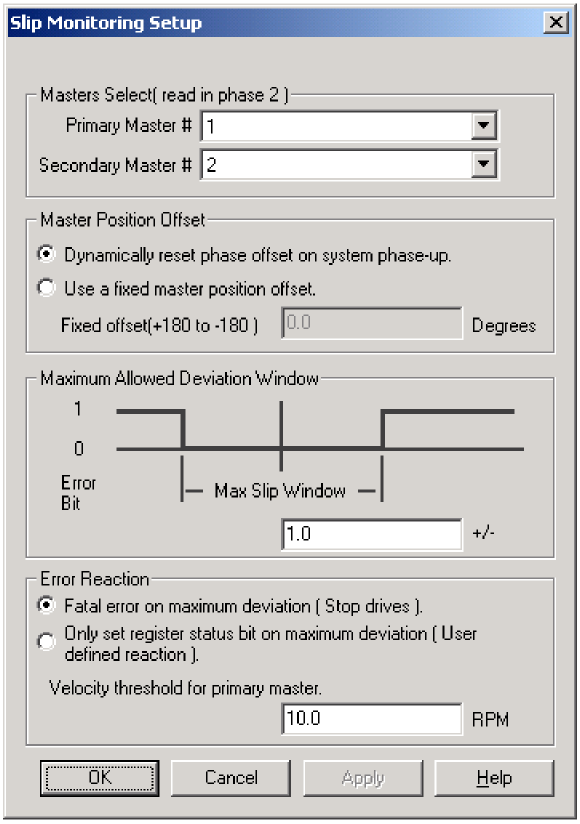
ELS_Virtual_Master.tif

Fig. 8-44: Edit Virtual Masters

For more information about the initial setup of Virtual Masters and setting the values in this window, refer to the section 4.5, *Setup ELS Process*, for details.

Slip Monitoring Setup

ELS System Master slip monitoring is used to initiate an error reaction when the phase difference between two system masters exceeds the maximum allowed deviation window variable. Refer to section 6.1, *Electronic Line Shafting*, in volume 1 for details.



SlipMonitoring.tif

Fig. 8-45: Slip Monitoring Setup

8.4 CAM Indexer

This tool allows for on-the-fly changes to a CAM Indexers in the active program that was setup using ***Setup ⇒ Processes ⇒ CAM Indexers*** or the **CAM Indexer** Icon. Variable assignment cannot be changed using this utility. Two options for correction are available in this runtime-utility (bottom left of window), which cannot be changed using the icon window:

- Reversals allowed
- Apply remaining correction to next index

Refer to section 4.6, *CAM Indexer Setup*, for details.

Registration Setup

Registration is the process of referencing a product edge or register mark. This allows positioning errors to be detected and corrected before an upstream (web, product) or downstream (die-cutter, print cylinder, etc.) process takes place, depending on the machine design.

This function tightly couples the control system with the **Probe Event** on the drive that the product is being referenced to. Registration is accomplished by comparing the captured position to a target value and correcting for the difference. The registration error is automatically assimilated into the axis motion profile, providing a smooth, seamless correction. The registration error can be corrected using S-curve, Triangular, and Trapezoidal correction profiles.

The Registration function integrates several tasks. It automatically executes a high priority process that is triggered by an axis probe event.

This process generates a *registration error* based on the settings of the drive, system and registration parameters.

The control will then automatically reconcile details such as rollover, probe On/Off position window, setting drive parameters (edge, arming, etc…), and routing the correction data to the appropriate axes.

The registration function is compatible with axes defined in ELS Phase, Drive CAM, and control CAM (table or indexer) modes. The values needed for registration are read from and written to the control with user program variables. Up to four axes can use the registration function.

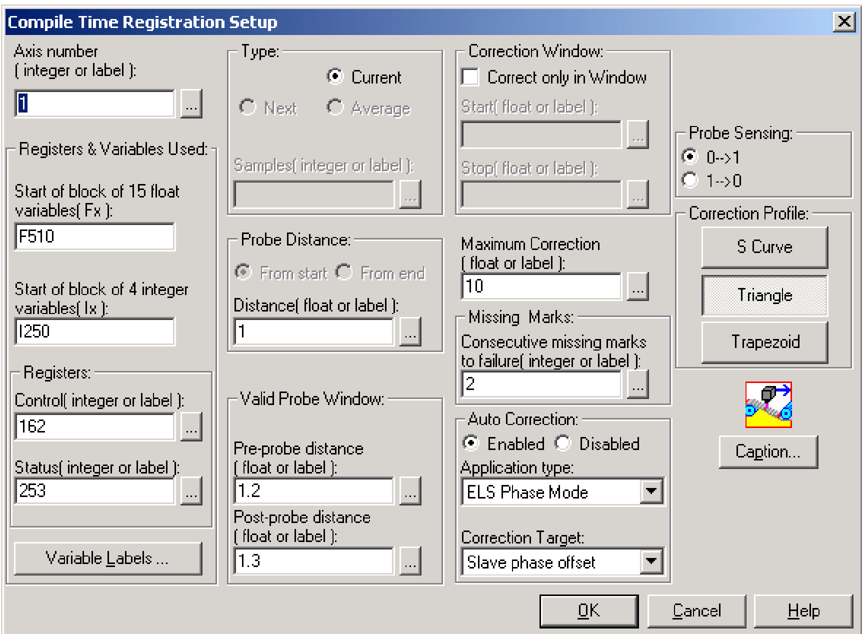
The Registration icon window assigns variables and I-O registers, and initializes axis parameters. All values are assigned to blocks of float and integer variables. This allows them to be read or written from a user interface or a PLC with no additional program code.

When the CAM Indexer is used, it performs the registration correction with a profile based on the *Correction Distance*. For other types of axes, a relative phase offset (either master or slave) is sent to the drive, and the drive performs the correction. The phase offset, equal to *Correction Distance*, is applied according to the *Correction Procedure* and *Correction Maximum* parameters.

The registration function can be used with the following Rexroth drive firmware versions:

DIAX04 - ELS-04VRS or later

ECODRIVE03- SGP-01VRS or later



CAM_Indexer_Compile.tif

Fig. 8-46: CAM Indexer Compile Time Registration

Registration Parameters

The following Parameters are used by the registration function. Most of them are written by the user with the registration icon window. Some of them are also calculated and written by the control or the Drive.

Parameters	Description	Variable	Written by
Type	Current, Next, Average	Int 0 bits 0-1	user
Probe Distance from Start/End	Select Setpoint to be calculated from start/end of move	Int 0 bit 2	user
Probe Distance	Expected position of mark from start or end of move	Float 0	user
Mark Setpoint	Absolute expected position of mark to calculate error	Float 6	control
Correction Distance	Correction distance (Error amount)	Float 7	control
Probe Value	Feedback position at probe 1 trigger	Float 8	drive
Valid Probe Window Start	Distance before Mark Setpoint to look for mark	Float 1	user
Valid Probe Window End	Distance after Mark Setpoint to stop looking	Float 2	user
Correction Window	Correct only in the correction window.	Int. 0 bit 12	user
Correction Window Start	Correction to start at this point	Float 3	user
Correction Window Stop	Correction to be completed before this point	Float 4	user
Maximum Correction	Maximum correction per period	Float 5	user
Missing Marks	Number of missed marks allowed	Int. 1	user
Auto Correction	Enable correction sent to phase adjust, etc.	Int. 0 bit 8	user
Correction Target	Select correction target: master offset, slave offset, etc.	Int. 0 bit 9-10	user
Position Source	Measure Master or Slave position	Int. 0 bit 4	user
Probe Sensing	Use Leading/Trailing edge of mark	Int. 0 bit 3	user
Correction Profile	Shape of the correction CAM	Int. 2	user

Table 8-9: Registration parameters

Type

This selects the desired type of registration correction calculation to be used.

- Current:
- Next:
- Average:

Samples (Integer or Label) -

Probe Distance from Start/End

This bit determines if the *Mark Distance* is relative to the Start or End of move. This option applies to CAM axes that use the indexer feature. For other modes, the mark distance is an absolute position.

0 - End of Move

1 - Start of Move

Probe Distance

This value is used to determine the calculated position of the axes that a mark is expected to detected.

CAM Indexer:

Based on the setting of *Mark at S/E* bit, the *Mark Setpoint* value is automatically calculated as follows. *Starting Position* is the position of the axis at the start of the index. *Ending Position* is the expected position at the end of the index (minus registration correction).

If Start of move: *Mark Setpoint* **=** *Starting Position* **+** *Mark Distance*

If End of move: Mark Setpoint = Ending Position - Mark *Distance*

ELS Phase, Control Table CAM, or Drive CAM Mode:

The *Probe Setpoint* is an absolute position on either the master or the slave (set via an integer option bit).

Probe Setpoint = Probe Distance

Mark Setpoint: This value is used to calculate registration error and various position points used in the registration process. See Correction Distance, Probe Window On/Off.

Correction Distance: This is the amount of correction distance that will be applied to the current index in units. Based on the selected *Correction Profile* type this error is added to the CAM profile or ELS slave position at the position defined by the correction window. Only one value is read during each index cycle.

Correction Distance = Mark Setpoint - Probe Value

In future versions, correction distance may be optionally buffered through several cycles in case the measurement is upstream, or may be averaged or filtered for a smoother correction.

Probe Value: This value indicates the measured probe position value that was captured by the drive when the registration mark was detected on the probe 1 input. It is saved only when the axis is within the probe window. It is used in the calculations associated with determining registration error.

Valid Probe Window Start/End

This window selects the period within the move cycle that the registration mark can be read. The probe position is captured and the correction distance is calculated only when the mark is found within this position range. The probe window is related to the selected measurement value (master or axis position).

Correction Window

This bit enables the period within the move cycle that the correction can be applied (defined by “Correction Window Start/End”). If this bit is clear, the correction is applied immediately after the probe position is captured.

Correction Window Start/Stop

This window selects the period in the move cycle to apply the registration correction. If the option *Correction Window Enabled* is not selected, the correction is applied immediately after the probe position value is captured. Otherwise, the correction starts at *Correction Window Start* and finishes before *Correction Window End*.

When using the CAM indexer, the start and end positions in relation to the master axis are entered in these parameters. Correction motion will only take place during this window. If the probe comes in during the correction window, for example at position x of the master, the maximum allowable correction during that cycle is calculated as follows:

(maximum correction)(end - x)/(end - start)

For other motion types, the positions are related to the measured value (master or axis position). The correction is added to the value selected in the *Correction Target* option. The drive handles the phase offset internally, based on its velocity, acceleration, and/or filter parameters. It is necessary to set these drive parameters so that the correction move finishes before the next cycle.

Maximum Correction

This is the maximum amount of correction distance that will be applied to the current index in units. If zero, the full amount of the correction is used for the current index. If a value is entered, the *Correction Distance* value is compared to *Correction Maximum* and correction up to this value will be applied. The remainder will be discarded.

Missing Marks

This defines the number of consecutive missed marks that are allowed before the control sets the Max Missed Marks status bit.

Auto Correction

By default, the control sends the *Correction Distance* to the drive to perform a phase offset, based on the correction window and correction target. It is possible to disable automatic correction, so that it can be handled in the user program. This option applies to all motion types except for the CAM indexer, for which automatic correction is always enabled.

0 - Enable Automatic Correction

1 - Disable Automatic Correction

Correction Target

These bits select the target parameter for the *Correction Distance.* When using the CAM indexer, the correction uses a CAM which is added to the CAM position generated by the indexer. For other modes, the following selections determine the affected parameter:

Bit 10	Bit 9	ELS Phase Sync	ELS Velocity Sync	Drive CAM	Control CAM (Table)
0	0	slave phase offset (S-0-0048)	additive velocity command (S-0-0037)	master phase offset (P-0-0061)	master phase offset (A-0-157)
0	1	invalid	ratio fine adjust (P-0-0083)	slave phase offset (S-0-0048)	slave phase offset (A-0-162)
1	0	invalid	invalid	CAM shaft distance (S-0-0093)	CAM H factor (A-0-33)

Position Source

This bit determines if the slave axis (0) or ELS master (1) position is used in the registration correction calculation. This option applies to ELS, control CAM (no indexer), and drive CAM modes only. With the CAM indexer, only the slave position can be measured. If slave position is enabled, the *Probe Value* captured by the drive is the axis feedback position. If master position is enabled, the *Probe Value* is the ELS master position.

0 - Axis Feedback Position

1 - ELS Master Position

Probe Sensing

This bit determines if the drive captures the axis position at the leading or trailing edge of the mark. The leading edge is indicated by a high (0-1 transition) on the probe input.

0 - Leading Edge

1 - Trailing Edge

Correction Profile

When using the CAM indexer, this value defines the shape of the correction CAM. Current selections are: S-curve, Triangular, and Trapezoidal.

Registration Control and Status Registers

The control and status registers that are used for each registration axis are allocated in the registration user program command.

Registration Control Register

10 - Enable Registration

Registration Status Register

9 - Correction at Max

10 - reserved

11- Mark in Window

13 - Registration Enabled

14 - Max Missed Marks

Enable Registration

Set this bit to (1) to enable the registration function. All relevant parameters will be initialized, then the 'Registration Enabled' bit in the registration status register will be set. To disable the registration function, set this bit to (0).

Correction at Max

This bit is set to (1) when the *Correction Distance* exceeds the *Correction Max* parameter. It is reset to 0 after the next mark is found and *Correction Distance* is less than or equal to *Correction Max.*

Mark in Window

This bit is set to (1) as soon as a registration mark has been detected within the probe window. It is cleared when registration is first enabled, or on the following cycle when no registration mark is found in the window. This bit can be mapped to a task input or system input event, so that the user can run additional program code when the mark is detected.

Registration Enabled

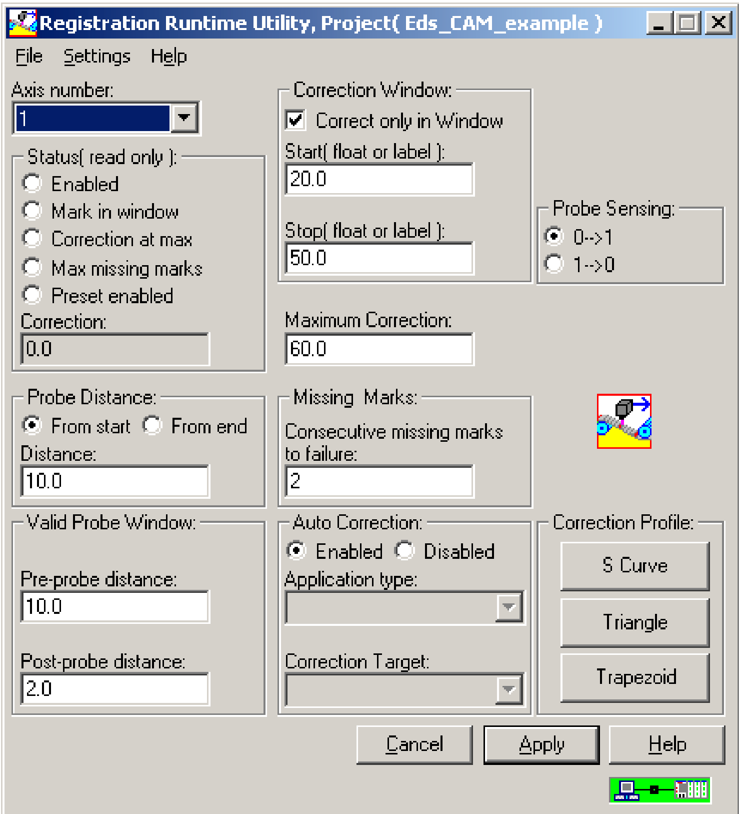
This bit is an acknowledgment of the Enable Registration control bit. Its value will not match that of the control bit until the control has completed the process of enabling or disabling registration. This process could take several milliseconds, depending on the parameters that need to be initialized.

Max Missed Marks

A missed mark counter is incremented when the probe window is traversed without a mark being found. This bit is set when the counter exceeds the *Missed Marks* parameter. The user program or PLC can then take appropriate action or alert the user of this condition. This bit is reset by the control only upon a 1-0 transition of the Registration Enable bit.

Registration Runtime Tool

Selecting ***Data ⇒ Registration…*** opens the Registration Runtime Tool. This tool is only available if the active program contains a valid registration function.



Registration.tif

Fig. 8-47: Registration Utility

Status (read only)

This setting reads the state of the status register setup for the registration functionality.

Enabled

This bit is an acknowledgment of the Enable Registration control bit. Its value will not match that of the control bit until the control has completed the process of enabling or disabling registration. This process could take several milliseconds, depending on the parameters that need to be initialized.

Mark in Window

This bit is set to (1) as soon as a registration mark has been detected within the probe window. It is cleared when registration is first enabled, or on the following cycle when no registration mark is found in the window. This bit can be mapped to a task input or system input event, so that the user can run additional program code when the mark is detected.

Correction at Max

This bit is set to (1) when the *Correction Distance* exceeds the *Correction Max* parameter. It is reset to 0 after the next mark is found and *Correction Distance* is less than or equal to *Correction Max.*

Max Missed Marks

A missed mark counter is incremented when the probe window is traversed without a mark being found. This bit is set when the counter exceeds the *Missed Marks* parameter. The user program or PLC can then take appropriate action or alert the user of this condition. This bit is reset by the control only upon a 1-0 transition of the Registration Enable bit.

Preset Enabled

This bit is an acknowledgment of the Enable Preset control bit.

Probe Distance

This function is to select the distance (from start or from end) within the move cycle that the registration mark can be read. The probe position is captured and the correction distance is calculated only when the mark is found within this position range. The probe window is related to the selected measurement value (master or axis position).

Valid Probe Window

This window selects the period within the move cycle that the registration mark can be read. The probe position is captured and the correction distance is calculated only when the mark is found within this position range. The probe window is related to the selected measurement value (master or axis position).

Correction Window

This function is to select the period in the move cycle to which the registration correction is to be applied. If the option *Correct only in Window* is not selected, the correction is applied immediately after, the probe position value is captured. Otherwise, the correction starts at the *Start* position, and must be finished before the *Stop* position.

When using the CAM indexer, the start and end position in relation to the master axis is entered here. Correction motion will only take place within this window. If the probe comes in during the correction window, say at position x of the master, then the maximum allowable correction during that cycle is calculated as follows:
(maximum correction)(x - start)/(end - start)

Select Correction Target - These bits select the target parameter for the *Correction Distance.* When using the CAM indexer, the correction uses a CAM, which is added to the CAM position generated by the indexer.

Maximum Correction

This is the maximum amount of correction distance that will be applied to the current index in units. If zero, the full amount of the correction is used for the current index. If a value is entered, the *Correction Distance* value is compared to *Correction Maximum* and correction up to this value will be applied. The remainder will be discarded.

Missing Marks

Enter the number of missing marks until an error is issued.

Auto Correction

Select enabled or disabled. By default, the control sends the *Correction Distance* to the drive to perform a phase offset, based on the correction window and correction target. It is possible to disable automatic correction, so that it can be handled in the user program. This option applies to all motion types except for the CAM indexer, for which automatic correction is always enabled.

Probe Sensing

Set the sensing to be either on the positive transition (0 ⇒ 1) or a negative transition (1 ⇒ 0).

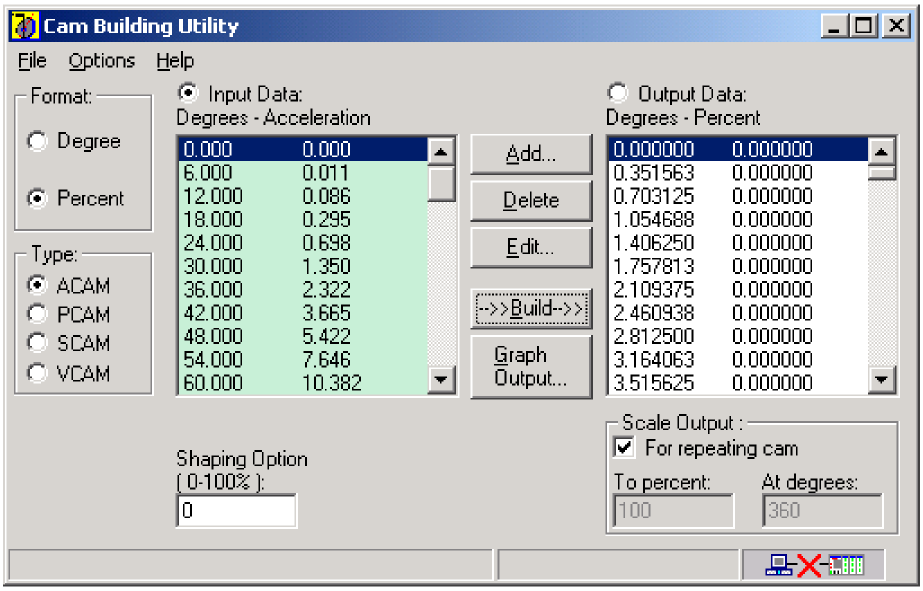
Correction Profile

When using the CAM indexer, this value defines the shape of the correction CAM. Current selections are S-curve, Triangular and Trapezoidal.

8.5 CAM Builder

This version of the CAM Builder is installed if the new CAM Builder tool is not selected during the initial installation of VisualMotion. This utility is used to build CAM tables for Rexroth's motion controls and drives.

Note: The new CAMBuilder documentation is only available in the form of an online help system. Press the F1 key after launching the new CAMBuilder.



CAM_Builder.tif

Fig. 8-48: CAM Building Utility Window

Format

This section defines the format, as *Degree* or *Percent*, that will be used to build a control or drive CAM. Pre-GPP 10 firmware can only build a control CAM using a degree format while GPP 10/11 or GMP 10/11 can use a degree or percent format. Drive CAMs can only be built using a percent format.

Degree CAM - output data is calculated where the master output position is in *Degrees* and the slave output position is in system *Units* (degree, in, mm, etc.).

Percent CAMs: - output data is calculated where the master output position is in *Degrees* and slave output position is in *Percent*.

Type

Select the algorithm that will be used to build the output data from the user-defined input data. The available types are:

- ***ACAM*** – slave input data is entered as an acceleration at a given master position in degrees.
- ***PCAM*** – slave input data is entered as a position at a given master position in degrees.
- ***SCAM*** – slave input data is entered as a position at a given master position in degrees. Output data is build by applying the input data to a 3rd order **spline** polynomial.
- ***VCAM*** - slave input data is entered as a velocity at a given master position in degrees.

Input Data

The *Input Data Table* displays the master and slave input values entered by the user or uploaded from an input file. When the **Input Data** radio button is selected, the *Input Data Table* becomes active and the background color changes to green. The available buttons are ***Add…***, ***Delete***, ***Edit…***, ***-->>Build-->>*** , and ***Graph Output…***

Input data are defined by clicking the ***Add…*** button and entering input position values for both the master and slave. Existing input data can be modified by selecting the relevant line and clicking the ***Edit…*** button. Once all input data is entered, click the ***-->>Build-->>*** button to calculate the output data.

Output Data

This *Output Data Table* displays the output position data for the master to slave relationship calculated from the input data. Also, the output position data for a control or drive CAM can be uploaded for editing from an output file or by using the ***CAM Transfer…*** function under the ***File*** menu. When the **Output Data** radio button is selected, the *Output Data Table* becomes active and the background color changes to green. The available buttons are ***Edit…***, and ***Graph Output…***

Output Steps

The *Output Steps* are only available for degree CAMs and defines the number of output steps (points) that will be used to build the CAM. The allowable range for output steps is 10 - 1025.

Shaping Option

Select the percentage of S-shaping to be used on the velocity curve to prevent a jump in acceleration (jerk limiting). This option is not available for SCAM profiles.

Output Modulo

The *Output Modulo* is only available for degree CAMS and defines the Output Modulo that will be used for scaling the range of ACAM and VCAM output calculations. By default, this scaling is set to 360.

Scale Output

The *Scale Output* is only available for percent CAMs and is used to scale the CAM's output profile.

For repeating CAM – by default, this checkbox is checked and is used for endless running CAMs, where the percent value is fixed at 100 and the degree value is fixed at 360.

To percent – scalar percentage multiplied to the CAM's H factor.

% x H factor = units of distance

At degree – master output position where the "To percent:" value is achieved.

Buttons

Add… - add an entry to the input data.

Delete - deletes selected entry in the input data.

Edit… - edits selected entry in the input or output data.

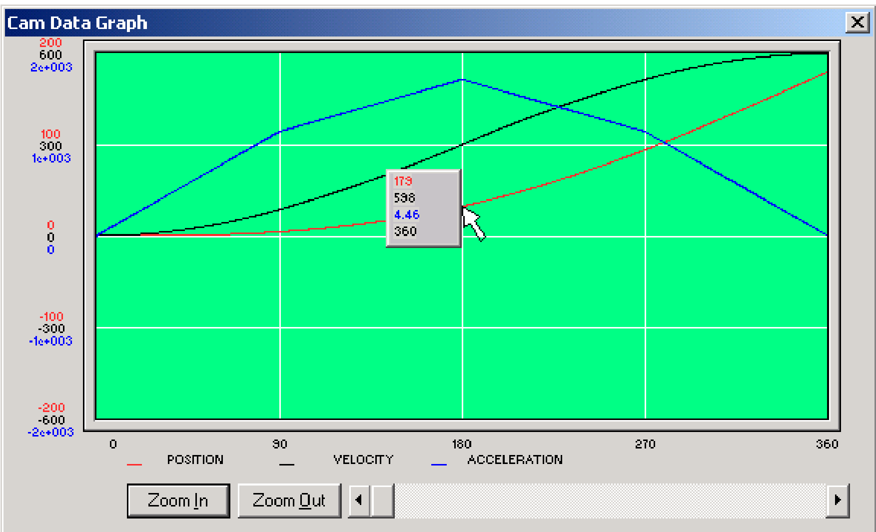
-->>Build-->> - builds position, velocity and acceleration CAM profiles and displays position data in the *Output Data Table*. Uses the algorithm for the selected *Type* to build the number of data sets chosen in *Output Steps* (10-1025 for degree CAMs and fixed to 1024 for percent or drive CAM) using the input data and the *Shaping Option* to modify.

Note: CAM for percent CAMs are allowed within the range of +/- 200%. If this range is exceed, an error will be issued and the slave output positions will be scaled back to 100%.

Graph Output… - creates or refreshes a graph of the position data in the *Output Data Table* using the speed in the *Simulation Speed for Graph* setting under the ***Options*** menu. The velocity is differentiated from the position and the acceleration is differentiated from the velocity.

Note: Measurement of any point of the graph can be taken by positioning the mouse over the desired graph portion and holding the left mouse button.

- **Zoom In/Out** – increases or decreases the size of the graph for more accurate measurements.

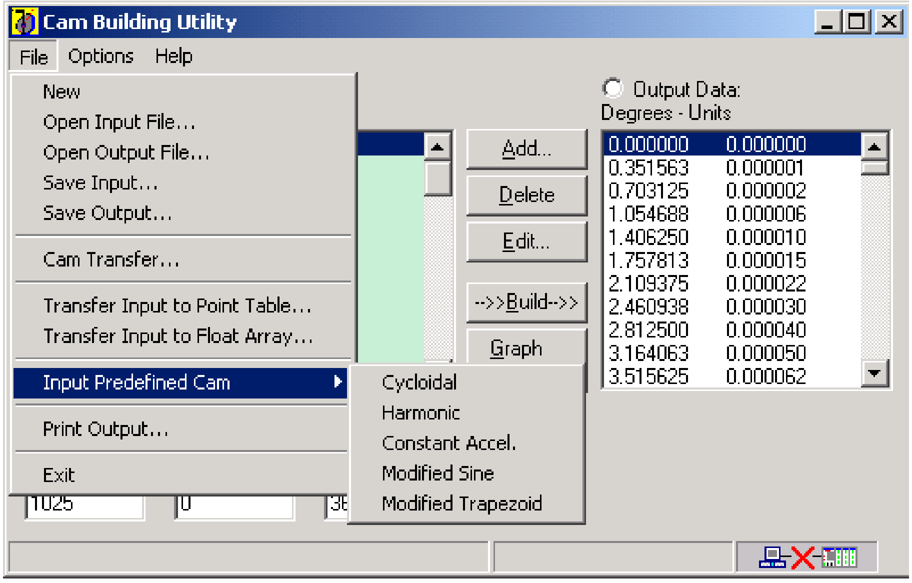


CAM_Data_Graph.tif

Fig. 8-49: CAM Data Graph

File Menu

The following figure displays the available menu selections under the ***File*** menu:



CAM_Builder_File_Menu.tif

Fig. 8-50: CAM Building Utility - File Menu

New

This menu selection clears the *Input* and *Output Data Tables*.

Open Input File…

This menu selection loads data from an input file (*.txt) into *Input Data Table*.

- User-selected input file. Input files are assumed to have the following format:

File identifier string.	;first line only
master value, slave value	;second line
master value, slave value	;third line
----	---
----	---
master value, slave value	;last line

Minimum and maximum number of data sets:

Build Types	ACAM	PCAM	SCAM	VCAM
Minimum Sets	2	2	5	2
Maximum Sets	1024	512	200	1024

Note: A 5^{th} order polynomial will be used when the maximum values are exceeded for PCAM and SCAM types.

Open Output File…

This menu selection loads data from an output file (*.csv) into *Output Data Table*.

- User-selected output file. Output files are assumed to have the following format:
  CAM output files are sent to the control using Rexroth VisualMotion.
  master value, slave value ;first line
  master value, slave value ;second line

Rexroth
Bosch Group

---- ---
master value, slave value ;last line

Save Input…

This menu selection saves values from the input data to a user-selected input file. The file is saved using text format (*.txt). Refer to *Open Input File…* for file format.

Save Output…

This menu selection saves values from the output data to a user-selected output file. The file is saved using standard spreadsheet format (*.csv). Refer to *Open Output File…* for file format.

CAM Transfer…

The *Transfer CAM* window is used to upload (Get), download (Send), or delete existing control or drive CAMs from a project in offline mode or from the control or selected drive in online or service mode.

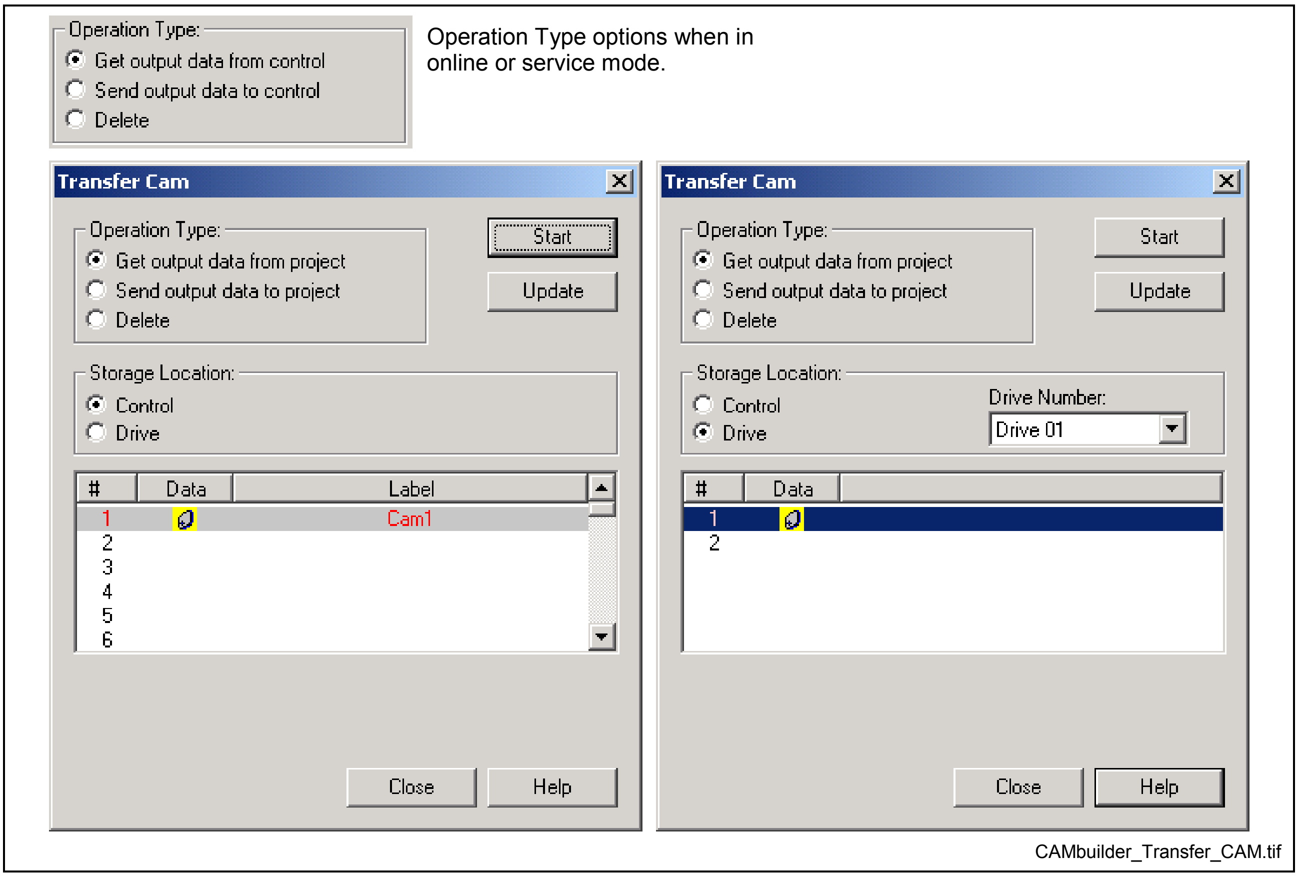


Fig. 8-51: Transfer CAM

Note: A CAM graphic under the **Data** column is an indication that the control or drive contains a CAM. Only existing control and drive CAMs can be retrieved or deleted.

- **Get output data from project (control) –** retrieves the selected CAM output data for the specified **Storage Location** (Control or Drive) from the project in offline mode and from the control's memory in online or service mode. An uploaded CAM is displayed in the *Output Data Table* of the *CAM Builder* utility.
- **Send output data to project (control) –** downloads the current output data in the CAM Builder to the project in offline mode and to the control or drive in online or service mode.

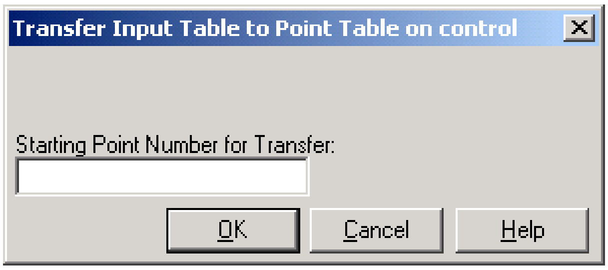
- **Delete** - deletes the selected CAM number from the specified storage location (Control or Drive) from the project in offline mode and from the control's memory in online or service mode.
- **Start -** this button becomes active under the following conditions:

Operation Type	Condition
Get output data from project (control)	An existing control or drive CAM is selected.
Send output data to project (control)	*Output Data Table* in CAM Builder contains data and a control or drive CAM number is selected
Delete	An existing control or drive CAM is selected.

Table 8-10: Start Button Activation

Transfer Input to Point Table…

This window is used to transfer the contents of the Input Data Table to a set of ABS points located in a project in offline mode or on the control in online or service mode.



CAM_Transfer_Input.tif

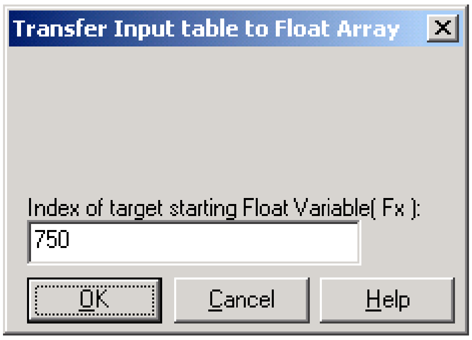
Fig. 8-52: Transfer Input Table to Point

- **Starting Point Number for Transfer** - Enter the beginning ABS point number to where the data should be copied. The values of the first line of the input data will be copied to this point. The values from each additional line of the input data will be copied to the next point. The number of consumed ABS points depends on the number of lines in the *Input Data Table*. The **OK** button starts the checking process and transfer. Before the transfer is started, checks are made for valid point and valid range. A window opens for confirmation. The **Cancel** button exits this window.

Rexroth Bosch Group

Transfer Input to Float Array…

This window is used to transfer the contents of the *Input Data Table* to a set of consecutive program floats in a project in offline mode and to the control in online or service mode.



CAM_Transfer_Float.tif

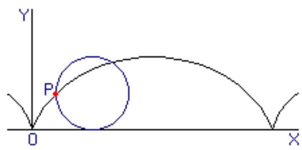
Fig. 8-53: Transfer Input to Float Array

- **Index of target starting Float Variable(Fx)** - Enter the program float number that will be the start for the transfer. The **OK** button starts the checking process and transfer. Before the transfer is started, checks are made for valid program float and valid range. A window opens for confirmation. The **Cancel** button exits this window.

Input Predefined CAM ▶

This menu selection uploads the input data of a predefined CAM file for the following types:

- Cycloid
- Harmonic
- Constant Acceleration
- Modified Sine
- Modified Trapezoid



Cycloid - The curve traced by a point on the circumference of a circle that rolls on a straight line

Print Output

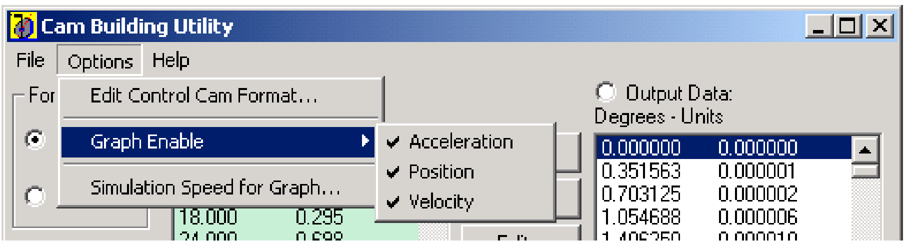
This menu selection prints graph and output data.

Exit

This menu selection closes the CAM Building Utility.

Options Menu

This *Options* menu is used for configuring control CAM format, graphing, and simulating speed for the graph.



CAM_Builder_Utility.tif

Fig. 8-54: CAM Building Utility, Options Menu

Edit Control CAM Format…

This option specifies the format (degree or percent) used by the control when running control CAMs. In offline mode, this setting is saved to the project and is written to control parameter C-0-3141 when the project is synchronized. In online mode, this setting is written to C-0-3141 in the control and project.

Note: When running control CAMs, the format of the active control CAM must match the format in C-0-3141 to ensure proper CAM profiles.

Graph Enable

This selection allows the user to choose between acceleration, position, and velocity graphs.

- *Acceleration Graph* - when checked, the acceleration graph is visible on the graph and printout.
- *Position Graph* - when checked, the position graph is visible on the graph and printout.
- *Velocity Graph* - when checked, the velocity graph is visible on the graph and printout.

Simulation Speed for Graph

This selection allows the user to specify a speed used in graphing the velocity and acceleration profiles.

Help Menu

The Help menu is used for accessing help system and identifying the product.

Rexroth
Bosch Group

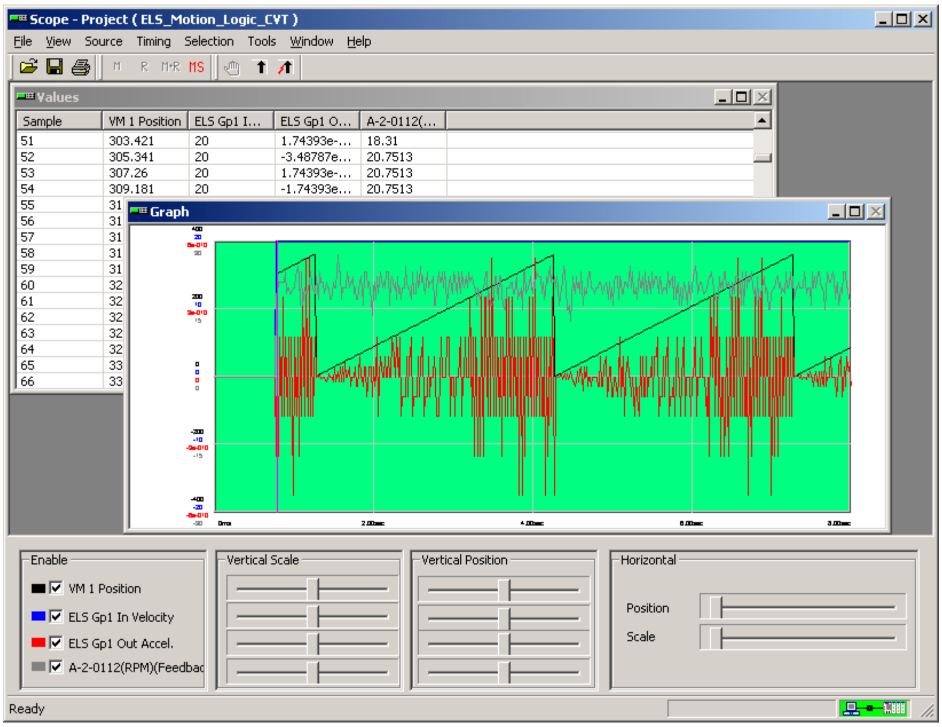
9 Diagnostic Tools

9.1 Oscilloscope

VisualMotion's oscilloscope function is used to capture predefined internal and external signals from the control or connected drive(s). Control signals, that are valid oscilloscope signals, are parameterized and stored in the control. Drive signals are parameterized and stored in the drive, and are dependent on whether or not the connected drive supports the oscilloscope function.

Feature	Control	IndraDrive	EcoDrive 03 or Diax 04	Description
Configurable Signals	4	4	2	N/A
Sample Rate	2ms, 4ms, 8ms, 16ms, 32ms, and 64ms	500µs, 1ms, 5ms, 10ms, 50ms, 100ms	250µs, 500µs, 1ms, 5ms, 10ms, 50ms	time interval for signal capture
Sample Count	100, 200, 500, 1000, 2000, and 4000	100, 500, 1000, 2000, 400, and 8000	50, 100, 200, 300, 400, and 500	number of captures for entire trace

Table 9-1: Oscilloscope Features



oscillo_utility.tif

Fig. 9-1: Oscilloscope Utility

After a trace is capture, the user can view a graphical representation of the data or the actual values for all the uploaded samples.

Oscilloscope Signals

Up to 4 different oscilloscope control signals can be parameterized. Three parameters are required to identify each oscilloscope signal. The following table lists the parameters for each signal:

Signal Number	Parameters
Signal 1	C-0-2501, C-0-2504, and C-0-2507
Signal 2	C-0-2502, C-0-2505, and C-0-2508
Signal 3	C-0-2503, C-0-2506, and C-0-2509
Signal 4	C-0-2524, C-0-2525, and C-0-2526

Table 9-2: Oscilloscope Signal Parameters

Drive signals, that are valid oscilloscope signals, are parameterized and stored in the drive. Refer to the relevant *Rexroth Digital Drive* firmware *Functional Description* manual for oscilloscope parameters.

Timing and Pretrigger Options

The capture duration, of configured oscilloscope signals, is calculated by multiplying the following two control parameters:

Parameter	Description
C-0-2510	Oscilloscope Sample Rate
C-0-2514	Oscilloscope Sample Count

Table 9-3: Oscilloscope Timing Parameters

For example, (C-0-2510 = 8ms) * (C-0-2514 = 500)] = 4000 ms

Pretrigger

A pretigger allows the capture of data for all configured signals before the trigger point. The pretrigger value is stored in control parameter **C-0-2515 Oscilloscope Trigger Post-Count**.

Enable Trigger and Upload

The recording of signal traces begins when the oscilloscope trigger is enabled. Triggers can be configured as either user or internally initiated.

Selecting ***Diagnostics ⇒ Oscilloscope*** opens the *Oscilloscope* window. The oscilloscope utility is used to capture and display run-time data when in online mode. In offline mode, saved captures can be opened and viewed. The capture can be of the control or on a drive that supports this feature. Selected data is acquired on the drive or control, passed to VisualMotion Toolkit, and displayed on the graphical format. The graphical display and supporting data can be printed, or the data can be saved to a file for later review.

File Menu

The File menu is used for retrieving file data, saving data to a file, printing, and exiting the oscilloscope utility.

Open - data from user selected input file is loaded into input data list-box.

Save - data from user selected output is loaded into output data list-box.

Print Output - the oscilloscope graph and its related data table is sent to the printer.

Exit - terminates this utility

Source Menu

When using VisualMotion Toolkit's oscilloscope utility, only those drives that support the oscilloscope function will be available as a signal source.

Selects the source from which the oscilloscope will gather signal data.

Drive 1 to n - lists of all drives on the SERCOS ring that supports the oscilloscope feature.

Control - when selected, the oscilloscope can then read control variables, parameters and registers.

Timing

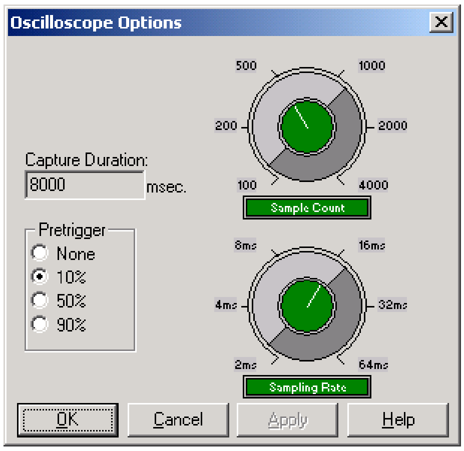
The Oscilloscope timing options in Fig. 9-2 are used for setting the **Sampling Rate** (How often a trace is captured) and **Sample Count** (How many sampling rates are captured). The **Capture Duration** field displays the total capture duration that is calculated by multiplying the Sample Count and Sampling Rate. A **Pretrigger** can be added and is a percentage of the capture interval.

The Sample Count can be set to a maximum of 8000 for IndraDrive, 4000 for control signals, and 500 for drive signals.

The Sampling Rate can be set to a maximum of 100ms for IndraDrive, 64ms for control signals, and 50ms for drive signals.

For example, a maximum count and rate setting for a control signal would generate a total capture duration of 256,000 ms (4 min – 16 seconds).

Note: The pretrigger appears on the oscilloscope screen as a vertical line.



oscillo_timing.tif

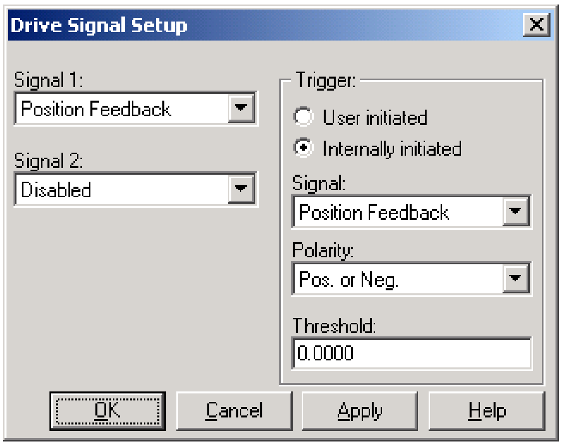
Fig. 9-2: Oscilloscope Options

Signal Selection

The Drive Signal Setup in Fig. 9-3 is available when a drive is selected under the Source menu.

Signal Selection for EcoDrive 03 and Diax 04

The *Drive Signal Setup* window below appears as follows for EcoDrive 03 and Diax 04 drives.



oscillo_drive_signal.tif

Fig. 9-3: Drive Signal Selection

Two drive signals can be captured and viewed. The following is a listing of the available drive signals.

- Position Feedback
- Velocity Feedback
- Velocity Deviation (from commanded value)
- Position Deviation (from commanded value)
- Torque Command Value (required to maintain the commanded Velocity/Position)
- Disabled (Signal 2 only)

Signal Selection for IndraDrive

The *Drive Signal Setup* window below appears as follows for IndraDrive drives.

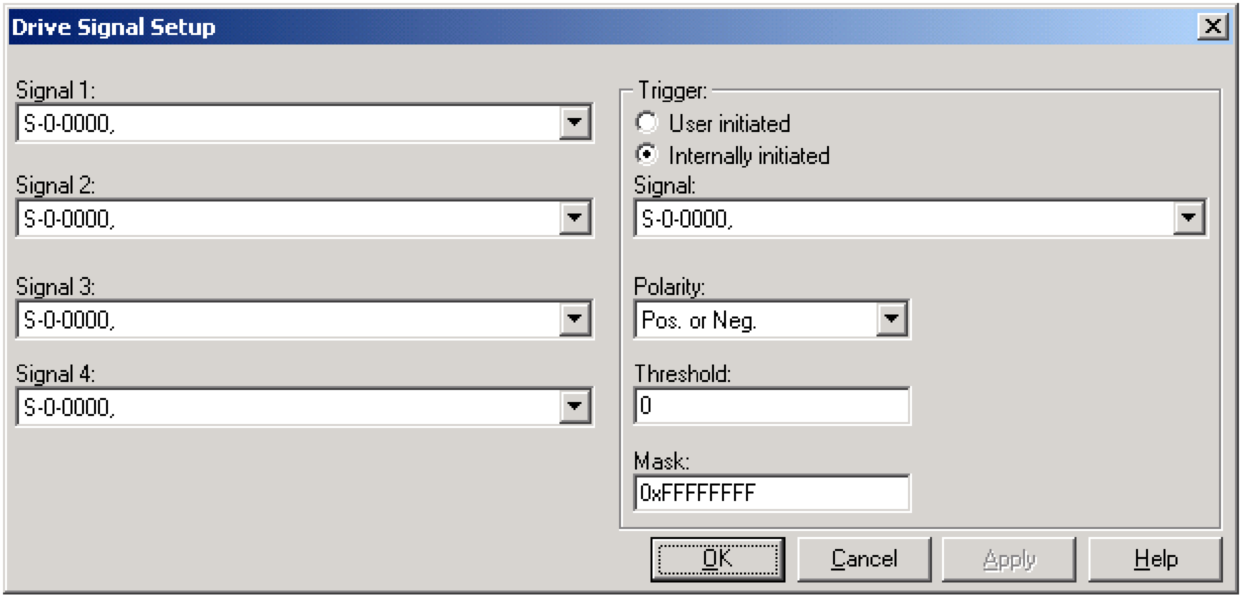


Fig. 9-4: IndraDrive Signal Selection

Up to four drive signals can be captured and viewed. The available parameters are listed in Sercos drive parameter P-0-0149.

Signal Selection for the Control

The Control Signal Setup in Fig. 9-5 is available when a Control is selected under the Source menu. Up to 4 different control signals can be selected and compared.

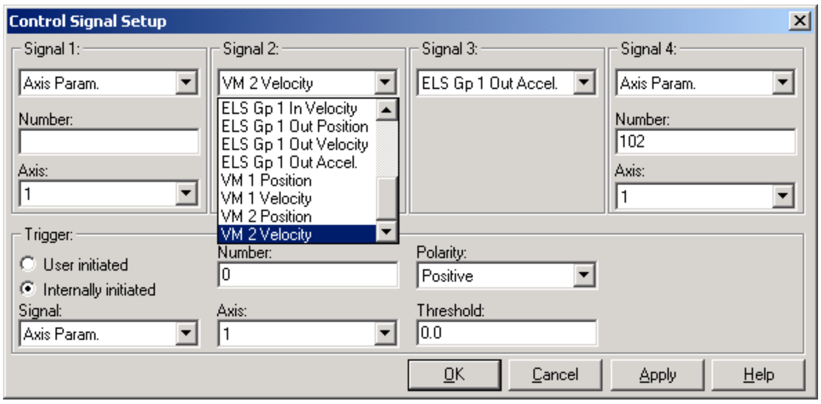


Fig. 9-5: Control Signal Selection

- Program Floats (Fx)
- Program Integer (Ix)
- Global Floats (GFx)
- Global Integers (GIx)
- Axis Parameters * of drives on SERCOS ring NOTE
- Register Bit
- +/- Register (could be used to monitor a register's value)
- Card Param NOTE

- Task Param [NOTE]
- ELS Gp # In Position.**
- ELS Gp # In Velocity.**
- ELS Gp # Out Position.**
- ELS Gp # Out Velocity.**
- ELS Gp # Out Acceleration.**
- VM1 Position (Virtual master signal)
- VM1 Velocity
- VM2 Position
- VM2 Velocity

Note: Only a selective list of axis, control and task parameters can be used as signal types. Refer to control parameter C-0-2504.

---

*Axis parameter must be in cyclic telegram. Use parameter A-0-0185 and A-0-0195 to add other drive parameters to cyclic data.

**The # symbol represents ELS Groups 1-8. This same signal is available for each ELS Group in the system.

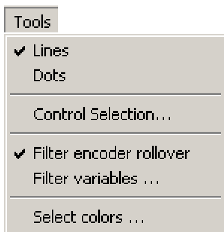
For either signal source, the sample acquisition may be **User initiated** or **Internally initiated**.

For **User initiated** captures, data acquisition starts as soon as the capture button  is pressed. This type of start capture is not deterministic. All other fields in the trigger section are grayed out.

For **Internally initiated** captures, the available signals are the same as the signals for Signals 1 – 4. The heading in the trigger fields will change based on the signal selected. The trigger polarity options are on positive edge, negative edge, or both. Signal threshold is the signal level to trigger.

Tools Menu

From the tools menu, a trace's appearance can be changed from *Lines* to *Dots*. When combined with the Time Controls feature on page 9-10, the user can scale (zoom) in to reveal the individual dots that makeup the trace.

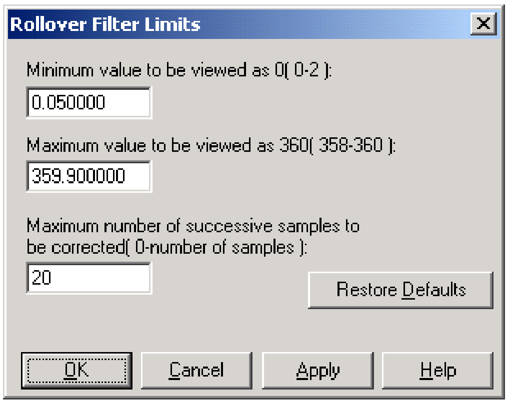


oscillo_option_menu.tif

Fig. 9-6: Options Menu

Filter encoder rollover

The position control loop in servo systems is continuously correcting the position of an encoder when at standstill. This continuous correction in position can cause dithering that will be captured by the oscilloscope. When selected, the *Filter encoder rollover* will eliminate any dithering based on the settings of the *Filter variables* … window in Fig. 9-7.



oscillo_rollover.tif

Fig. 9-7: Filter variables

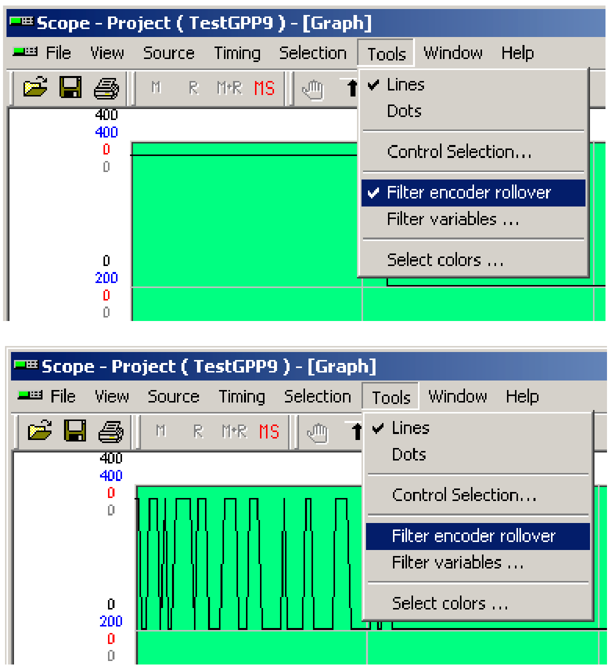
Minimum value to be viewed as 0

A value between 0 and 2 degrees will be used as the filter window for ignoring position dithering. While holding position at 360°, any value between 0 and the minimum value entered will be interpreted as a dither and seen as 360 degrees up to a maximum number of successive samples.

Maximum value to be viewed as 360

A value between 358 and 360 degrees will be used as the filter window for ignoring position dithering. While holding position at 0°, any value between 360 and the maximum value entered will be interpreted as a dither and seen as 0 degrees up to a maximum number of successive samples.

The oscilloscope will immediately capture any position value outside the minimum or maximum filter windows. An example of the filter encoder rollover is shown in Fig. 9-8.



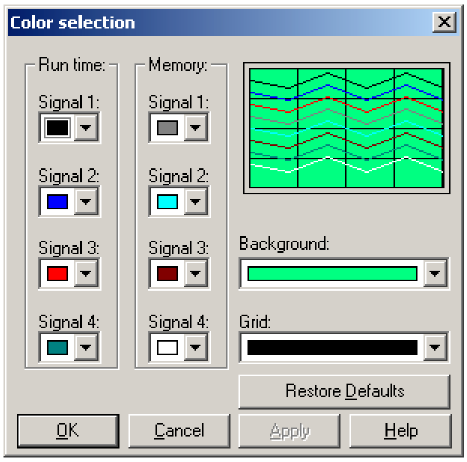
oscillo_filter_encoder.tif

Fig. 9-8: Filter encoder rollover

Rexroth
Bosch Group

Select colors…

The four signal traces are color coded and can be changed for both run time and memory by selecting a color for each trace in the Signal color selection window in Fig. 9-9. In addition, the background and grid colors can also be changed.

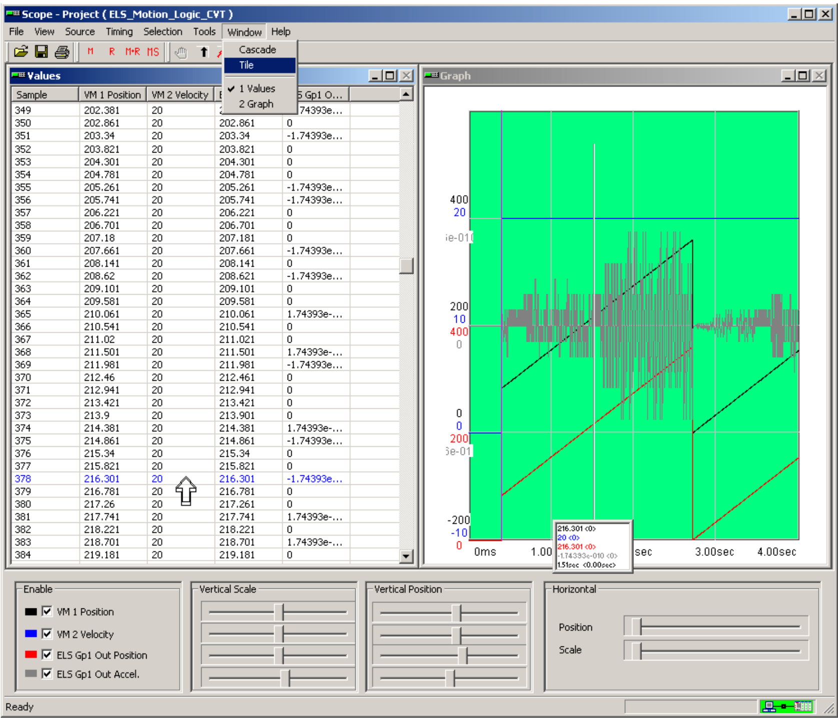


oscillo_color_select.tif

Fig. 9-9: Signal color selection

Window Menu

The Window menu allows the user to cascade or tile the **Values** and **Graph** windows or change the focus between them.



oscillo_window_view.tif

Fig. 9-10: Viewing Different Windows

Selecting a Sample in the Values Table

When a capture sample is selected in the *Values* table, the values are also displayed on the *Graph* window when holding down the left mouse button.

Abort, Upload and Enable Trigger

The *Abort*, *Upload* and *Enable Trigger* buttons are used to trigger trace captures of configured signals. The *Upload* and *Enable Trigger* functions are also available under the Options menu.

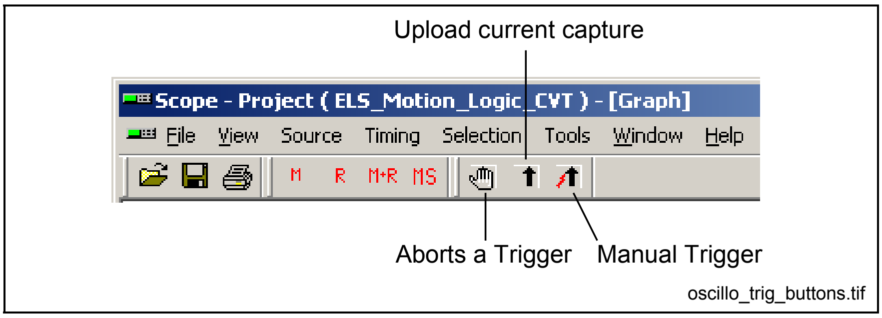


Fig. 9-11: Signal triggering buttons

Oscilloscope Memory Buttons

Using memory buttons, traces can be stored into memory for viewing and comparing.

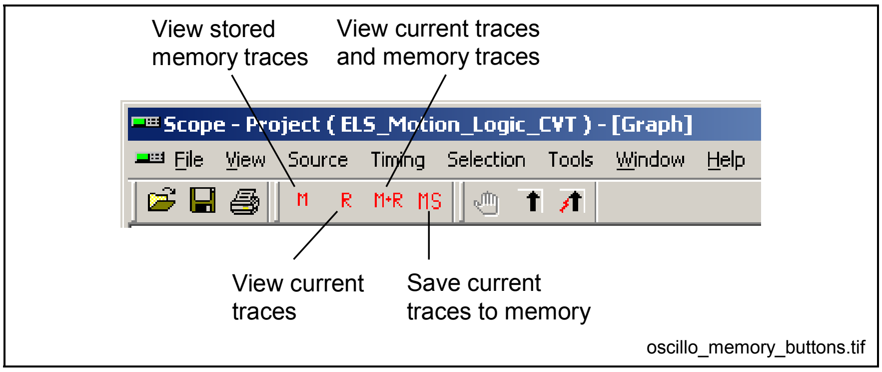


Fig. 9-12: Oscilloscope memory buttons

Manipulating Trace Signals

When multiple traces are captured at one time, they can be positioned and scaled independently of each other by sliding the cursor left or right for each trace as shown in Fig. 9-13. The traces can also be turned on or off by clicking on the check boxes to the right of the signal description.

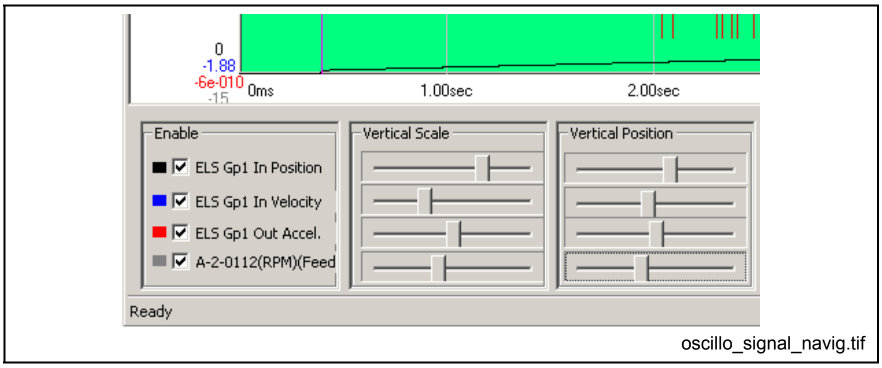
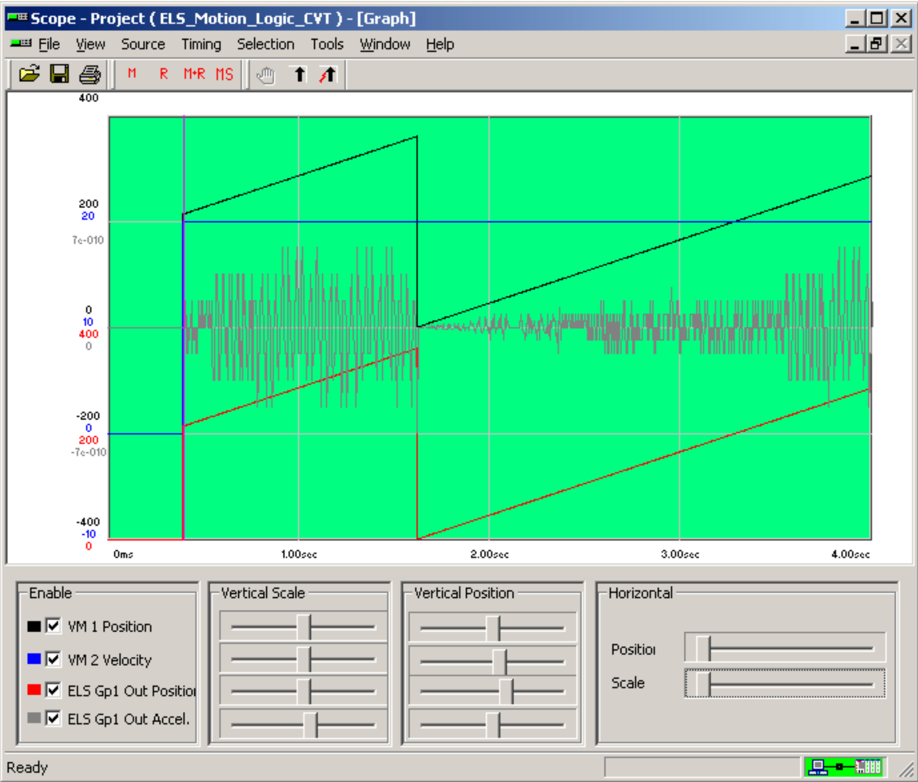


Fig. 9-13: Manipulating trace signal

By positioning the traces above and below each other, the user can more easily distinguish between the signal. Fig. 9-14 shows an example of three traces repositioned for clarity.



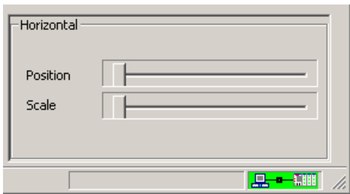
oscillo_signal_move.tif

Fig. 9-14: Separated oscilloscope traces

Time Controls

The position and scale functions in Fig. 9-15 are used to analyze a specific area of a captured trace. The scale function acts as a zoom, allowing the user to view smaller sections of a trace, while the position function controls the horizontal scrolling.

Note: The position function only works after using scale.

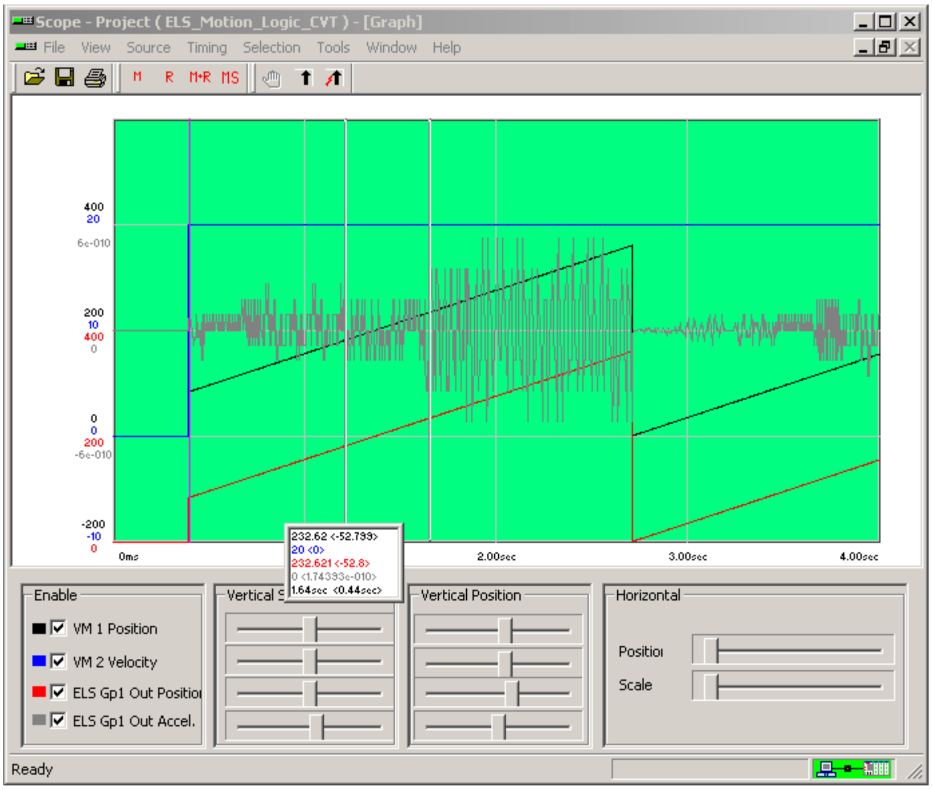


oscillo_time_control.tif

Fig. 9-15: Time Controls

Measuring Trace Signals

Oscilloscope trace signals are measured by placing the cursor at a start position on the graph. When the mouse is clicked and held, a vertical line and small measurement window appear initiating the start of a measurement. As the cursor is moved (while held down), a second vertical line appears.



oscillo_measure.tif

Fig. 9-16: Measuring Signal Traces

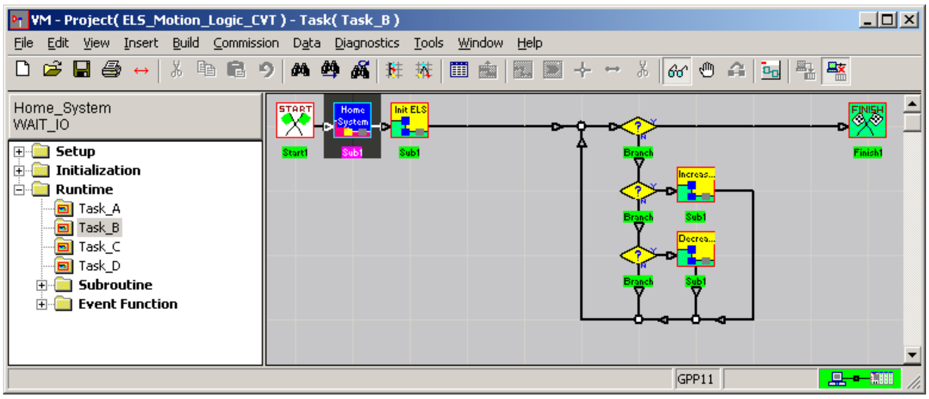
The small measurement window simultaneously measures both the current position measurement and a difference measurement from start (in brackets < >) for all 4 signal traces. The first 4 measurements represent the 4 trace signals. The last measurement represents the time line measurement. When the cursor is held and moved, left or right, the measurement is increased or decreased based on direction.

9.2 Show Program Flow

Selecting ***Diagnostics ⇒ Show Program Flow F7*** highlights the currently executing icon in a running program, as illustrated in Fig. 9-17. Many icons are scanned quickly and may appear to be skipped.

During *Show Program Flow*, a check mark appears next to the menu item and other menu items are grayed-out. Re-selecting *Show Program Flow* removes the check mark and re-enables the other menu items.

Note: This menu selection is only available for online mode. Show program flow is not available for *Event Functions*.



Program_Flow.tif

Fig. 9-17: Show Program Flow View

Showing Program Flow in Subroutines

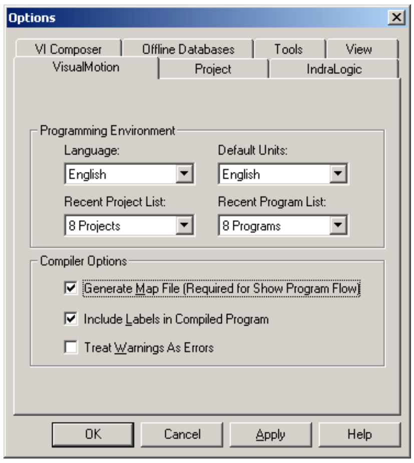
To view the program flow for a subroutine used in multiple tasks, the user must identify the task controlling the subroutine. If a subroutine is opened by entering the function (through the icon) from the task level, VisualMotion knows the task – subroutine association. However, if the subroutine is opened by selecting *View ⇒ Subroutine* or selecting it from the *Project Navigator*, a window opens asking the user for task association.

Note: Although VisualMotion supports the programming of up to 40 nested subroutines, the highlighting of icons in nested subroutines, when using *Show Program Flow*, is limited to 10 levels.

Generate Map File

Show program flow uses a map file generated at compile time to tag the screen location of an instruction. If this map file is not found, the error message "Cannot open file \\...*.map!" appears (this usually means that the file was not compiled or downloaded to the control).

The compiler option for generating a map file is found under the menu selection, ***Tools ⇒ Options***. By default, the *Generate Map File* option is selected. For larger programs that do not require a map file, removing the options decreases compile time.

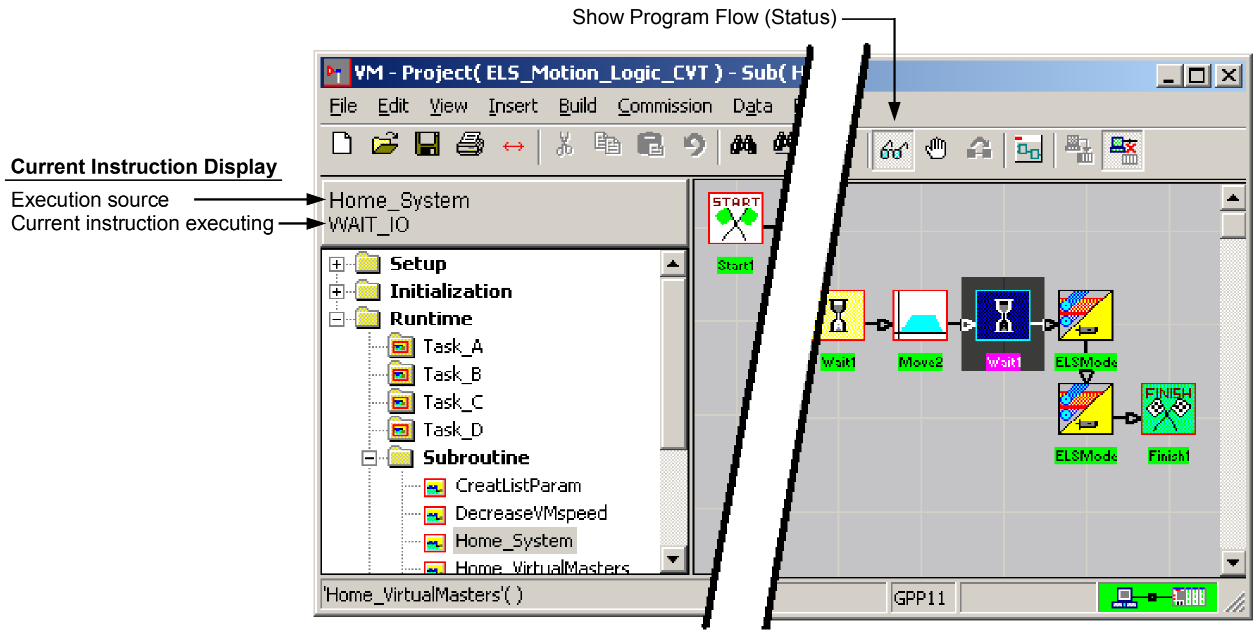


VisualMotionOptions.tif

Fig. 9-18: Generate Map File

Current Instruction Display

Selecting the ***Show Program Flow (Status)*** toolbar button, switches VisualMotion Toolkit's programming environment to status mode. While in status mode, program instructions currently executing are displayed in the top portion of the *Project Navigator* window. The icon palettes are not displayed while *Show Program Flow* is active.



CurrentInstDisplay.tif

Fig. 9-19: Current Instruction Display

The current instruction is displayed in the *Project Navigator* whether or not a program is running. Once the control is initialized, the *Initialization* task runs to completion and program flow stops on the *Start* icon of each task (A-D), if programmed. This occurs before program execution is initialized by the user. Once the program is running, the location and current instruction execution displays change based on the current instruction (icon). The name displayed for the current instruction might not always match the expected name of the icon. The name displayed is that of the compiled instruction for that icon.

Rexroth Bosch Group

Displaying Task Function Arguments and Local Variables

Function arguments can be defined in the ***Start*** icon of any subroutine (initialization and standard). Local variables can be defined in the ***Start*** icon of any task, subroutine or event function. When a task, subroutine or event function is executing and the ***Show Program Flow*** toolbar button is set, any function argument and/or local variable are displayed below the *Current Instruction Display* section. The values of function arguments and local variables may change quickly while a program is running. However, the function arguments and/or local variables displayed can be helpful when using a breakpoint in the desired task or subroutine. The user can monitor the values for function arguments and/or local variables by single stepping through the task or subroutine. Refer to *Set Breakpoint using VisualMotion Toolbar Buttons* below for details.

Note: When no values are passed to the function arguments and/or local variables, the values display question marks.

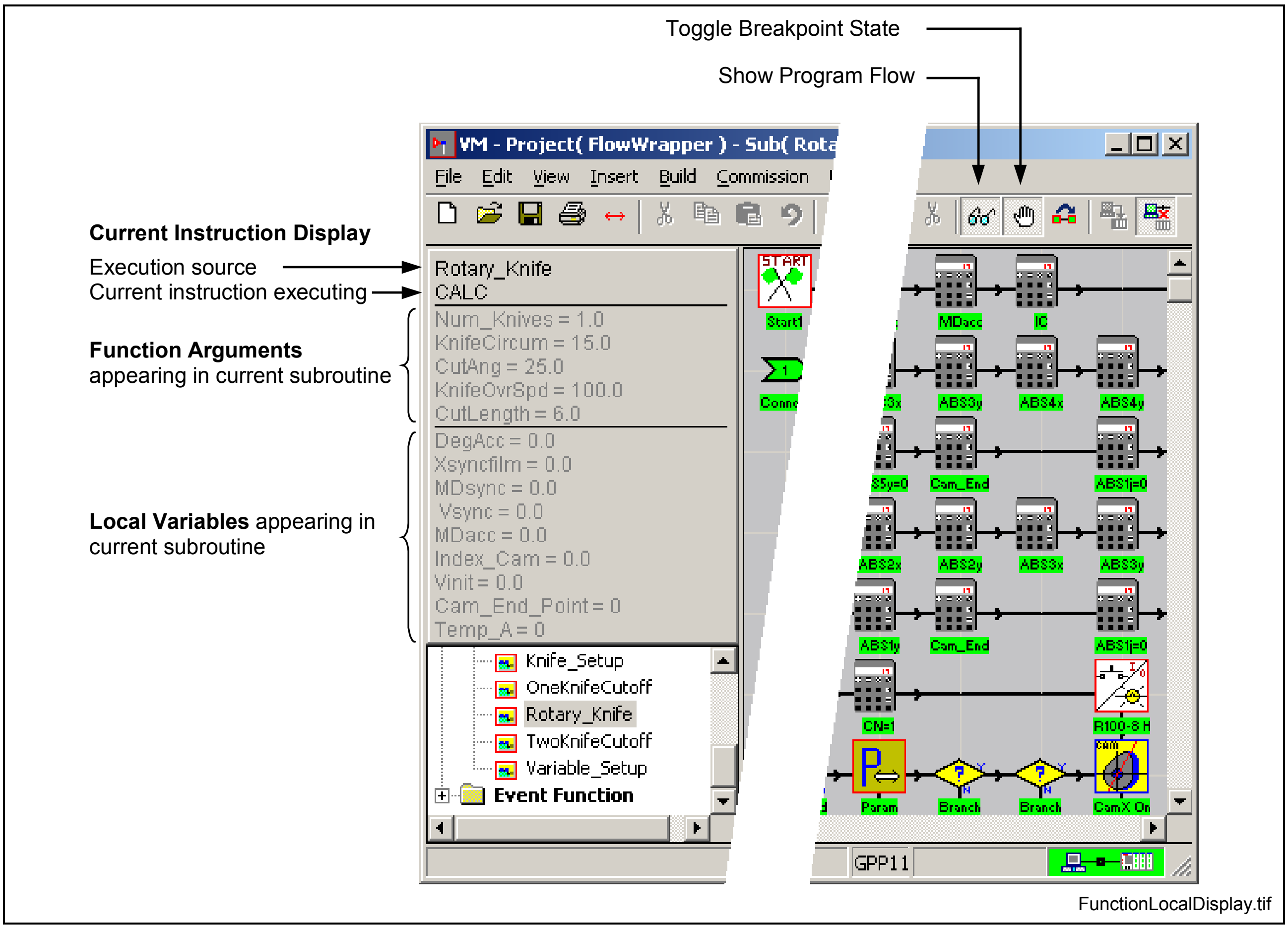


Fig. 9-20: Displaying Function Argument and Local Variables

9.3 Breakpoint

Set Breakpoint using VisualMotion Toolbar Buttons

Use the following steps to set a breakpoint using the VisualMotion ***Toggle Breakpoint State*** and ***Single Step*** toolbar buttons.

1. Select a task (A-D) or standard subroutine.
2. Click the ***Show Program Flow*** toolbar button and then click on the ***Toggle Breakpoint State*** button.

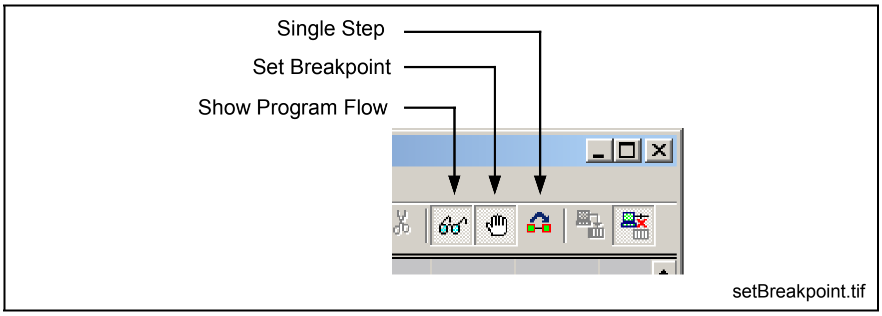
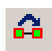


Fig. 9-21: Set Breakpoint

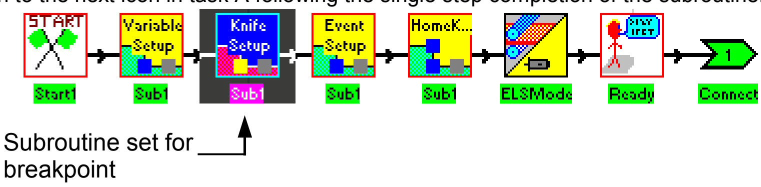
3. Start the VisualMotion program with the task or subroutine displayed.

When the breakpoint is reached, the control switches to single step mode, program flow stops at the first icon following the *Start* icon and the *Current Instruction Display* section above the Project Navigator turns red.

Note: Refer to Current Instruction Display on page 9-13 for details.

4. Press the F8 key or click the Single Step  button to step through the program icons. If the task or subroutine contains a subroutine icon, all icons contained within the subroutine will be single stepped before proceeding to the next icon.
5. Once all program icons and/or subroutines are single stepped, program flow returns to the subroutine or task that called the subroutine or task.

For example, if a subroutine (contained in task A) is set for breakpoint, program flow will return to the next icon in task A following the single step completion of the subroutine.



Note: If a different item (in the Project Navigator) is selected, the single step function becomes inactive. The control remains in single step mode and requires that the user toggle the Cycle_Start_Resume (bit 6) of the task control register (2-5) for the selected task to resume program flow.

For best results, enable a breakpoint before starting the program or before the task or subroutine program flow is started. Once the Start icon in a task or subroutine is processed, a breakpoint can not be set.

9.4 System Diagnostics

System

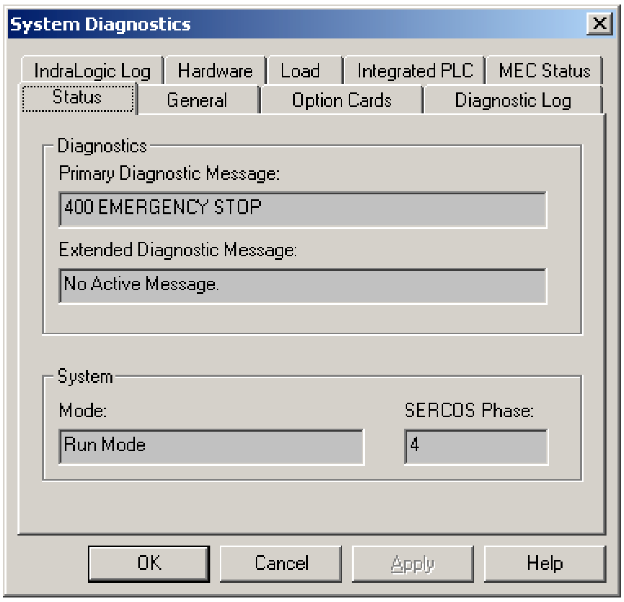
Selecting ***Diagnostics*** ⇒ ***System*** opens the *System Diagnostics* window. This window displays diagnostic information about the active VisualMotion system. The window uses tabs to display diagnostic information for the following:

- Status
- General
- Option Cards
- Diagnostic Log
- IndraLogic Log
- Hardware
- Load (System)
- Integrated PLC
- MEC Status

Note: Holding the cursor over any field will display the corresponding parameter number.

Status Diagnostics

The *Status* tab displays the current diagnostic message with an extended message, if available. The system's current mode and Sercos phase is also displayed.

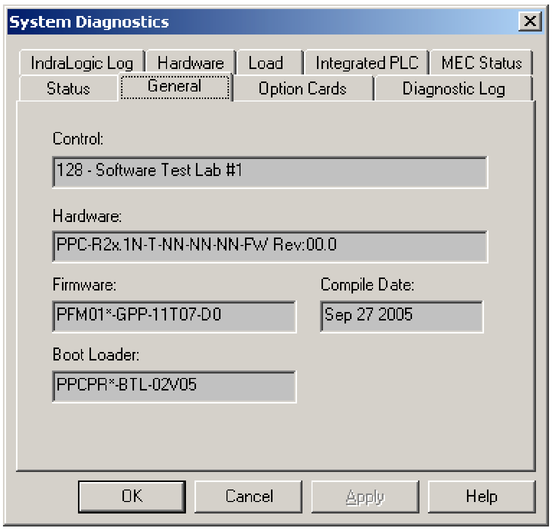


DiagStatus.tif

Fig. 9-22: Status Diagnostics

General Diagnostics

The *General* tab displays the control's address and label, hardware typecode, firmware version, firmware compile date and boot loader version.

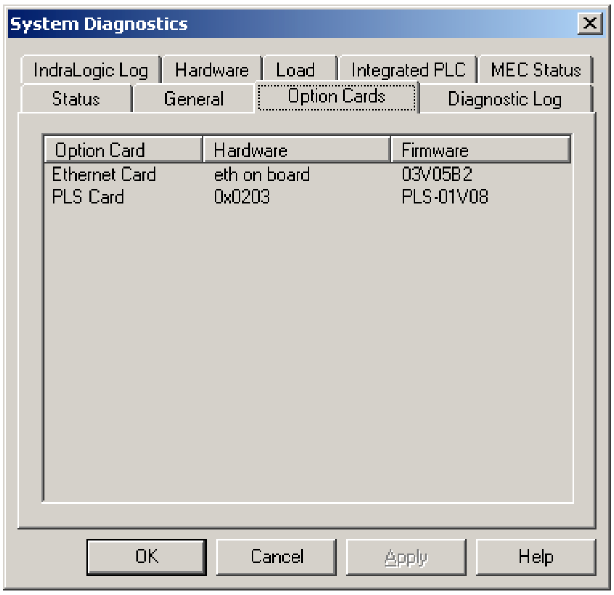


DiagnosticGeneral.tif

Fig. 9-23: General Diagnostics

Option Cards Diagnostics

The Option Card tab displays the current hardware card(s) installed in the control. Refer to the *VisualMotion 11 Project Planning* manual for a complete listing of available hardware configurations.

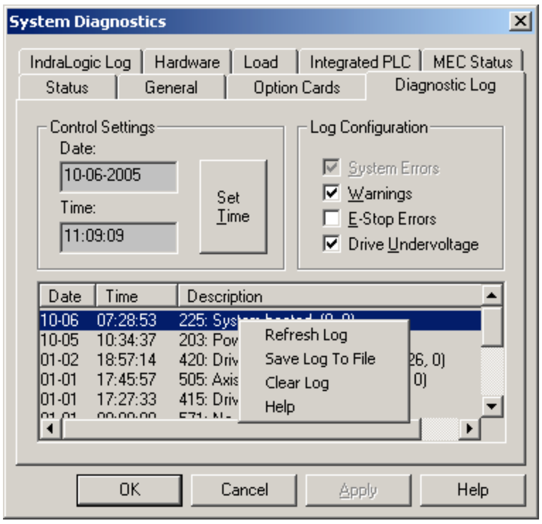


DiagOptionCards.tif

Fig. 9-24: Option Cards Diagnostics

Diagnostic Log

The Diagnostic Log tab displays the last 100 errors the control has encountered. Along with the error messages, the date, time and extended error codes are displayed.



Diagnostic_Log.tif

Fig. 9-25: Diagnostic Log

Control Settings

The date and time are relative to the power on of the control. The time can be set to the computer’s clock by clicking the ***Set Time*** button. Once set, the time and date will stay current as long as the control has power. On power down, an internal capacitor maintains the real-time clock running for up to 3 days.

Log Configuration

The user can select which errors are to recorded and displayed by checking the appropriate checkbox.

Log Data

The last 100 control errors are displayed and can be sorted by Date, Time or Description by clicking the appropriate heading.

An option pop-up window is displayed by right clicking the mouse. From here, the user can refresh the current display, save the log to a file, clear the log or request help on the selected diagnostic.

Note: Help on any displayed diagnostic error can be selected from the pop-up window or by double clicking on the desired error in the list.

IndraLogic Log

The IndraLogic Log displays any PLC-related error generated from function blocks, functions, communication, etc. Error functionality created by the user using the ErrorLog library will also be displayed. Checking the *I/O Image Update Errors* checkbox will include data update errors that may occur between the motion and logic portion of VisualMotion.

The **Clear Function Block Error Log** button is used to clear the list of errors displayed in the IndraLogic Log.

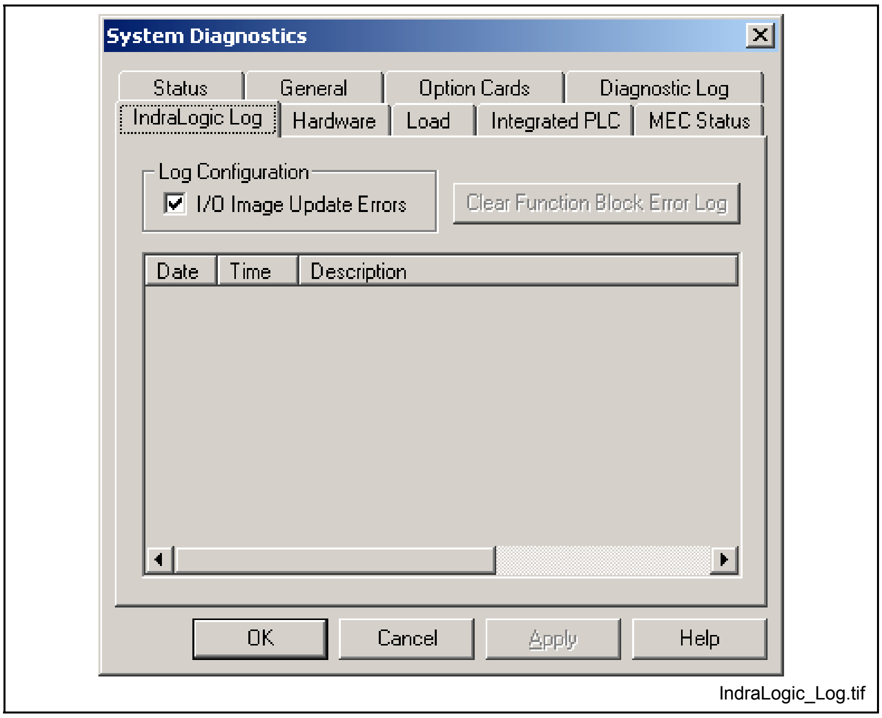
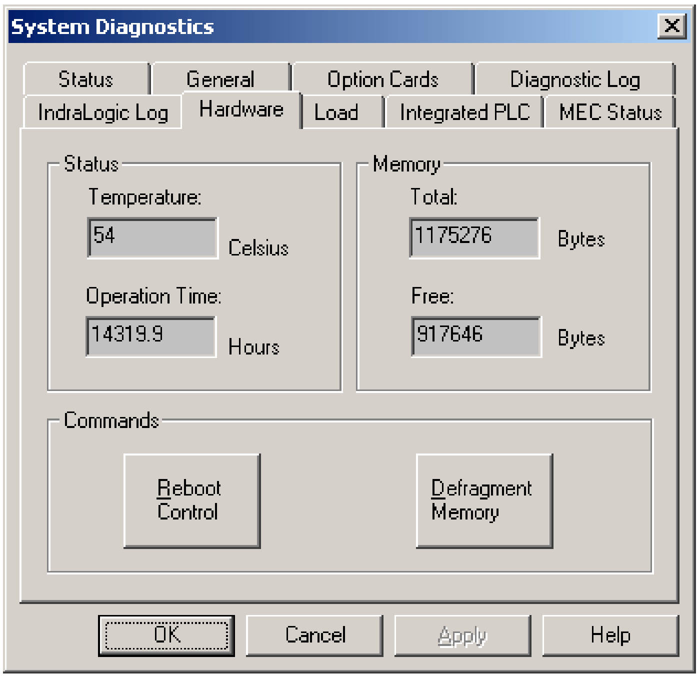


Fig. 9-26: IndraLogic Log

Hardware Diagnostics

The *Hardware* tab displays control related information, such as current operating temperature and control's total operating time. The memory section displays the control's total memory and available free memory.



DiagHardware.tif

Fig. 9-27: Hardware Diagnostics

Reboot Control

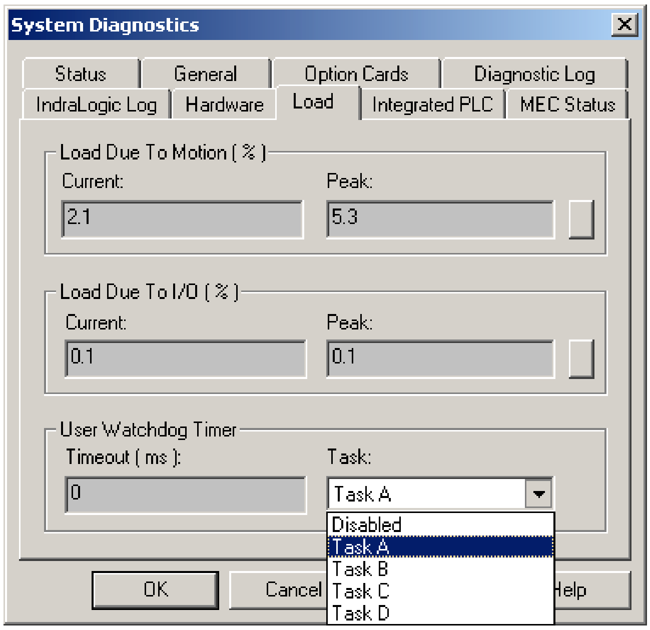
Clicking the ***Reboot Control*** button enables the *Software Reset for Control* command of control parameter C-0-0993. The user is warning before the control is rebooted. The control must be in parameter mode in order to reboot the control using this button.

Defragment Memory

The defragment button is used to request a defragmentation of the control's memory. The determination of a necessary defragmentation is performed by the control and not by clicking on the button. Clicking the button queues the control.

Load Diagnostics

The *Load* tab displays the control's current and peak time required by the processor to process motion task and I/O completion as a percentage of parameters C-0-0099 and C-0-3001, respectively. The User Watchdog Timer section is used to specify a time (in ms) for a selected task to complete its program before an error is issued.

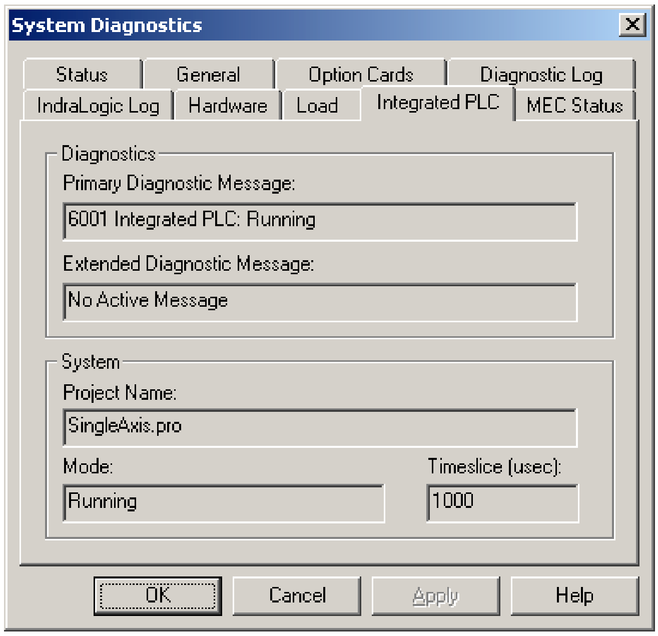


DiagLoad.tif

Fig. 9-28: Load Diagnostic

Integrated PLC Diagnostics

The *Integrated PLC* tab displays IndraLogic's diagnostic messages, the PLC program's name, mode and Sercos Timeslice.

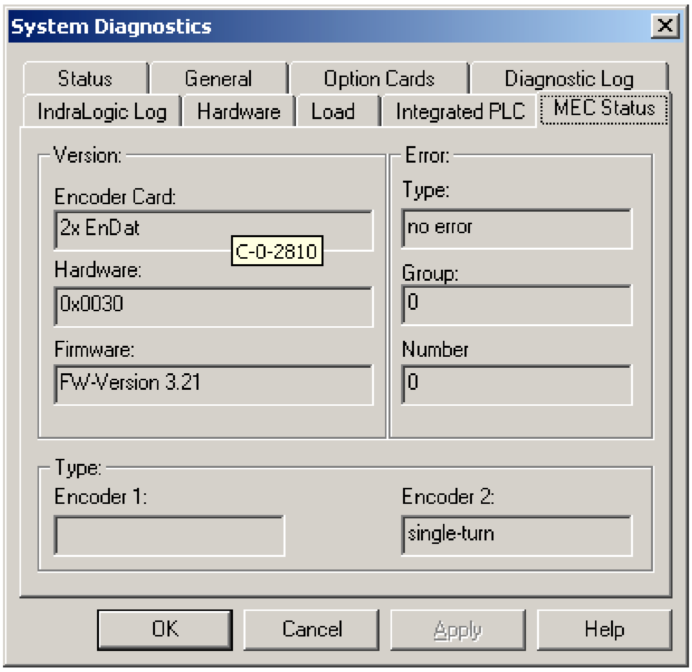


Diagnostic_IndraLogic.tif

Fig. 9-29: Integrated PLC Diagnostics

MEC Diagnostics

The MEC tab displays diagnostics for the Master Encoder Card (MEC). Hardware version and firmware is displayed along error information and connect encoder types.



Diagnostic_MEC.tif

Fig. 9-30: MEC Diagnostics

10 Project Management Tools

10.1 Synchronizing a Project

Synchronization involves a process where specific checks and validations are preformed before an offline project can be compiled and downloaded to the control.

This process involves the following steps:

1. Communication to the control is checked.
2. Project firmware target is validated with control firmware target.
3. Drives setup in project are validated with detected drives on Sercos ring.
4. Project data is validated before the program is compiled and downloaded to the control. This includes the validation of global variables, creation of icon map file, the assignment of a unique identifier for the current project status, etc.

The synchronization process is initiated by clicking the **Toggle Online/Offline** icon toolbar button, selecting ***File*** ⇒ ***Online***, or by pressing the **F9** key.

Communication Errors

When a communication error occurs between a project and the control, the user is notified with an error message. Typically communication problems occur when an error exists in the Control Selection settings. Verify and check the communication setting and retry.

Detecting Drive Differences

As part of the project validation process, all axes configured under *Setup* are compared with those detected on the Sercos ring. If a difference is detected, the user is notified and is asked to correct the situation or continue with synchronization. Refer to Drive Differences Detected When Synchronizing on page 10-5 for details.

Switch to Online Mode

After all synchronization checks and validations are performed, the *Synchronize Project Data* window opens and presents the user with a list of components that should be compiled and downloaded to the control. Refer to Synchronizing a Modified Project on page 10-2 for details. The following two online synchronization modes are supported:

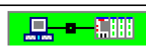
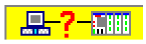
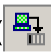
Synchronization Mode	Description	Status Indicator
Synchronized	Communication with the control is active and all components between the project and the control's memory match and are up to date.	
Unsynchronized	Communication with the control is active, however, not all components in the project have been updated on the control.	

Table 10-1: Synchronization Modes

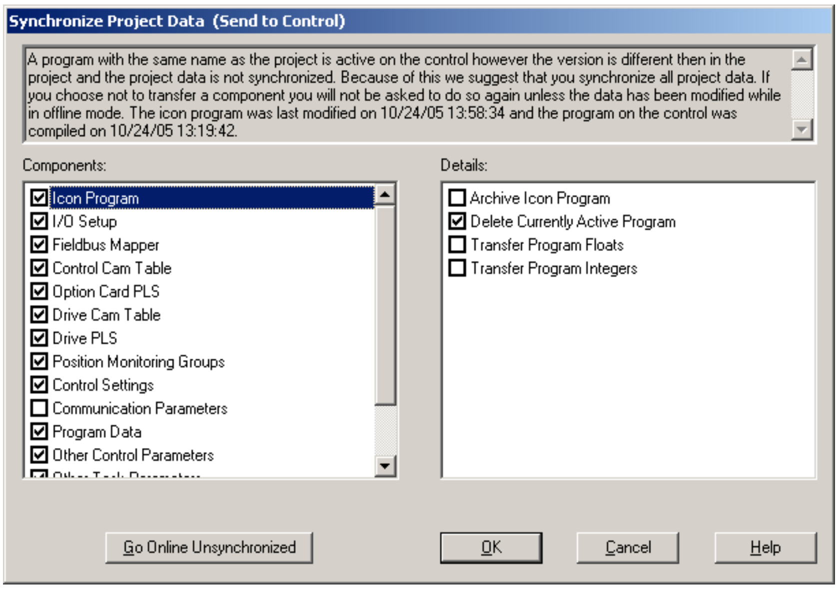
Synchronizing a Modified Project

The synchronization of a modified project to the data on the control is performed using the *Synchronize Project Component* toolbar button (). This button is only available when VisualMotion detects unsynchronized project data. An unsynchronized condition occurs when project data does not match the data on the control.

To synchronize modified project components online, select ***File ⇒ Synchronize Project Component*** or click on the *Synchronize Project Component* toolbar button. The *Synchronize Project Data* window opens displaying project components whose offline data has changed since the last time the project was downloaded to the control.

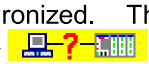
Note: When synchronizing project data, the message area at the top of the window provides the user with a date/time comparison between the icon program being downloaded and the same project on the control.

Note: Communication with any VisualMotion runtime tool such as the Data Editor or the ELS Runtime Tool is suspended while the *Synchronize Project Data* window is opened. Communication is resumed after the synchronization process is completed or if the **Cancel** button is selected.



SychProject_Data.tif

Fig. 10-1: Synchronize Project Data

At this point, the user downloads the modified project components to the control, and if no errors are detected, the project becomes synchronized. If all the components are ***not*** downloaded to the control, the project is considered unsynchronized and the state of the project will remain **un**synchronized. This is indicated by the unsynchronized status bar indicator  at the lower right hand corner.

Phase Dependent Component Download

Some components can be downloaded in Sercos phase 4, while others require the system to be in parameter mode. VisualMotion will notify the user if the control needs to be in parameter mode before a component(s) can be downloaded.

Components and Details

The Component section displays all project components that were detected to be different from those currently in the control's memory for the same project name. Any component that is unchecked is not downloaded to the control.

Note: The first time a new project is downloaded to the control, any component that is not synchronized (unchecked) will not be asked to be synchronized the next time the project is downloaded unless that specific component was modified.

The Details section displays all sub-items that are available for the currently selected component. Any individual detail that is unchecked will render the system unsynchronized.

The following table lists the components that can be selected for download along with a brief description of the detail section:

Component	Description of Detail Section
Icon Program	Options for archived and active program as well as transferring of floats and integers between programs.
I/O Setup	I/O system options and user configuration parameters
Fieldbus Mapper	fieldbus related parameters
Control CAM Table	CAM table parameters for CAMs 1-37
Option Card PLS	Option Card related parameters
Drive CAM Table	Drive CAM related parameters for configured drives
Drive PLS	Drive PLS parameters for configured drives
Position Monitoring Groups	PMG related parameters for 8 configurations
Control Settings	basic control setting parameters and task option parameter for all 4 task
Communication Parameters	By default, this component is not checked. It list the standard communication parameters for serial and Ethernet.
Program Data	user program data such as points, control PLS, event table, Global variables, program variables, and zones
Other Control Parameters	modified control parameters that are not part of one of the functionality components listed above
Other Task Parameters	modified task parameters that are not part of one of the functionality components listed above
Other Drive Parameters	modified drive parameters that are not part of one of the functionality components listed above
Other Axis Parameters	modified axis parameters that are not part of one of the functionality components listed above

Table 10-2: Possible Synchronization Components

Special Options for Icon Program Details

For the Icon Program component, the **Delete Currently Active Program** detail is checked every time a modified icon program is downloaded. Any detail that is unchecked is not downloaded to the control.

The **Archive Icon Program** detail is used to archive the currently opened icon source file (*.str) to the control's memory. This detail is required in order to create a new project from program and data on the control memory.

The **Transfer Program Float / Integer** details are used to transfer program float and integer variables from the currently active program to the project being downloaded. A specific range of program floats and integer can be selected for transfer by selecting ***Tools ⇒ Control Settings ⇒ Data Transfer Ranges***.

Note: Program floats and integers for the selected range will be overwritten with the data on the control that is within the same range.

Modifying Default Project Synchronization Settings

Project synchronization default settings can be modified from the VisualMotion *Options* window under ***Tools ⇒ Options ⇒ Project***.

Go Online Unsynchronized

The **Go Online Unsynchronized** button is used to establish communication with the control without downloading any data.

If VisualMotion detects changes to the data in a project, the user is requested to Save, Compile and Download before switching the online mode. If VisualMotion Toolkit is currently online, the menu selection reads, "*Offline …F9*".

Commissioning Tool Synchronization

Commissioning tools can be used in offline or online mode to modify project data. In offline mode, all modified data is written to the project files. These modifications can be synchronized with the control when switching to online mode.

When a commissioning tool is used to modified data for a project that is online and synchronized, the following functionality will be observed:

Note: Several commissioning tools write to parameters that can not be modified while the system is in Sercos phase 4.

- In Sercos phase 2 (parameter mode), online modifications are saved to both the project and the control. The state of the project remains synchronized if no errors are detected.
- In Sercos phase 4, the following window opens when saving parameters that can only be written to in Sercos phase 2.

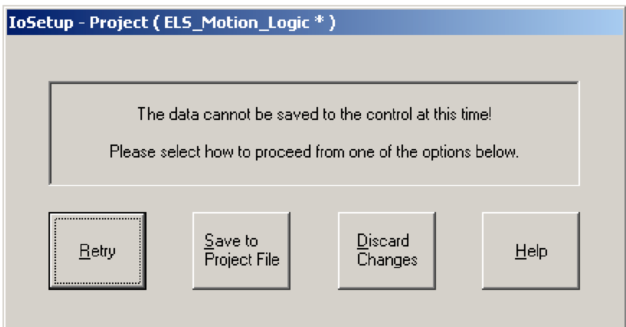


Fig. 10-2: Saving Commissioning Tool Data

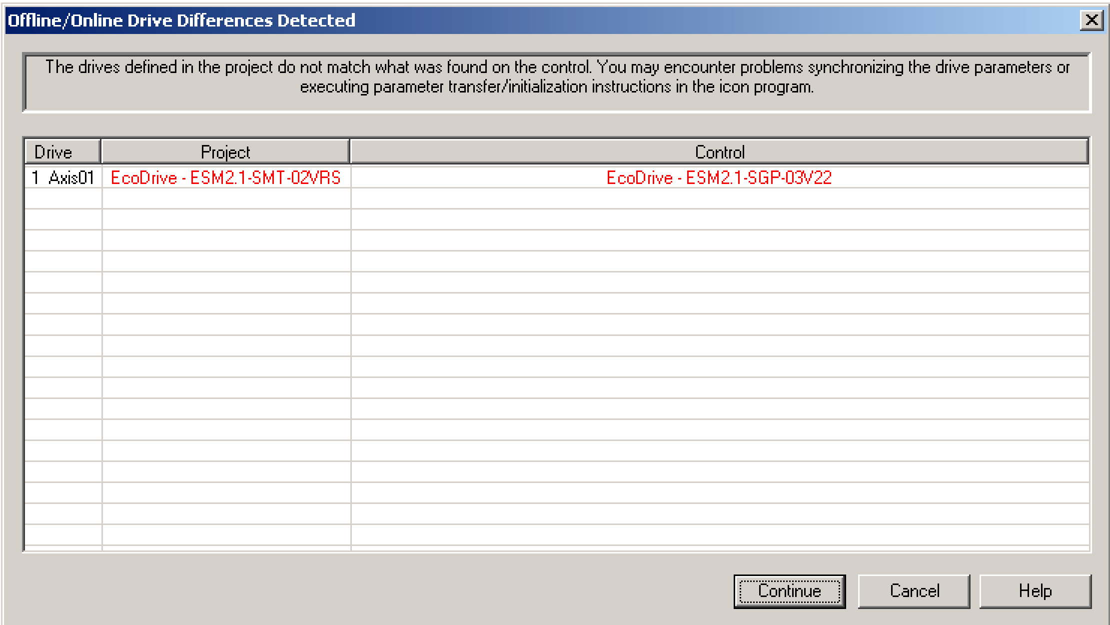
If the system can be transitioned into parameter mode, the **Retry** button can be used to repeat the download process. If the modifications are saved using the **Save to Project File** button, the project becomes unsynchronized with the control. The project must then be synchronized either by going offline and then back online or by selecting the *Synchronize Project Components* toolbar button. Refer to Synchronizing a Modified Project on page 10-2 for details.

Note: After a commissioning tool is modified (online) and saved, the data displayed will be that of the saved file and not the current status of the control. To view the current status of the control, click on the ***Status*** toolbar button ( ). The tool will then reload the display with the current data from the control.

Drive Differences Detected When Synchronizing

By default, this window is launched when a drive hardware or firmware difference is detected between the current project and the control when switching to online mode. The user can chose to **Continue** to synchronize the project with the control. This is only a warning and does not interfere with synchronization. However, problems may be encountered when synchronizing drive parameters or when executing parameter transfer or initialization instructions in the icon program. Make the necessary modifications and re-synchronize the project.

Note: This validation step can be bypassed by removing the check for the *Compare Axis Definitions with Hardware* option under menu selection ***Tools ⇒ Options ⇒ Project***.



DriveDifferenceDetected.tif

Fig. 10-3: Drive Difference Detected When Synchronizing

Rexroth
Bosch Group

10.2 Browsing a Project for Data Usage and Icon Flow

The browse feature in VisualMotion 11 allows the user to locate usage of program labels in a project and usage of subroutines and event functions in the icon flow.

Browse Project for Data Usage

The Data menu selection allows the user to browse the current project for the usage of program labels by type. Selecting ***Browse*** ⇒ ***Data*** or clicking opens the *Browse Project for Data Usage* window in Fig. 10-4. All program tasks, subroutines and event functions are scanned for program data types. In addition, tools such as the I/O Setup, PLS, and Fieldbus editors are opened, scanned and then minimized.

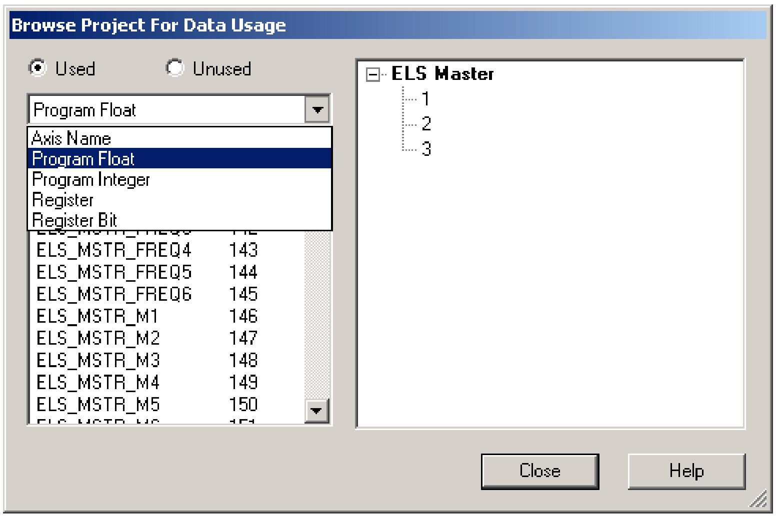
The user can choose to view project data that is **Used** in the project or data that has been declared in the project but **Unused** by selecting the relevant radio button.

The available data types are listed by usage in the table below:

Data Type	**Used**	**Unused**
Axis Name	**X**	
Program Float	**X**	**X**
Program Integer	**X**	**X**
Register	**X**	**X**
Register Bit	**X**	**X**

Table 10-3: Available Data Type by Usage

Note: The window can be resized to allow complete view of the scanned data type. All data types used in the current program are scanned and displayed at the instances the window is opened. Any data type added or modified in the program while the window is opened will not be scanned and added to the list. The window must be close and reopened.



BrowseUsage.tif

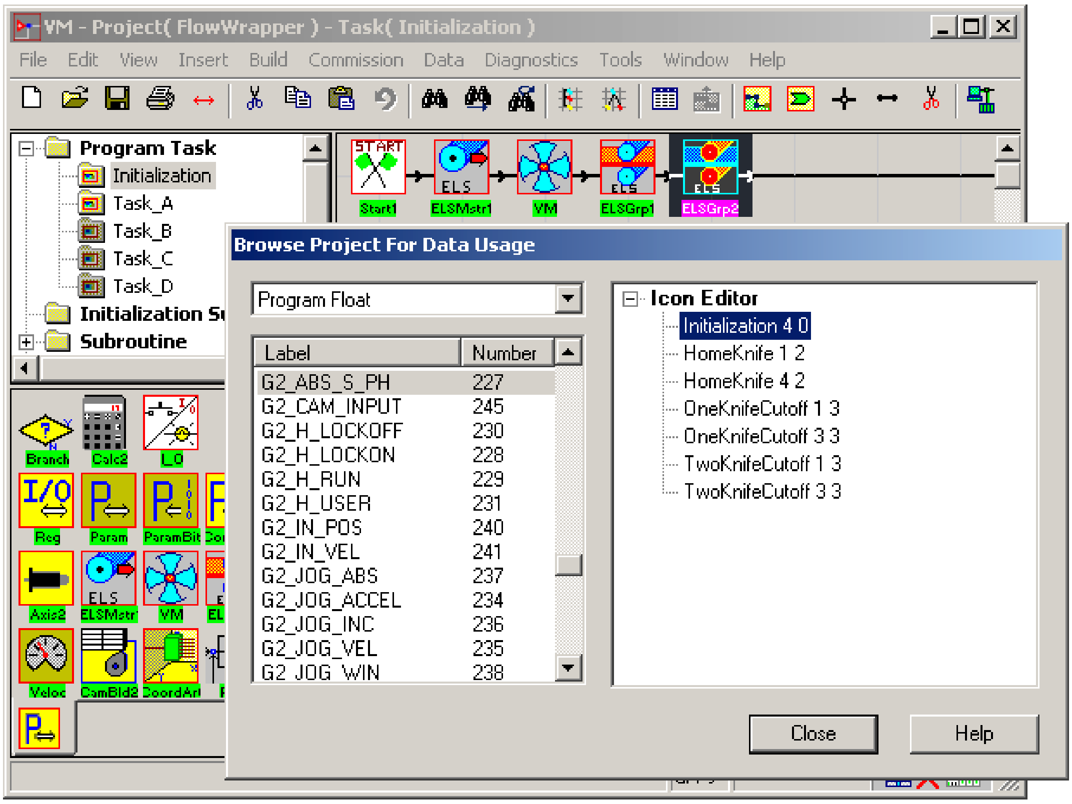
Fig. 10-4: Browse Project for Data Usage

When a data type label is selected from the drop-down list, all assigned labels and numbers used in the program are displayed just below the drop-down list.

When a label is selected from the left-hand list, the editor (Icon, Ladder or Fieldbus) in which the data type is used is displayed in bold text in the right-hand tree structure. Below the editor name, the location or instance of usage, whether in a task or subroutine, is displayed in a tree structure.

Note: Every instance of data usage is displayed in the right-hand section of the window. If a data type is used in more than one area of the current program, duplicate editor names will appear indicating where the data type is used.

When the actual location is selected from below the editor name, the Project Navigator and icon is highlighted. If the data type is used within the Ladder editor or Fieldbus editor, the tool opens displaying the location where used.



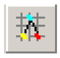
DataUsageExample.tif

Fig. 10-5: Data Usage Example

Browse Program for Icon Flow

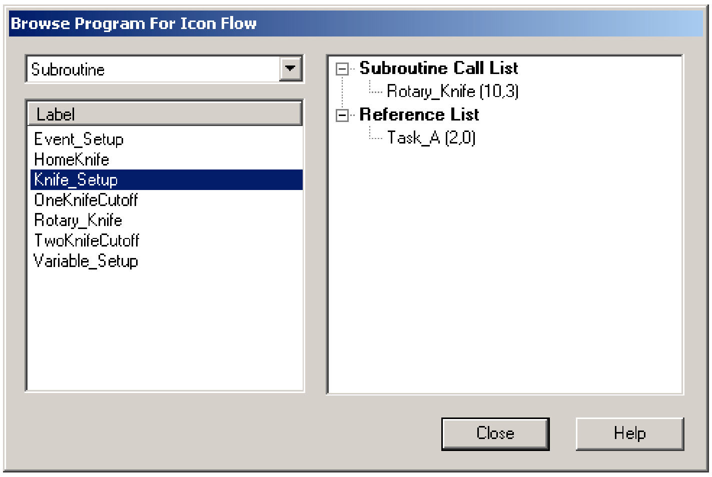
The Icon Flow menu selection allows the user to browse the current project for the use of subroutines and event functions. Subroutines and event functions can be located within any program task, subroutine or event function icon flow. This feature can also be used for:

- visualizing subroutine nesting (up to a maximum of 40) in a tree structure
- locating multiple references of subroutines and/or event functions, and
- overall relationship of subroutines and/or event functions

Selecting ***Browse ⇒ Icon Flow…*** or clicking  opens the *Browse Program for Icon Flow* window in Fig. 10-6. All program tasks, subroutines and event functions are scanned for the use of subroutines and event functions. The available selections in the drop-down list are listed in the table below.

Program Task	Subroutine
Initialization Subroutine	Event Functions

Table 10-4: Icon Flow Type Drop-Down List Selections



BrowseIconFlow.tif

Fig. 10-6: Browse Project for Subroutines and Event Functions

When an item is selected from the drop-down list (i.e., Subroutine), all labels of the selected types are displayed just below the drop-down list.

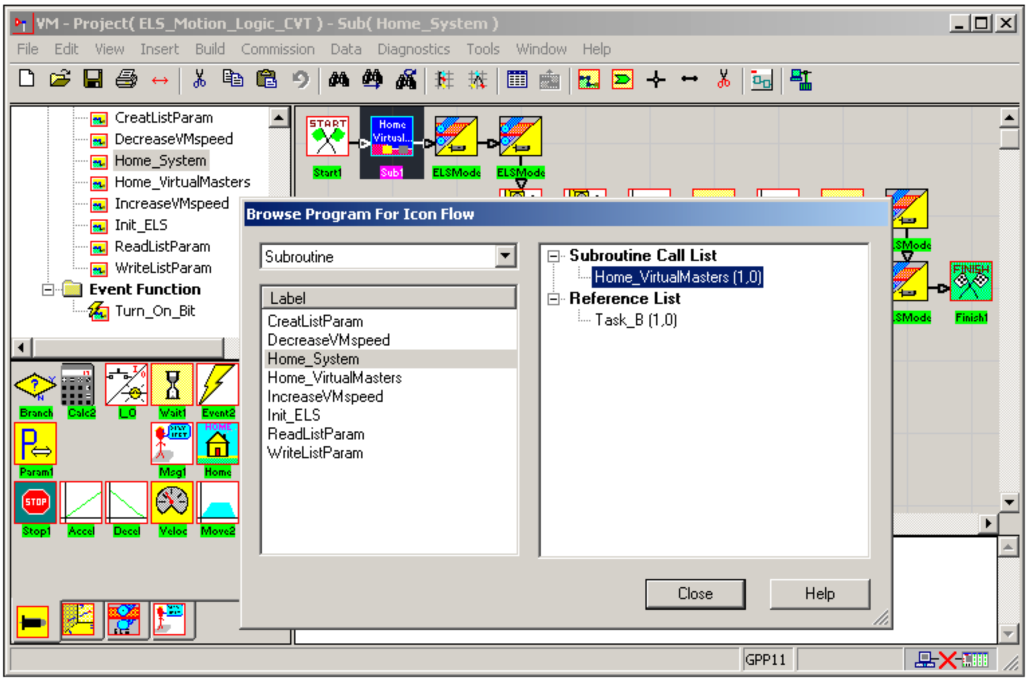
For example

If a user selects “Subroutine” from the drop-down list, the labels of all subroutines existing in the project are displayed. When a subroutine label is selected, the following tree items are possible:

- *Subroutine Call List* – This tree item is displayed if the selected subroutine or event function label contains subroutines and/or event function as part of its icon flow. (What is called from current icon flow).
- *Reference List* – This tree item is displayed if the selected subroutine or event function label is reference or called by another icon flow.

The sub-items that appear below each main List item are the actual locations in the project where the subroutines or event functions are located.

When a sub-item is selected, the Project Navigator and icon is highlighted.



IconFlowExample.tif

Fig. 10-7: Icon Flow Example

Note: Selecting a subroutine in the tree structure beyond 10 nested levels will not highlight the icon in the program flow. VisualMotion supports nested subroutines up to a maximum of 40 levels; however, the visual display functionality is limited to only 10 levels.

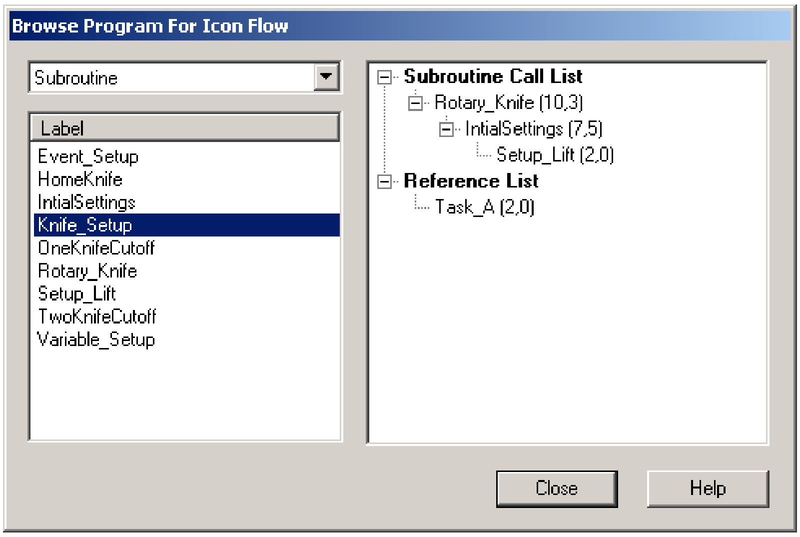
The user can navigate to actual subroutines and event function icon programs by selecting the label of the desired subroutine or event function. VisualMotion Toolkit's title bar indicates if the selected item is either a subroutine or an event function.

TitleBar.tif

Fig. 10-8: Title Bar Descriptions

Subroutine Nesting Example

When a project contains nested subroutines, selecting the subroutine label will display the nested subroutines in an easy to follow tree structure.

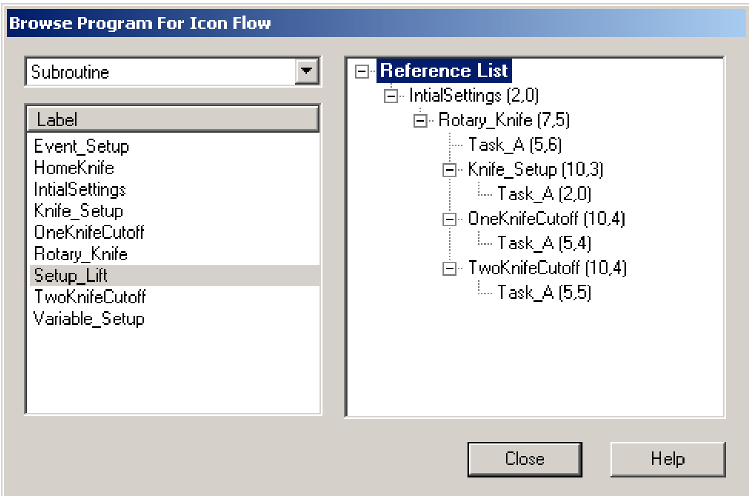


BrowseNestedSubs.tif

Fig. 10-9: Nested Subroutine Example

Referenced Subroutine Example

The user can expand the Reference List to locate where in the project a subroutine and/or event function is being called).



ReferenceSubroutines.tif

Fig. 10-10: Referencing Subroutine Example

10.3 Archiving and Restoring Project Data

VisualMotion 11 supports the full and selective archiving and restoring of motion and logic project data. Motion functionality can be archived and restored in offline, online and service modes. IndraLogic PLC data can only be archived and restored in online or service mode.

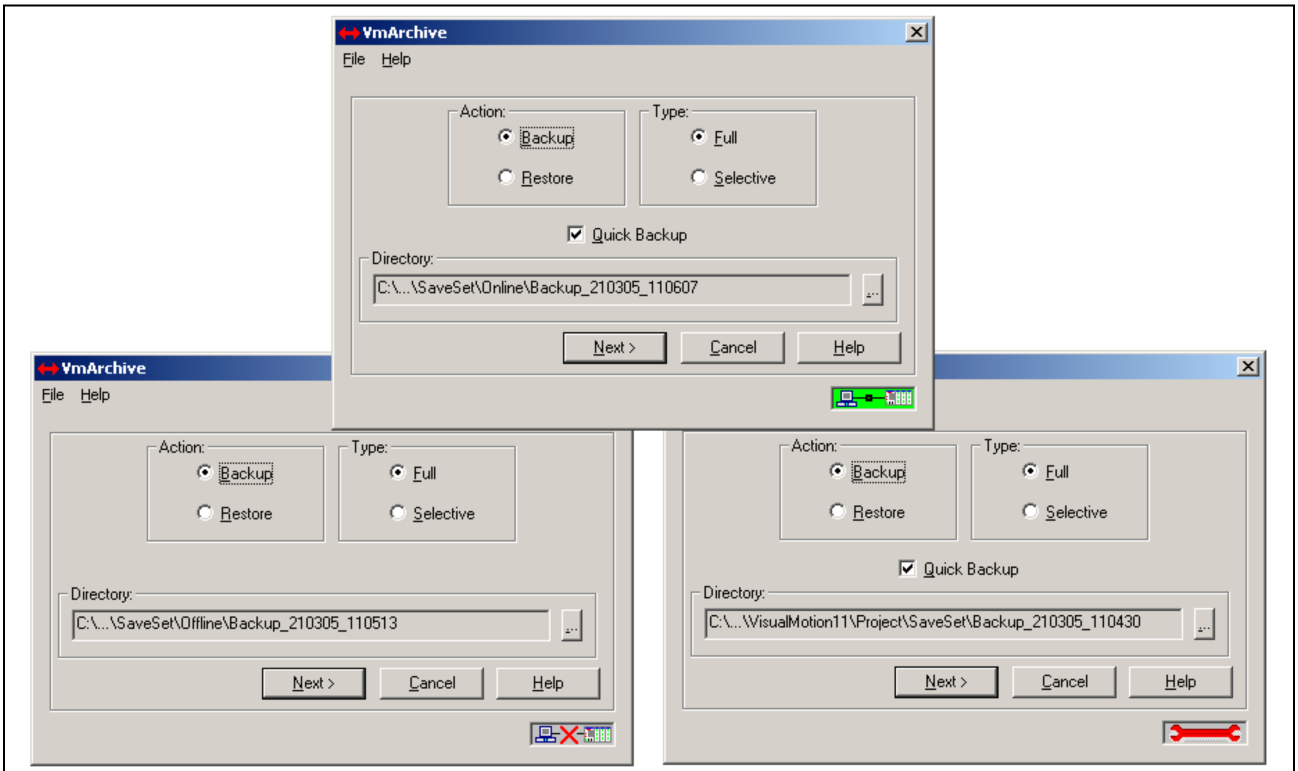
The VisualMotion VmArchive tool is launched from menu selection ***Commission ⇒ Archive ⇒ Project*** and is used to perform the archiving and restoring of VisualMotion project data. Project data for both motion and logic is referred to as components. When project components are archived, a *Backup* folder, containing the PC's current date and time, is created by default. The *Backup_datetime* folder is placed under a different location based on the current programming mode, as listed in the table below.

Mode	Archive Location
Online	…\project\project name**SaveSet\Online**\Backup_datetime
Offline	…\project\project name**SaveSet\Offline**\Backup_datetime
Service	…\project\project name**SaveSet**\Backup_datetime

Table 10-5: Archiving Folder Structure

A separate IndraLogic folder, containing the relevant PLC project data, is created and stored under either the online or service mode folder location.

Note: The default folder naming convention allows for multiple archive folders for any given date without worrying that a previous archive will be overwritten.



Commission_Archive.tif

Fig. 10-11: VmArchive Tool

System Project Components

Motion and logic project components are stored as individual data files under the appropriate folder locations as described in Table 10-5: Archiving Folder Structure.

The following table lists all the system components, including IndraLogic that can be archived and restored in online mode:

Component	Data File	Description
Program(s)	program.ex*	Compiled program file resident on control's memory. (.ex* - represents program #, i.e., .ex1, .ex2 and so on)
IndraLogic	--	refer to IndraLogic Component
Global Variables	globalintegers.dat	program global integer data file
	globalfloats.dat	program global float data file
A-Parameters	Axis0**.par	Axis parameter set (A-0-2001) (0** - represents axis number up to a maximum of 64)
C-Parameters	Control001.par	Control 1 parameter set (C-0-2001)
T-Parameters	TaskA.par	Task A parameter set (T-0-2001)
S/P-Parameters	Drive0**.par	Both S and P drive parameter sets (S-0-0192) (0** - represents drive number up to a maximum of 64)

Table 10-6: System Project Components

Control Log File: Chklist.txt

In addition to the files listed in Table 10-6, a log file named "Chklist.txt" is also created. This file contains relevant information about the archive, including time and date as well as hardware configurations for both the control and drives, control error log and a list of archived files for the control.

IndraLogic Component

The following table lists the individual IndraLogic project data files that can be archived and restored in online mode:

Data File	Description
DEFAULT.PRG	Boot project file containing all information required to load PLC program after booting
DEFAULT.CHK	Boot project code checksum
BOOT.SDB	Boot project symbol file of loaded boot project. After booting, data is copied to DOWNLOAD.SDB
config.dat	Project communication settings
DEFAULT.STS	Status file containing the state of the runtime system before ending (run/stop)
source.dat	IndraLogic source code containing PLC project data
DOWNLOAD.SDB	Source code symbol file downloaded with PLC program
persist.dat	Source code file containing the definitions for persistent variables
retain.bin	Snapshot of all retain data captured at the end of the PLC cycle when the archive command was issued

Table 10-7: IndraLogic Project Data

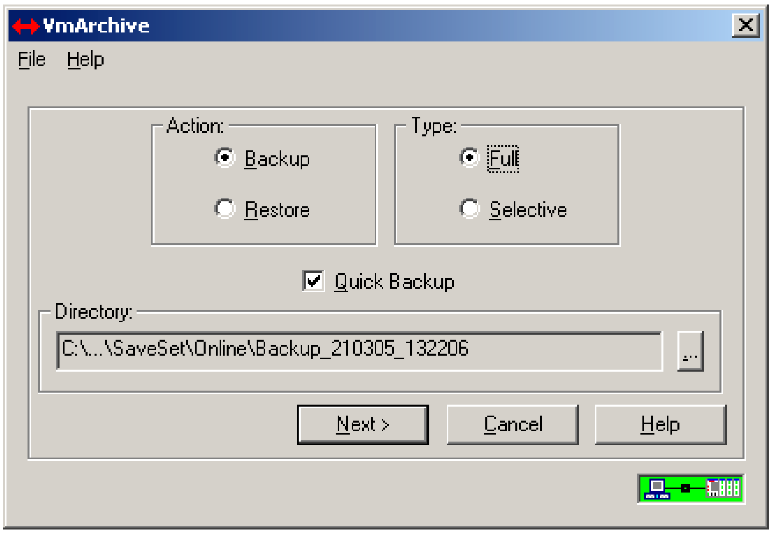
IndraLogic Log File: IndraLogic_log.txt

In addition to the files listed in Table 10-7, a log file named "IndraLogic_log.txt" is also created. This file contains a log of the archive procedures perform on IndraLogic data, including any errors.

Online Mode Archiving and Restoring

A full or selective online archive copies all current project data for both motion and logic stored on the control's memory and saves it to the default backup folder or user-defined folder location. A full or selective online restore copies previously archived control and drive data from the project folder to the control's memory.

Refer to System Project Components on page 10-12 for a complete list of control and drive data that can be archived or restored.



OnlineArchive.tif

Fig. 10-12: Online Archive

Quick Backup The Quick Backup checkbox, available for online and service modes, determines which save format to use when creating a backup. By default, the Quick Backup checkbox is selected. Removing the check will generate a more detailed *.par backup file for control and drive parameters. Refer to the following examples:

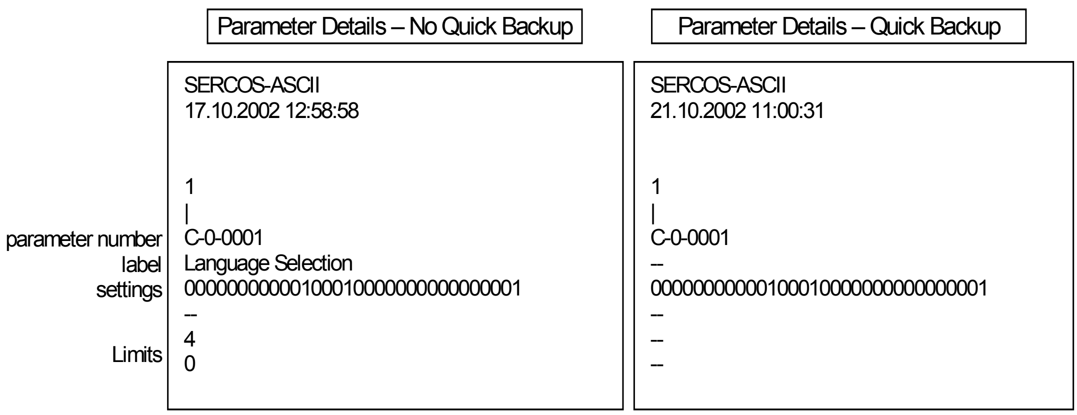


Fig. 10-13: Quick Backup Example

Rexroth Bosch Group

Online Full Backup

An online full backup automatically selects the following motion and logic project components:

- Program(s)
- IndraLogic
- Global Variables
- Axis Parameters
- Control Parameters
- Task Parameters
- Sercos and Product Parameters

Refer to System Project Components on page 10-12 for details.

To perform an online full backup:

1. Switch VisualMotion Toolkit to online mode.
2. Select ***Commission ⇒ Archive ⇒ Project***.
3. Select the Backup and Full radio buttons, change the default Directory (if desired) and click the **Next >** button.
4. From the *Online Project* window, select the **Start** button.

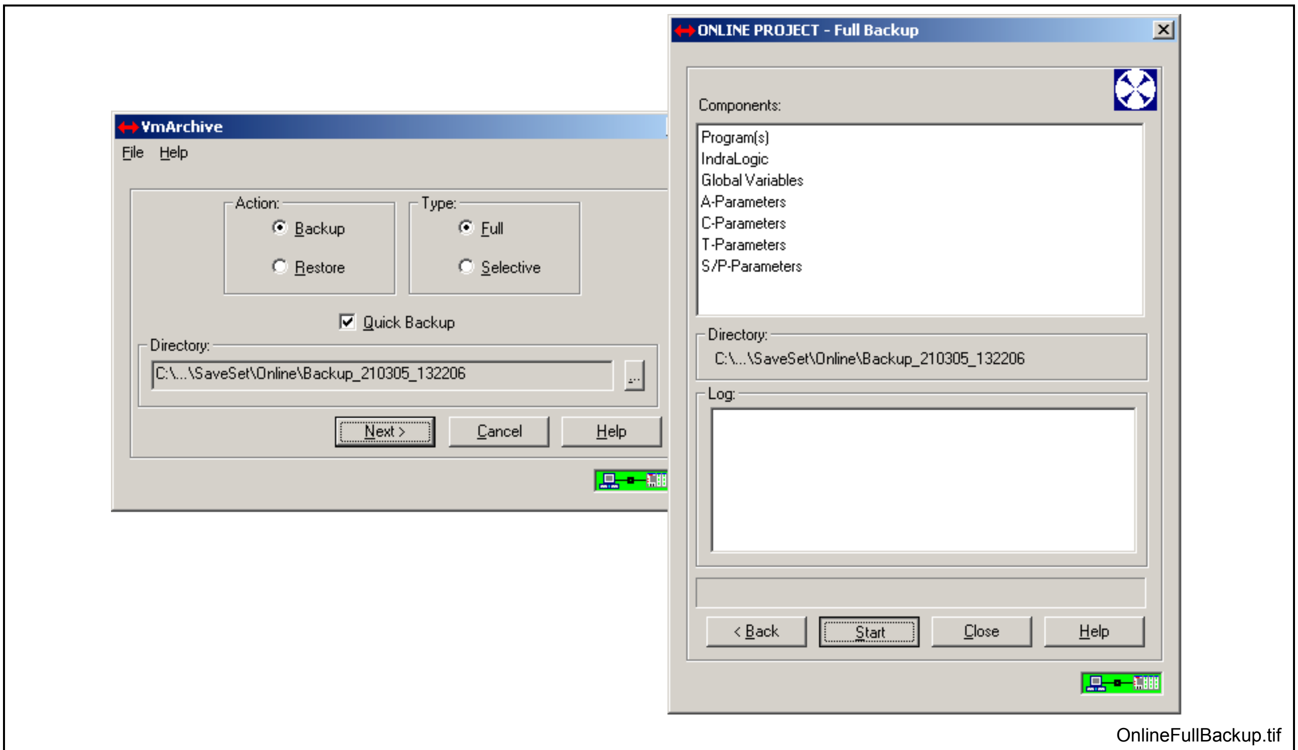
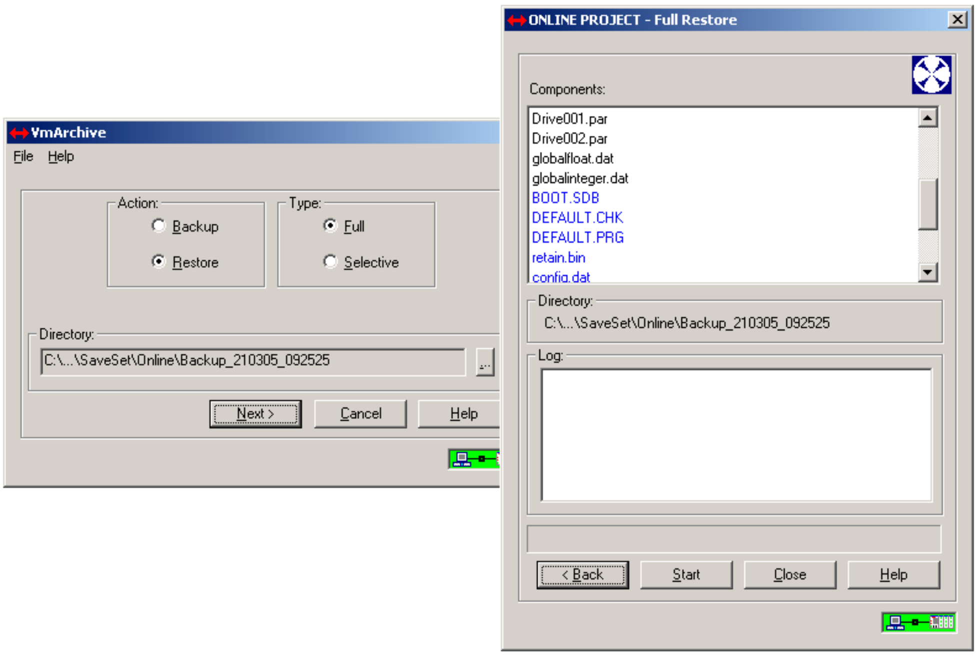


Fig. 10-14 Online Full Backup

Online Full Restore

An online full restore can only be performed if a valid Chklist.txt file exists in the selected **Directory**. The system reads the Chklist.txt file and displays a list of all the components that were originally backed up. Clicking on the ***Start*** button will begin the restore process.

Note: IndraLogic PLC project data appears as blue text.



OnlineFullRestore.tif

Fig. 10-15 Online Full Restore

A message will appear warning the user that all programs on the control's memory will be lost.

Note: The control must in *Parameter* mode before a Full Online Restore can be performed.

Communication Errors After Program Activation

During a restore procedure, the system is automatically switched to Sercos phase 4 when a restored program is activated in order to configure the axes. Switching to Sercos Phase 4 also runs the active program's Setup and Initialization tasks.

During program activation, modifications to serial port X10 or X16 communication parameters (C-0-0003, C-0-0004, etc.) to values different than that of the current settings can cause communication errors between VisualMotion and the control. This means that the user is communicating on the same serial port that is being modified to a different value by the restored program.

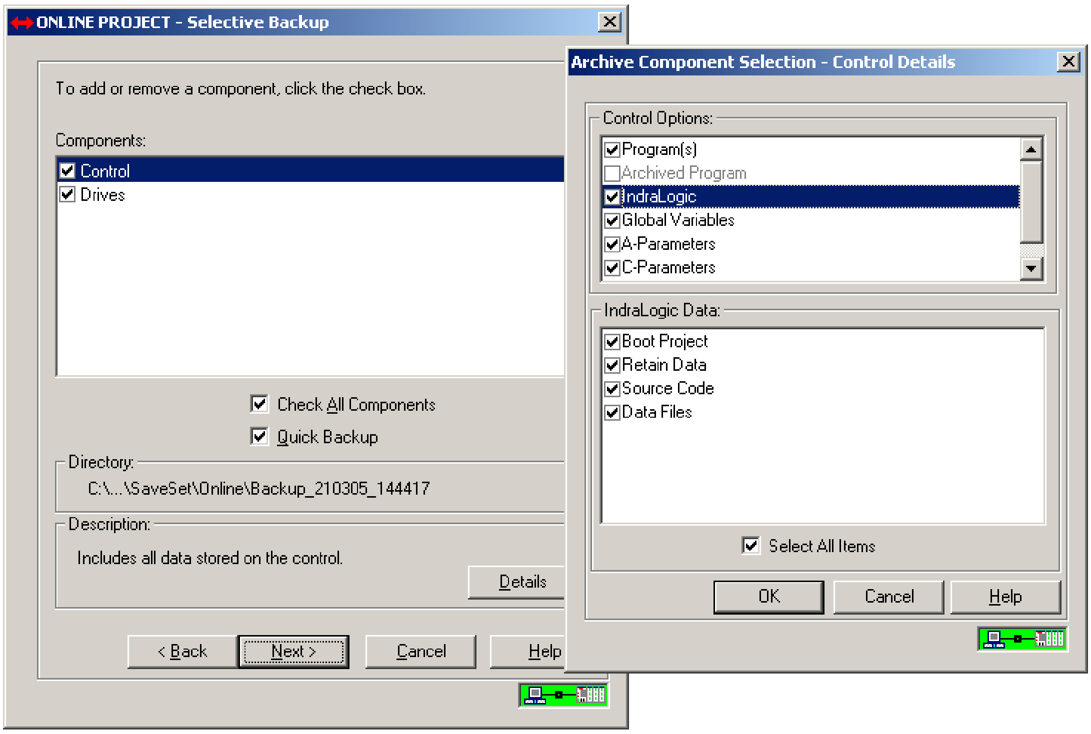
This can occur if the user program performs parameter transfers (Param1 icon) and/or parameter initializations (Setup Parameter) to the serial port communication parameters while the system is in Sercos phase 4. During this Sercos phase, any settings that are modified such as the baudrate can cause the loss of communication if the new setting is different from the current setting. This does not affect Ethernet or PCI communication.

After program activation is completed, the system is switched back to Sercos phase 2 and the restore process continues.

Online Selective Backup

An online selective backup allows the user to selectively choose which components to archive to the SaveSet folder. The *Selective Backup* window displays components for both the control and drive. The **Details** button towards the bottom of the window opens the *Archive Component Selection* window where the user can choose which details, of the selected component, to backup. By default, all items are selected. Unchecking an item will remove it for the list of items to backup.

Note: IndraLogic PLC project data is located under the Control component selection.

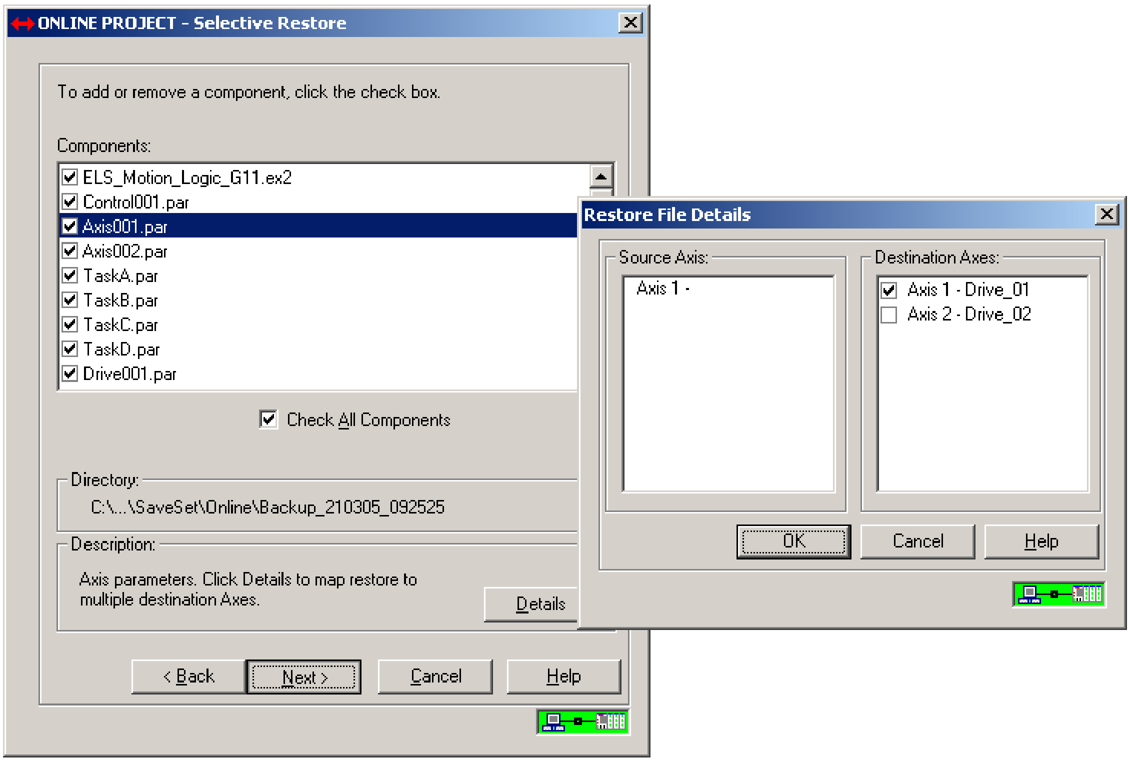


OnlineSelectBackup.tif

Fig. 10-16: Selective Online Backup

Online Selective Restore

An online selective restore allows the user to selectively choose which components to restore from the SaveSet\Online folder to the control's memory. The *Selective Restore* window displays all the components available in the selected SaveSet folder. By default, all items are selected. Unchecking an item will remove it for the list of items to backup. The **Details** button towards the bottom of the window is only available for axis and drive parameter files. This button opens the *Restore File Details* window where a user can restore a parameter file to the same drive number or any other drive available in the current project.



OnlineSelectRestore.tif

Fig. 10-17: Selective Online Restore

Note: Communication errors can occur when restoring a user program. Refer to Communication Errors After Program Activation on page 10-15 for details.

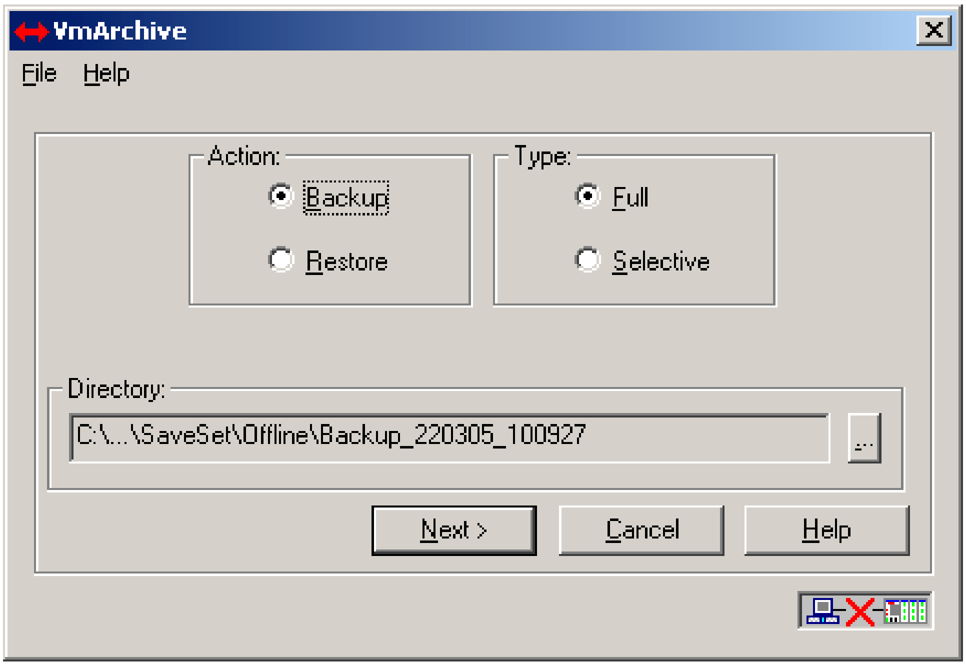
Rexroth
Bosch Group

Offline Mode Archiving and Restoring

A full or selective offline archive backs up only motion functionality for the currently opened project in VisualMotion Toolkit. All files (excluding subfolders) that are located in the project's main folder are backed up to the default backup folder or user-defined folder location. A full or selective offline restore copies the files from the offline backup folder to the project's main folder, overwriting existing files.

Refer to Table 10-5: Archiving Folder Structure for details.

Note: Offline mode archiving and restoring does not process IndraLogic project files. An active connection to the control is required. Refer to Online Mode Archiving and Restoring for details.



OfflineArchive.tif

Fig. 10-18 Offline Archive and Restore

Offline Full Backup

An offline full backup automatically selects all the files (excluding subfolders) located under the main project folder of the currently opened project and backs them up to the “…SaveSet\Offline\Backup” folder.

To perform an offline full backup:

1. Launch VisualMotion Toolkit and open the project to be backed up.
2. Select ***Commission ⇒ Archive ⇒ Project***.
3. Select the Backup and Full radio buttons, change the default Directory (if desired) and click the **Next >** button.
4. From the *Offline Project* window, select the **Start** button.

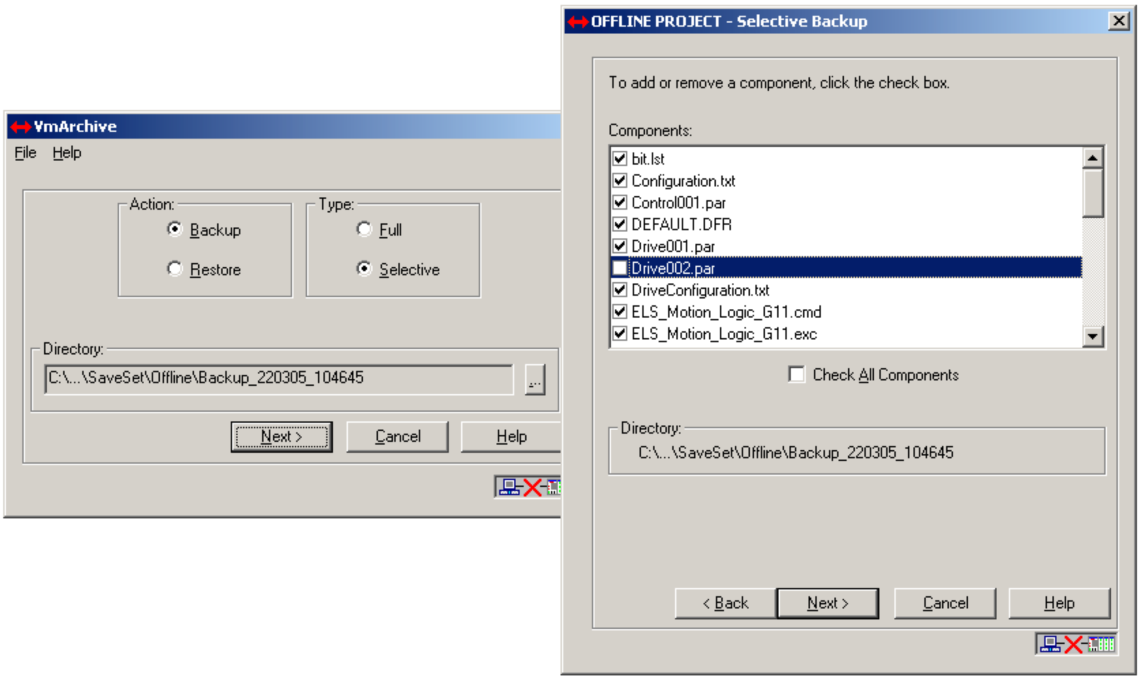
Offline Full Restore

An offline full restore copies all the files from the selected offline backup directory and overwrites the offline data of the current project opened in VisualMotion Toolkit. Clicking the ***Start*** button opens a message window warning the user that all current project files will be overwritten.

Note: After an offline restore, the icon editor reloads the new data into the project.

Offline Selective Backup

An offline selective backup allows the user to selectively choose which files to backup to the "…SaveSet\Offline\Backup" folder. The *Selective Backup* window displays all the files located under the main project folder. The user can uncheck those files that are not part of the project before creating a backup.



offlineSelectBackup.tif

Fig. 10-19: Offline Selective Backup

Offline Selective Restore

An offline selective restore allows the user to selectively choose which components to restore from the SaveSet\Offline folder to the current project. The *Selective Restore* window displays all the components available in the selected SaveSet\Offline folder. By default, all items are selected. Unchecking an item will remove it for the list of items to restore. The **Details** button towards the bottom of the window is only available for drive parameter files. This button opens the *Restore File Details* window where a user can restore a parameter file to any drive available in the current project.

Note: After an offline restore, the icon editor reloads the new data into the project.

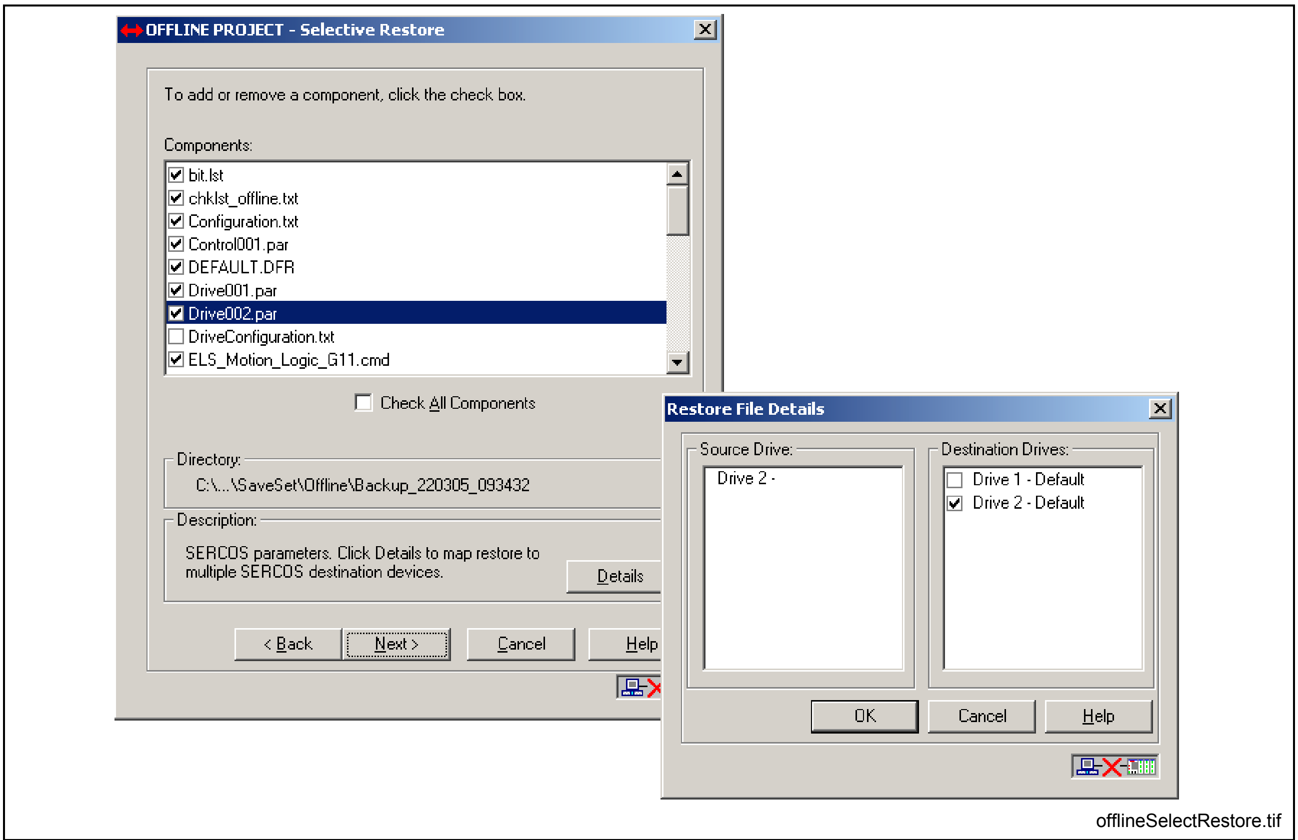


Fig. 10-20: Offline Selective Restore

Archive History and Flash Backup

Selecting ***Commission* ⇒ *Archive* ⇒ *Other …*** allows the user to obtain archive history information from the *History* tab and a list of online or offline backups that exists as part of the project data from the *Archives* tab. The Backup tab is used to create a backup of the non-volatile flash and to create a complete backup of the compact flash contents. The availability of tabs depends on the current programming mode.

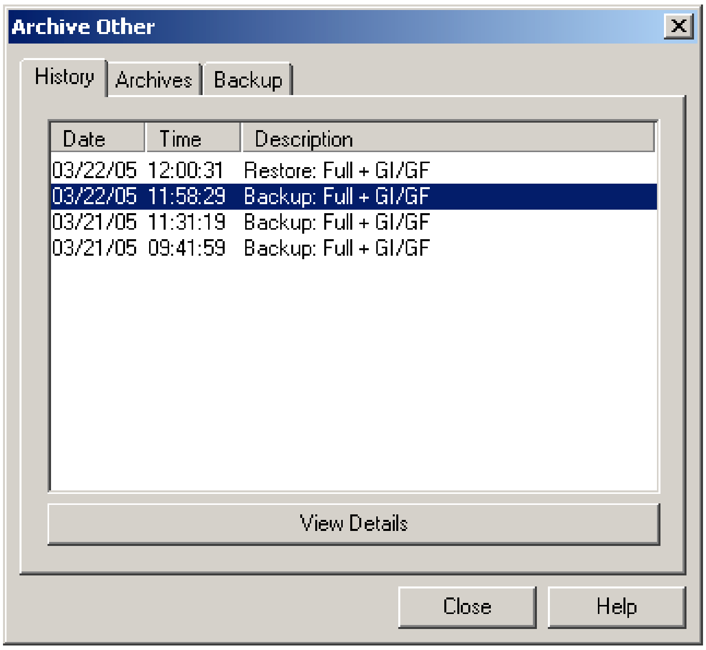
Refer to the following table for details:

Program Mode	History	Archives	Backup
Offline Mode	X	X	
Online Mode	X	X	X
Service Mode	X		X

Table 10-8: Archive Window Tabs

History

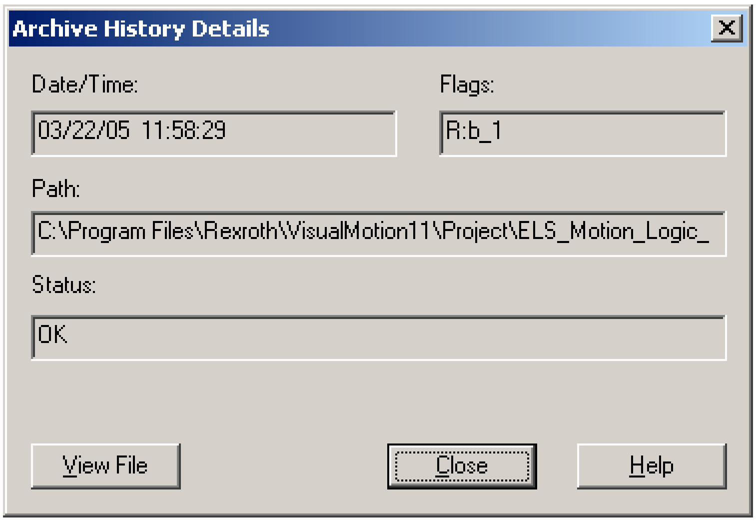
The *History* tab contains a list of the current project's archive history. This history is only available for backups and restores that were performed while the project was online with the control.



ArchiveHistory.tif

Fig. 10-21: Archive History

Double clicking on any list item or selecting the item and clicking the **View Details** button opens the *Archive History Details* window in Fig. 10-22.

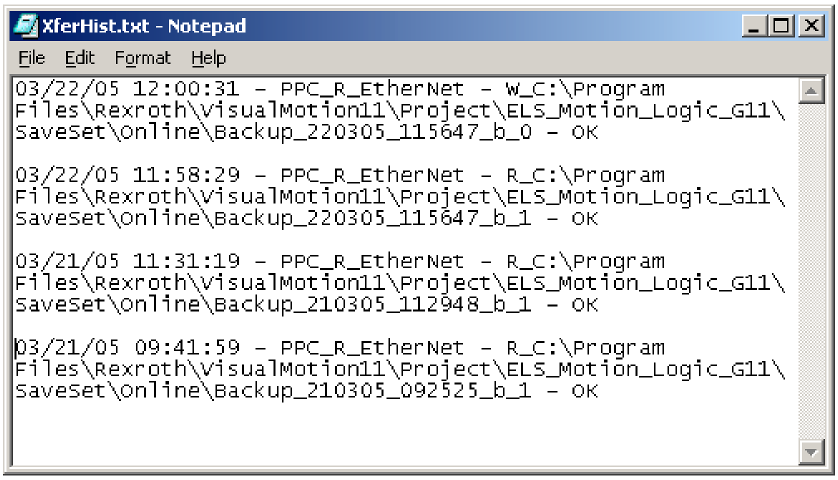


ArchiveHistoryDetails.tif

Fig. 10-22: Archive History Details

The *Archive History Details* window displays information for the currently selected archive. Information such as, Date/Time, Path and Status are displayed here. The *Flags* field indicates the VisualMotion data transfer operation code used for the archive. Refer to the VisualMotion Transfer Utility in the VisualMotion 11 Help System for details.

Clicking the **View File** button opens the ***XferHist.txt*** log file where the details of the current project's online archive history are saved.

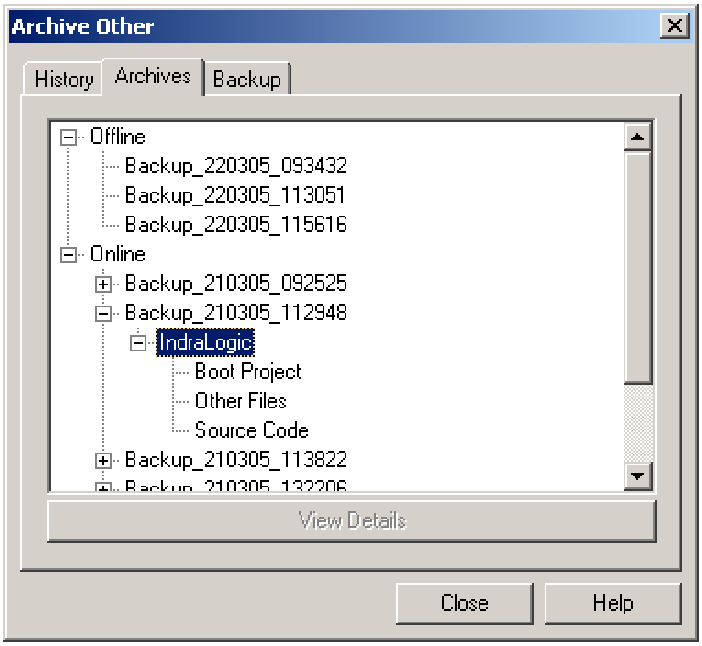


XferHistory.tif

Fig. 10-23: Transfer Utility History

Archives

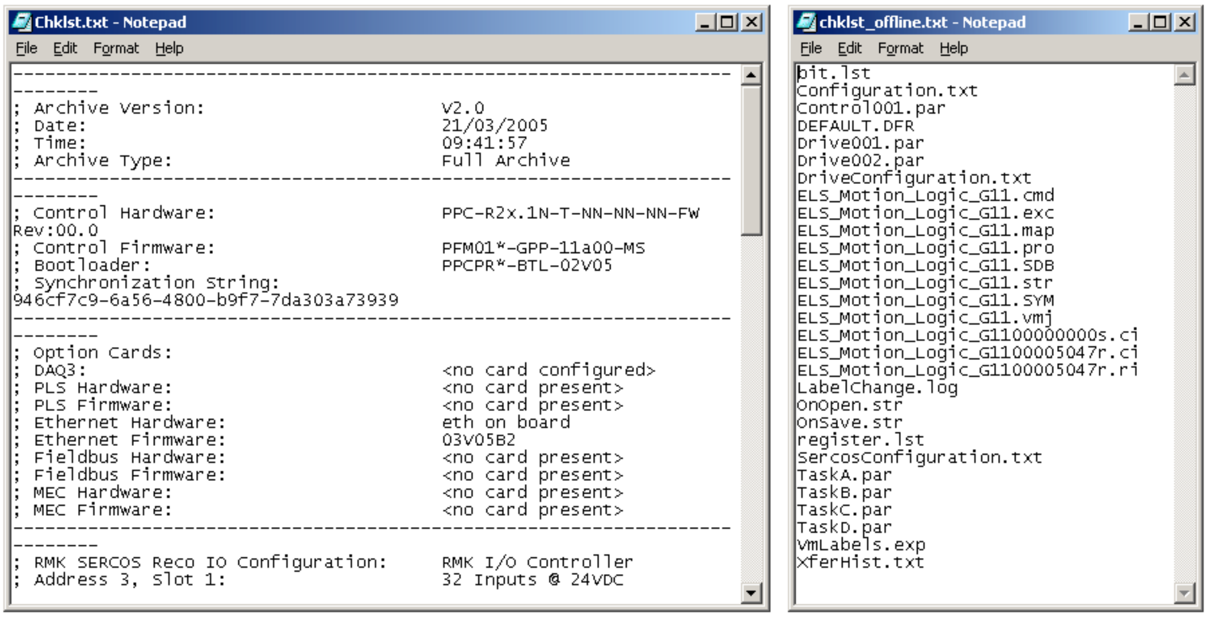
The *Archives* tab contains all offline and online backups and restores that were previously performed for the current project. When the project is in service mode, only archives that were performed in service mode will be displayed.



ArchiveArchives.tif

Fig. 10-24: Archives

Double clicking on any backup or restore list item or selecting the item and clicking the **View Details** button opens the ***ChkList.txt*** log file for online archives and the ***chklist_offline.txt*** log file for offline archives.



Chklist_log_files.tif

Fig. 10-25: ChkList.txt and Chklist_offline Log Files

NVRam Backup

The NVRam on VisualMotion controls store the file system, parameters, and variables. On power up, the contents of the NVRam are automatically copied to compact flash. If the control is to be replaced, the validity of the NVRam should be checked.

To create a valid backup of the NVRam:

1. Click on the **Create Backup** button under the *Non-volatile Data* section. When a valid copy of the NVRam exists, a depression labeled **Valid Backup Exists** will appear.

Note: NVRam restore functionality is not required. During power up, the NVRam is restored automatically from flash if a difference is detected between the current NVRam content and the copy backup in flash.

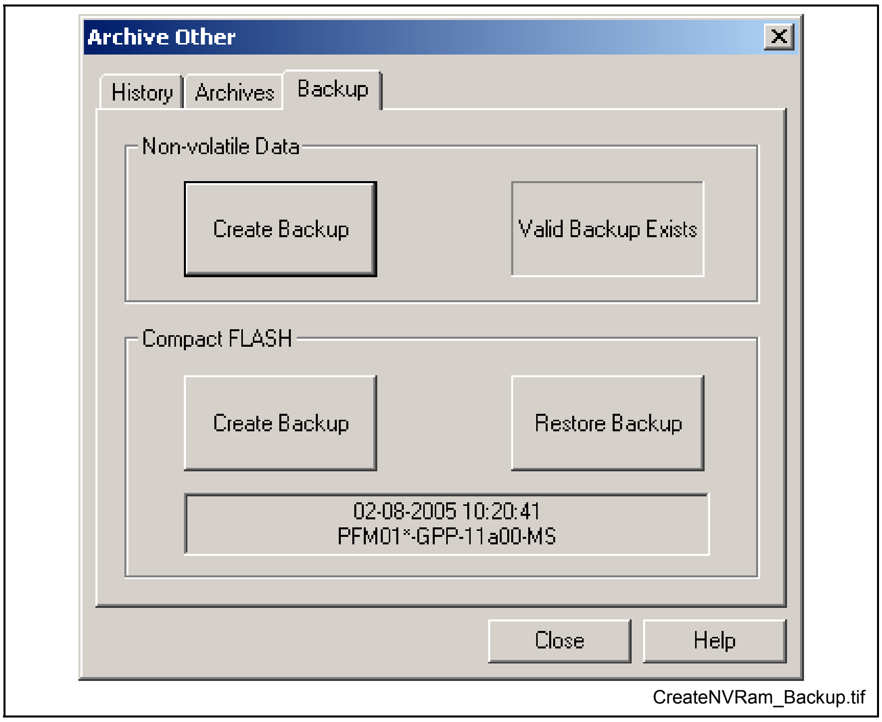


Fig. 10-26: Archive NVRam

Compact Flash Backup and Restore

For controls using a 32 MB compact flash card, the following procedure creates a complete backup of the compact flash memory area:

1. Clicking on the **Create Backup** button divide the compact flash memory into two equal parts and create a complete backup of the compact flash including the NVRam and firmware.
2. To restore the compact flash, click on the **Restore Backup** button.

10.4 Importing Project Data

When using the import feature, VisualMotion Toolkit is considered the target location. The source location is dependent on whether VisualMotion Toolkit is in offline or online mode.

Import Data in Offline Mode

The "*Import a Program and/or Data into the project*" window opens when selecting ***File ⇒ Import Project Component…*** in offline mode. Data is copied from a selected project or file to the project currently opened in VisualMotion Toolkit.

Transfer Data from

Specify the path and name of a project file (*.vmj) or individual file (i.e., system.prm) from where data will be imported to the current project. Select the browse button to help locate a file. Once a project or individual file is entered into the **Project path** field, the **Components** and **Details** fields display all data that is part of the project or file.

Note: This process can be used to import data from older file formats of VisualMotion into the current project.

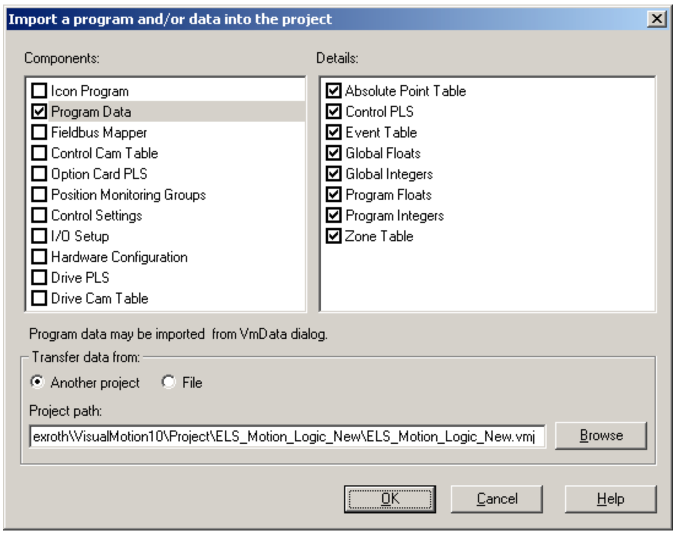
Components:

Data stored on the hard drive, either from another project file (*vmj) or individual file (i.e., system.prm), is display in the **Components** field. The components displayed are only those that are part of the project or file.

Details:

The details displayed are dependent on the component type selected. By default, all details are selected for each component. To de-select a detail, click on the appropriated checkbox. Once all components and details are selected, click the ***OK*** button to import the data to the current project opened in VisualMotion Toolkit.

Note: All data imported is automatically stored to the current project.



ImportComponentsoffline.tif

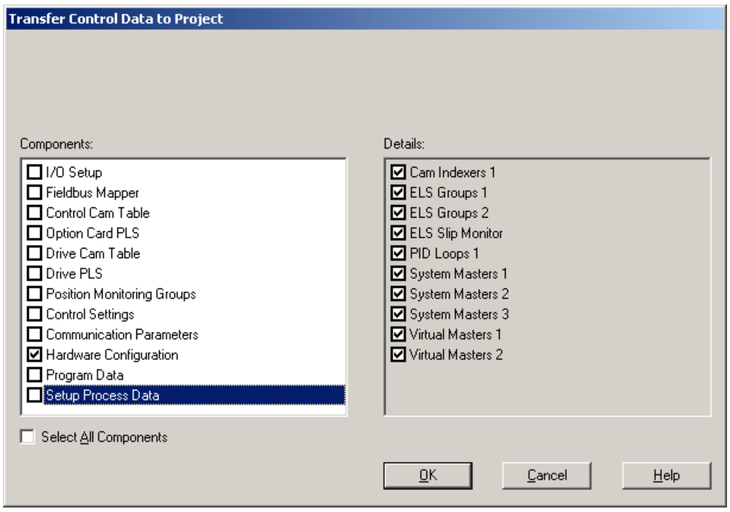
Fig. 10-27: Import a Program Data into the Project

Rexroth Bosch Group

Once the components have been imported from a project or file to the current project, the user can switch VisualMotion Toolkit to online mode. VisualMotion detects any modifications in data and request that the project be recompiled and downloaded to the control before switching to online mode.

Import Data in Online Mode

In online mode, selecting ***File ⇒ Import Project Component…*** opens the *Transfer Control Data to Project* window. Data is copied from the control's memory to the project currently opened in VisualMotion Toolkit.



ImportComponentsOnline.tif

Fig. 10-28: Transfer Control Data to *Offline Project*

Component and detail data displayed are those found on the control's memory for the currently active program. By default, Hardware Configuration is checked. Select any additional components and details to import from the control's memory to the project currently opened in VisualMotion Toolkit.

10.5 Exporting Project Data

When using the export feature, the currently opened project is considered the source location. The target location is dependent on whether VisualMotion Toolkit is in offline or online mode.

Export Data in Offline Mode

The *Transfer project data to another project or to files* window opens when selecting ***File ⇒ Export Project Component…*** in offline mode. Data is copied from the project currently opened in VisualMotion Toolkit to another project file (*.vmj) or individual file.

Transfer Data to

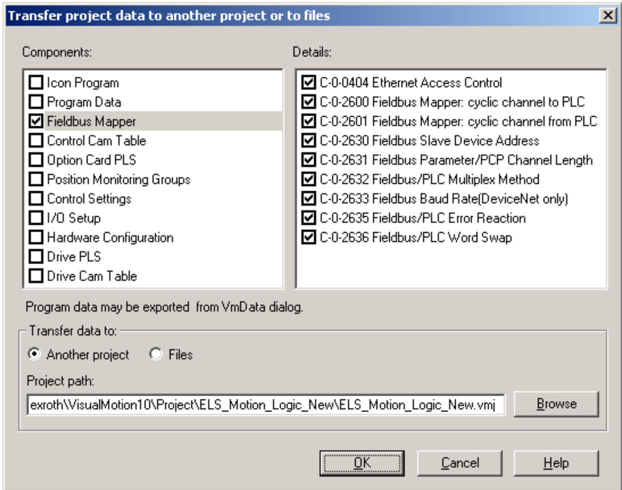
Specify the path and name of a project file (*.vmj) or folder (for an individual file) to where data will be exported from the current project. Select the browse button to help locate a file or folder.

Components:

Data components in the currently opened project are uploaded and displayed as available components that can be stored to another project file (*.vmj) or as individual data files. Components are selected by placing a check in the box next o the component label. Once a component is selected, details associated with the component are displayed to the right.

Details:

The details displayed are dependent on the component selected. By default, all details are selected for each component. To de-select a detail, click on the appropriated checkbox. Once all components and details are selected, click the ***OK*** button to export the data to the current project opened in VisualMotion Toolkit.

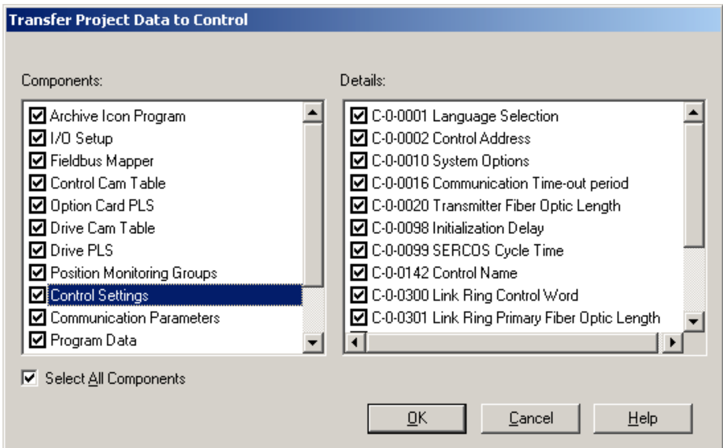


ExportComponentsoffline.tif

Fig. 10-29: Transfer Project Data to Another Project or to Files

Export Data in Online Mode

In online mode, selecting ***File ⇒ Export Project Component…*** opens the *Transfer Project Data to Control* window. Data is copied from the project currently opened in VisualMotion Toolkit to the control's memory.



ExportComponentsOnline.tif

Fig. 10-30: Transfer Project Data to Control

Component and detail data displayed is that of the project currently opened in VisualMotion Toolkit. By default, no components are checked. Select those components and associated details to export to the control's memory and click the *OK* button. Selecting the *Display All Components* and/or *Display All Details* checkboxes displays additional components and details not necessary associated with the current project.

Rexroth Bosch Group

Note: Selecting components and details that are visible only when the *Display All…* checkboxes are selected will write default values to each parameter.

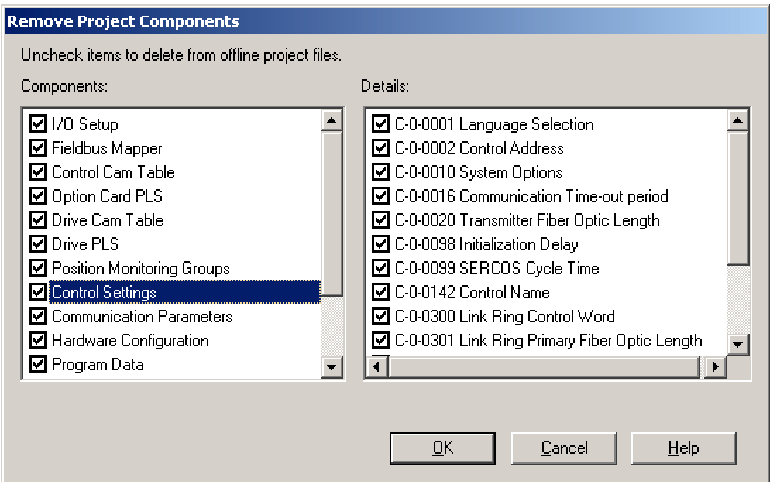
10.6 Remove Project Components

Remove project components allows the user to selectively remove complete components or just certain details of a component from the project currently opened in VisualMotion Toolkit.

Note: Project components removed using this feature are only removed from the project files. Project components downloaded to the control will remain resident in the control's memory.

Selecting ***File ⇒ Remove Project Components*** opens the window in Fig. 10-31. To remove a project component, uncheck the component(s) and/or detail(s) to remove from the project and click the OK button. The selected component and/or details will be removed from the project files.

Note: If a component or detail was removed in error, and was previously downloaded to the control, use VisualMotion's Import project Component feature. Refer to Import Data in Online Mode on page 10-26 for details.

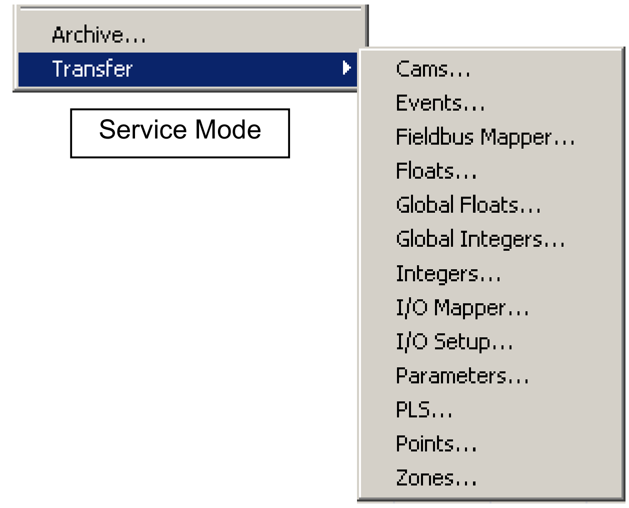


RemoveComponent.tif

Fig. 10-31: Remove Project Components

10.7 Transferring Data in Service Mode

The Transfer command is used to transfer data between the control and the PC. Selecting ***Commission*** ⇒ ***Transfer*** opens a third menu level where the user can select the type of data to transfer. The Transfer menu selection is only available in service mode.



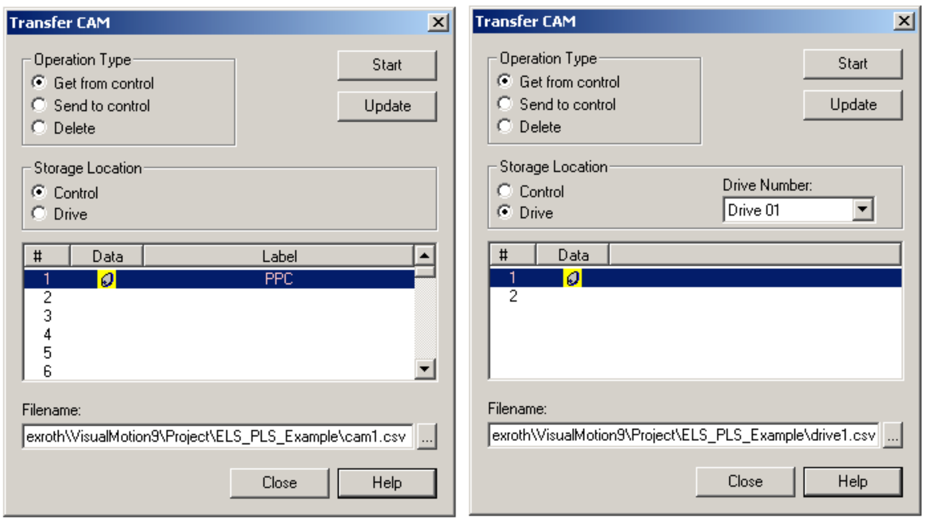
TransferMenu.tif

Fig. 10-32: The Transfer Menu

CAM

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***CAM...*** opens the *Transfer CAM* window in Fig. 10-33. This window can be used to transfer existing CAM tables from the control or drive to a file, and from a file to the control or drive. CAM table files are stored with the ".csv" file extension in the project directory along with other project files.

Note: The ".csv" format is a standard used by Microsoft Excel and other spreadsheet software programs.



Transfer_CAM.tif

Fig. 10-33: Transfer CAM

Note:	Before CAMs can be transferred, they must first exist in the control and/or drive. A CAM graphic under the **Data** column is an indication that the control or drive contains a CAM table. CAM tables are created and downloaded by selecting ***Tools ⇒ CAM Builder*** from Rexroth VisualMotion Toolkit's main menu.

The *Transfer CAM* window allows access to built CAMs using the following operation types:

- **Get From Control –** retrieves the selected CAM number from the specified **Storage Location** (Control or Drive) on the control's memory.
- **Send to Control –** downloads the selected CAM table file to the control.
- **Delete** - deletes the selected CAM number from the specified storage location (Control or Drive) on the control's memory.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field and a CAM number is selected. Valid filenames for CAM tables have a "csv" extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

To refresh the list of CAM tables, click on the **Update** button.

There are 40 CAM tables possible on controls. Each drive in a system can have up to 2 CAM tables. These tables are dynamically allocated to conserve memory when not used.

CAM tables may be downloaded or deleted at any time as long as the CAM table number is not active. If the CAM is already active, the control responds with a communication error, "CAM is already active for axis 'x' ". To download a new table, either switch into parameter mode or deactivate the CAM for all axes. (A CAM is active when it is assigned to an axis by the CAM icon.)

Events

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***Events*** opens the following window:

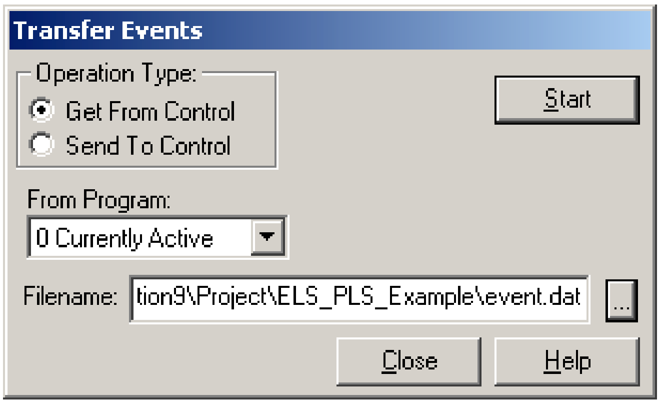


Fig. 10-34: Transfer Events

The *Transfer Events* window allows access to event tables using the following two operation types:

- **Get From Control –** retrieves and saves the events table as a data file “event.dat” from the selected program to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected file “event.dat” to the selected program on the control.

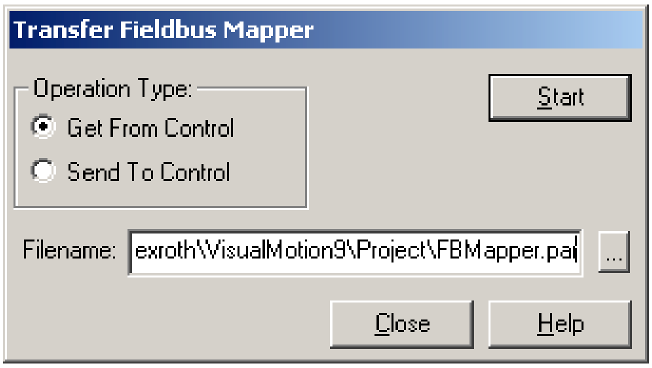
Note: Use the **Send to Control** operation type to download older event table file formats “event.evt” to the selected program.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for event tables have a “dat” or “evt” extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

Fieldbus Mapper

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***Fieldbus Mapper*** opens the following window:



Transfer_Fieldbus.tif

Fig. 10-35: Transfer Fieldbus Mapper

The *Transfer Fieldbus Mapper* window allows access to fieldbus mapper data using the following two operation types:

- **Get From Control –** retrieves and saves the fieldbus mapper as a parameter file “FBMapper.par” from the currently active program to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected file “FBMapper.par” to the currently active program on the control.

Note: Use the **Send to Control** operation type to download older fieldbus mapper file formats “FBMapper.prm” to the currently active program.

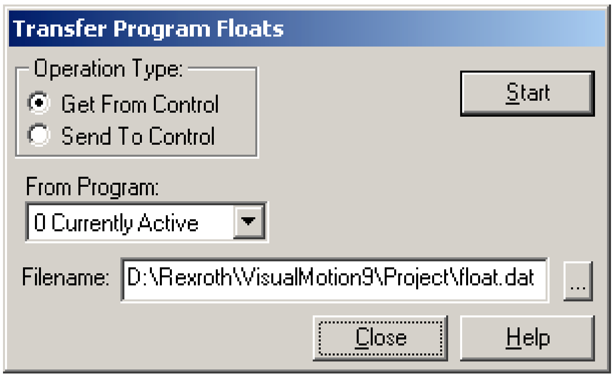
Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for a fieldbus mapper have a “par” or “prm” extension. Enter a filename, with a valid extension, directly in the

Filename field or click the browse button, locate a directory and then specify a filename.

Floats

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***Floats*** opens the following window:



Transfer_Float.tif

Fig. 10-36: Transfer Program Floats

The *Transfer Program Floats* window allows access to program floats using the following two operation types:

- **Get From Control –** retrieves and saves the program floats as a data file “float.dat” from the selected program to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected file “float.dat” to the selected program on the control.

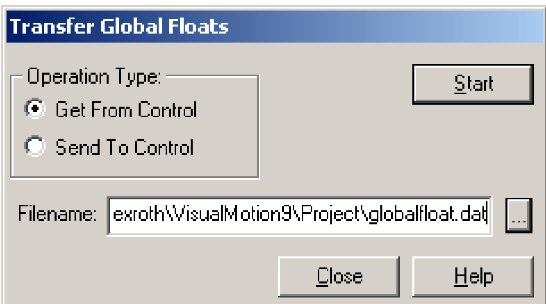
Note: Use the **Send to Control** operation type to download older float file formats “float.vtr” to the selected program.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for floats have a “dat” or “vtr” extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

Global Floats

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***Global Floats*** opens the following window:



Transfer_GlobalFloat.tif

Fig. 10-37: Transfer Global Floats

The *Transfer Global Floats* window allows access to program floats using the following two operation types:

- **Get From Control –** retrieves and saves the global floats as a data file “globalfloat.dat” from the control’s memory to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected file “globalfloat.dat” to the control’s memory.

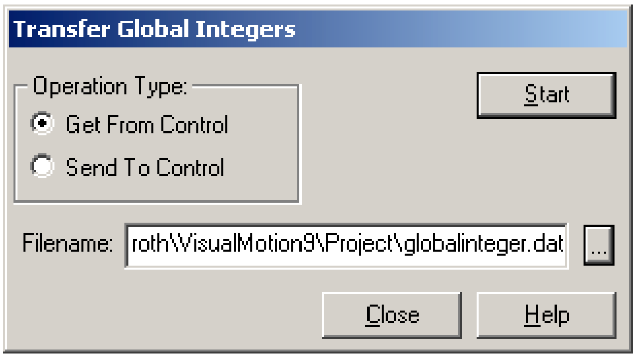
Note: Use the **Send to Control** operation type to download older global float file formats “gloabalfloat.vtr” to the control’s memory.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for global floats have a “dat” or “vtr” extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

Global Integers

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***Global Integers*** opens the following window:



Transfer_GlobalInteger.tif

Fig. 10-38: Transfer Global Integers

The *Transfer Global Integers* window allows access to program floats using the following two operation types:

- **Get From Control –** retrieves and saves the global integers as a data file “globalinteger.dat” from the control’s memory to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected file “globalinteger.dat” to the control’s memory.

Note: Use the **Send to Control** operation type to download older global integer file formats “globalinteger.vtr” to the control’s memory.

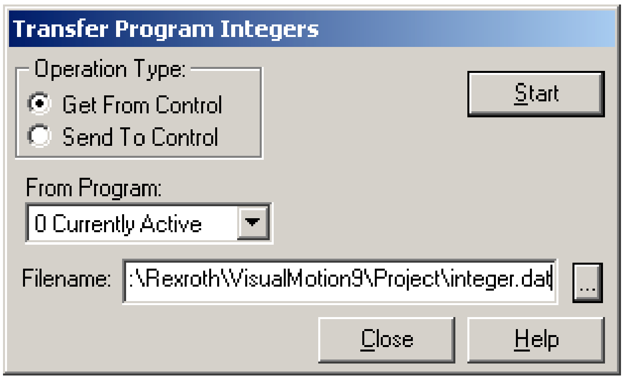
Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for global integers have a “dat” or “vtr” extension. Enter a filename, with a valid extension, directly in the

Filename field or click the browse button, locate a directory and then specify a filename.

Integers

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***Integers*** opens the following window:



Transfer_Integer.tif

Fig. 10-39: Transfer Program Integers

The *Transfer Program Integers* window allows access to program integers using the following two operation types:

- **Get From Control –** retrieves and saves the program integers as a data file “integer.dat” from the selected program to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected file “integer.dat” to the selected program on the control.

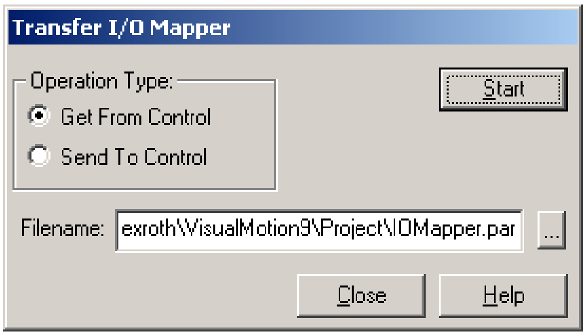
Note: Use the **Send to Control** operation type to download older program integer file formats “integer.vtr” to the selected program.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for integers have a “dat” or “vtr” extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

I/O Mapper

The I/O Mapper is available in support of pre-GPP/GMP 10 firmware. Selecting ***Commission*** ⇒ ***Transfer*** ⇒ **I/O Mapper** opens the following window:



Transfer_IOmapper.tif

Fig. 10-40: Transfer I/O Mapper

The *Transfer I/O Mapper* window allows access to an I/O Mapper using the following two operation types:

- **Get From Control –** retrieves and saves the I/O Mapper as a parameter file “IOMapper.par” from the currently active program to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected file “IOMapper.par” to the currently active program on the control.

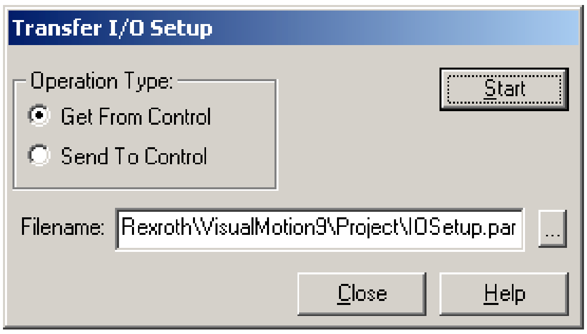
Note: Use the **Send to Control** operation type to download older I/O Mapper file formats “IOMapper.iom” to the currently active program.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for I/O Mapper have a “par” or “iom” extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

I/O Setup

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ **I/O Setup** opens the following window:



Transfer_IOsetup.tif

Fig. 10-41: Transfer I/O Setup

Rexroth
Bosch Group

The *Transfer I/O Setup* window allows access to the I/O Configuration using the following two operation types:

- **Get From Control –** retrieves and saves the I/O Setup as a parameter file "IOSetup.par" from the currently active program to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected file "IOSetup.par" to the currently active program on the control.

Note: Use the **Send to Control** operation type to download older I/O Setup file formats "IOSetup.prm" to the currently active program.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for I/O Setup have a "par" or "prm" extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

Parameters

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***Parameters*** opens the following window:

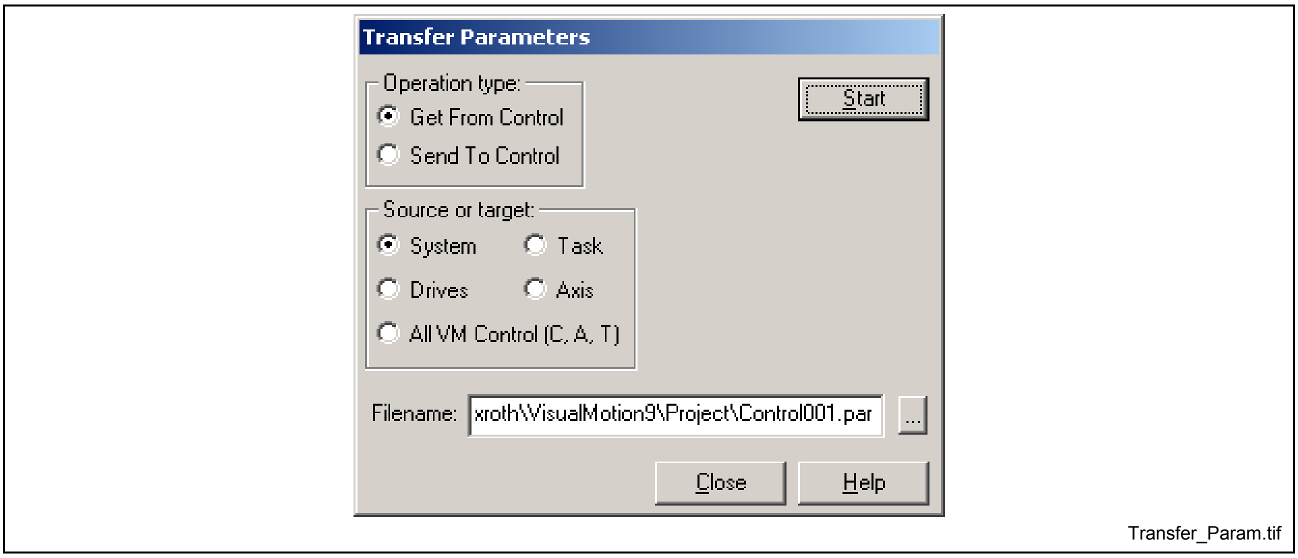


Fig. 10-42: Transfer Parameters

The *Parameters* window allows access to parameters using the following two operation types:

- **Get From Control -** saves the selected source type to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected filename type to the control.

Note: Use the **Send to Control** operation type to download older parameter file formats "*.prm" to the selected **Source or Target**.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for parameters have a “par” or “prm” extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

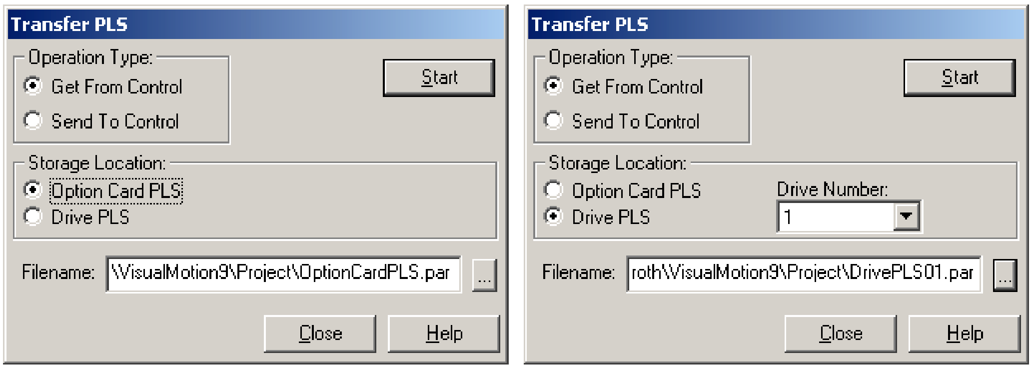
All VM Control (C, A, T) parameters can be transferred at one time, or individually. In addition, the parameter set for a selected drive that is connected to the control can be transferred through the SERCOS communication system. The following table lists the different file types available:

Source or Target	Filename	Description
System	Control001.par	Parameters listed in C-0-2001
Task	Task*.par	Parameters listed in T-0-2001 where * = A, B, C or D
Axis	Axis0**.par	Parameters listed in A-0-2001 where ** = axis number up to 64
Drives	Drive0**.par	Parameters listed in S-0-0192 where ** = drive number up to 64

Table 10-9: Available Parameters for Transfer

PLS

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***PLS*** opens the Transfer PLS window in Fig. 10-33. This window can be used to transfer existing Option Card and Drive PLSs from the control or drive to a file, and from a file to the control or drive. Option Card and Drive PLS files are stored as parameters using the ".par" file extension.



Transfer_PLS.tif

Fig. 10-43: Transfer PLS

Note: Before Option Card or Drive PLS files can be transferred, they must first exist in the control and/or drive. Option Card and Drive PLSs are created and downloaded by selecting ***Commission*** ⇒ ***PLS*** from Rexroth VisualMotion Toolkit’s main menu.

The *Transfer PLS* window allows access to existing PLS configurations using the following two operation types:

Rexroth
Bosch Group

- **Get From Control –** retrieves and saves the selected Option Card or Drive PLS file “OptionCardPLS.par” from the specified storage location to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected Option Card or Drive PLS file to the control.

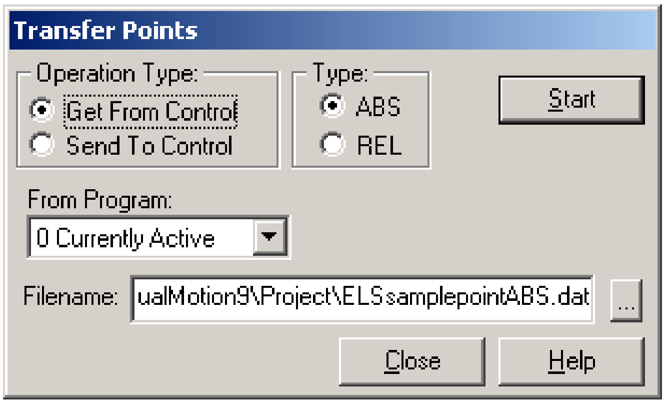
Note: Use the **Send to Control** operation type to download older Option Card or Drive PLS file formats “OptionCardPLS.prm” to the currently active program.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for PLS have a “par” or “prm” extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

Points

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***Points*** opens the following window:



Transfer_Points.tif

Fig. 10-44: Transfer Points

The *Transfer Points* window allows access to point tables using the following two operation types:

- **Get From Control –** retrieves and saves the point tables as a data file “pointABS.dat” and/or “pointREL.dat” from the selected program to the location entered in the **Filename** field.

Note: If the selected program contains both relative and absolute point tables, a data file for each type will be created when using the **Get From Control** operation type.

- **Send to Control –** downloads the selected file “pointABS.dat” or “pointREF.dat” to the selected program on the control. Relative and absolute point data files must be sent to the control, one at a time.

Note: Use the **Send to Control** operation type to download older relative and absolute file formats “*.abs” or “*.rel” to the selected program.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for relative or absolute points have a “dat” , “rel” or “abs” extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

Zones

Selecting ***Commission*** ⇒ ***Transfer*** ⇒ ***Zones*** opens the following window:

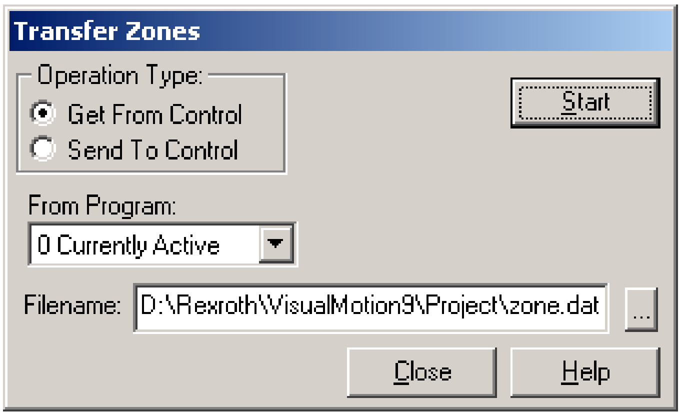


Fig. 10-45: Transfer Zones

The *Transfer Zones* window allows access to the zone table using the following two operation types:

- **Get From Control –** retrieves and saves the zone table as a data file “zone.dat” from the selected program to the location entered in the **Filename** field.
- **Send to Control –** downloads the selected file “zone.dat” to the selected program on the control.

Note: Use the **Send to Control** operation type to download older zone file formats “zone.zon” to the selected program.

Start Button Activation

The ***Start*** button becomes active only after a valid filename is entered in the **Filename** field. Valid filenames for zones have a “dat” or “zon” extension. Enter a filename, with a valid extension, directly in the **Filename** field or click the browse button, locate a directory and then specify a filename.

10.8 Labels

Note: The Labels menu selection is only available in service mode.

Labels are symbolic names assigned to system resources, such as axes, drives, etc.. Labels may also be used for absolute or relative point names, or in place of "literal" constant or variable values in expressions. For example, once assigned, the label "PI" can be used throughout a program, instead of repeatedly entering the literal value 3.14159.

Labels may use up to twenty ASCII characters and are case-sensitive. Blank spaces are not allowed within a symbol. Use a printable character as a separator if it is required for clarity. For example, "next_move," rather than "next move". The first character of a label must be an alpha character.

The control compiler, used for both icon and text language programming, allows the use of a literal integer value (i.e., a number such as "1" or "5"). Provided it is within the range of integers that are valid for the specified argument. Integers used to specify system devices, such as an axis or drive, must be within the range permitted by the complete Rexroth VisualMotion system and installed software.

For example: a Rexroth VisualMotion system with eight digital drives installed can specify an axis or drive using an integer from 1 to 8. The compiler must be able to resolve a symbol used as a table index argument to an integer index within the range, or size, of the table.

Rexroth VisualMotion has a number of keywords, which it uses for command instructions. These keywords cannot be used as labels. If a keyword is used as a label, Rexroth VisualMotion will issue the error "Label is a Keyword!" when the user tries to save the label.

Variable Labels

Selecting ***Edit*** ⇒ ***Edit Labels*** ⇒ ***Variable Labels*** opens the **User Defined Labels** window, used to provide symbolic ASCII names for Float, Integer, Global Float, Global Integer, Absolute or Relative point values.

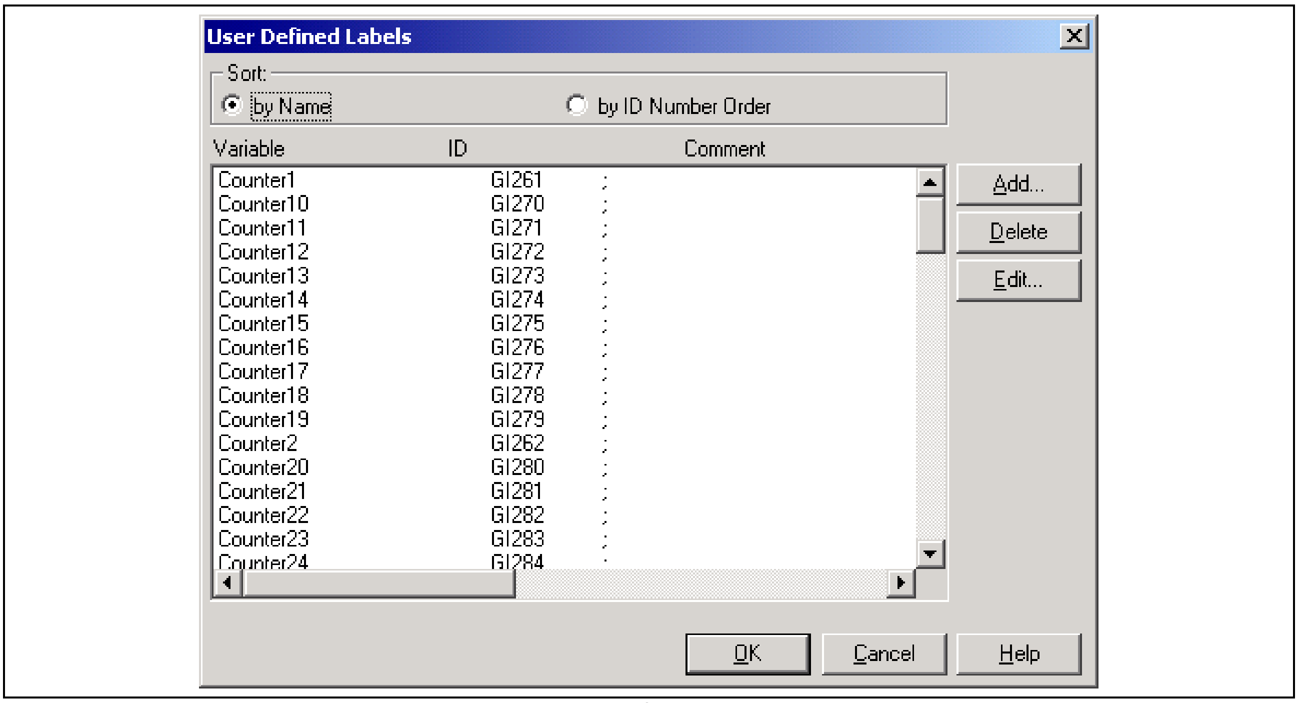


Fig. 10-46: "User Defined Labels" Window

The **User Defined Labels** window allows you to assign an ASCII name to a value or system component, as previously described. To sort the list alphabetically by label, select the radio box **by Name**. . To sort the list alphabetically by ID, select the radio box **by ID Number Order**. After labels are defined, instead of explicitly entering a value or redefining a system component, the label can be entered by accessing the **User Defined Labels** window and selecting the appropriate label.

To add a new label:

1. Click **Add…** to open the **Add Variable Label** window below.

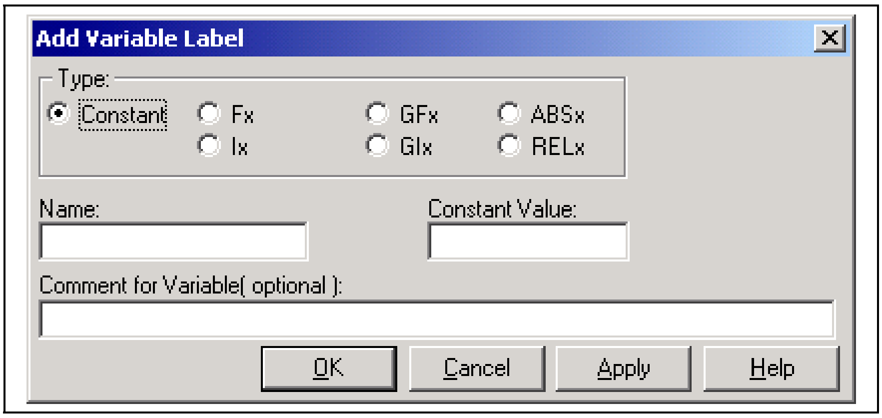


Fig. 10-47: "Add Variable Label" Window

2. Enter the desired **Type**, **Name**, **Constant Value** (if applicable) and **Comment** (up to 80 characters).
3. Click **Apply** or **OK**.

To delete an existing label:

1. Highlight a desired label and click **Delete**.

To edit an existing label:

1. Highlight a desired label and click **Edit…** .

   The **Edit User Label** window opens just below. The label's **Type**, **Name**, **Global Integer** and **Comment** are filled in as they were originally designated.

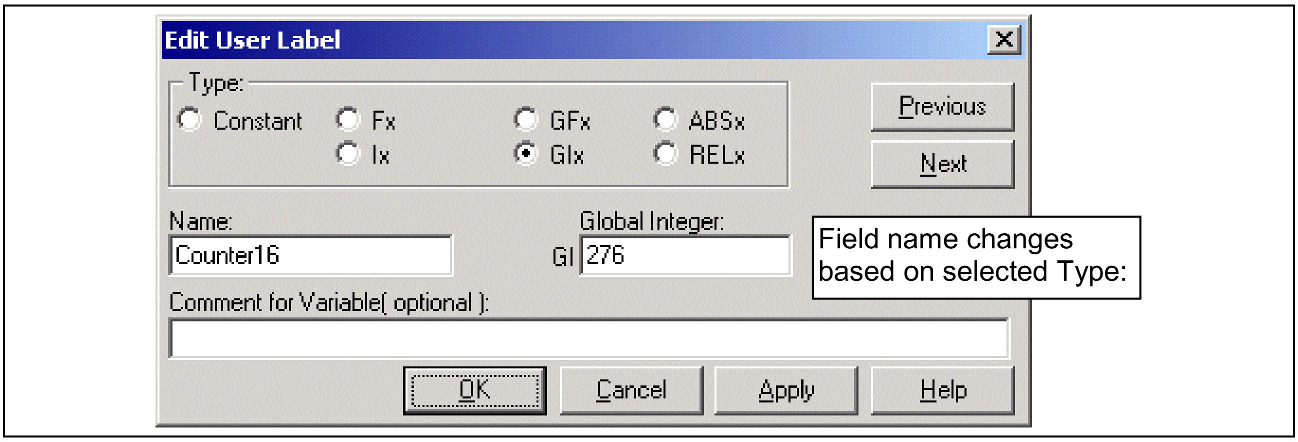


Fig. 10-48: "Edit User Label" Window

2. Enter the desired **Type**, **Name**, **Global Integer** (if applicable) and **Comment** (up to 80 characters).
3. Click **Apply** or **OK**.

Register Labels

Selecting ***Edit*** ⇒ ***Edit Labels*** ⇒ ***Register Labels*** opens the **Register Labels** window, used to provide symbolic ASCII names for the Rexroth VisualMotion control status and I/O registers.

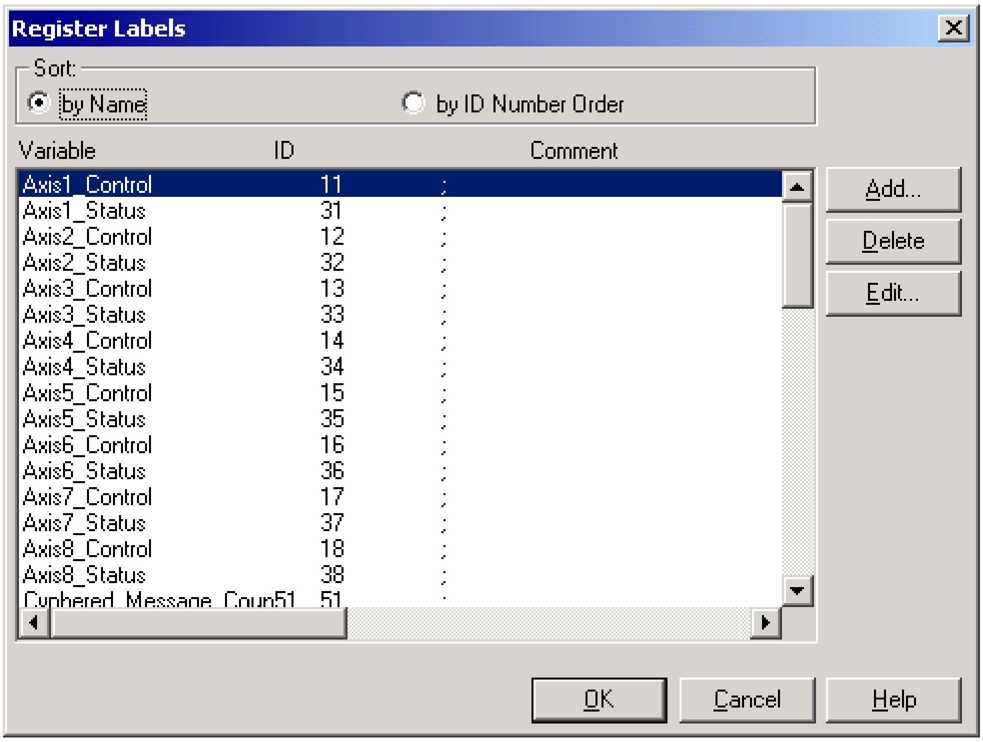


Fig. 10-49: "Register Labels" Window

To sort the list alphabetically by label, select the radio box **by Name**. To sort the list alphabetically by ID, select the radio box **by ID Number Order**. After register labels are assigned and the program is saved, the labels are embedded in the motion program (the .str file) and will not be lost if the program is later transferred to a different Rexroth VisualMotion system. New programs are loaded with the default register names. Refer to chapter 4, Registers, for default register names.

To add or edit a register label:

1. Click **Add…** or select the desired register and click **Edit…** .

   An Add/Edit Register Labels window opens. The **Name** must be explicitly entered. (When editing an existing label, the current **Name**, **Register Number** and **Comment** are automatically filled in.)

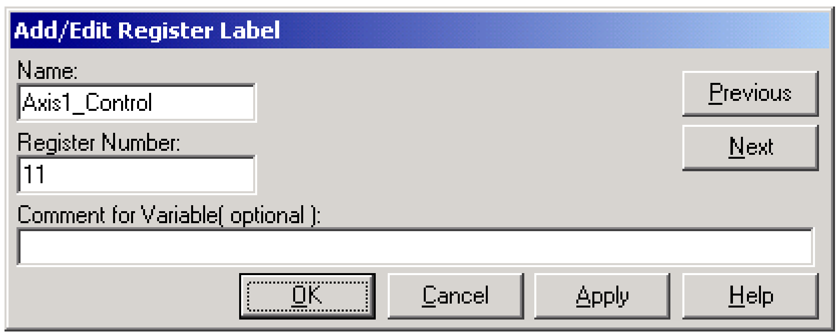


Fig. 10-50: "Add/Edit Register Label" Window

2. Enter the desired **Name**, **Register Number** and **Comment** (up to 80 characters).

3. Click **Apply** or **OK**. The **Previous** and **Next** buttons scroll through the defined labels.

Bit Labels

Selecting ***Edit*** ⇒ ***Edit Labels*** ⇒ ***Bit Labels*** opens the **Bit Labels** window, used to provide symbolic ASCII names for individual bits within Rexroth VisualMotion control and I/O registers.

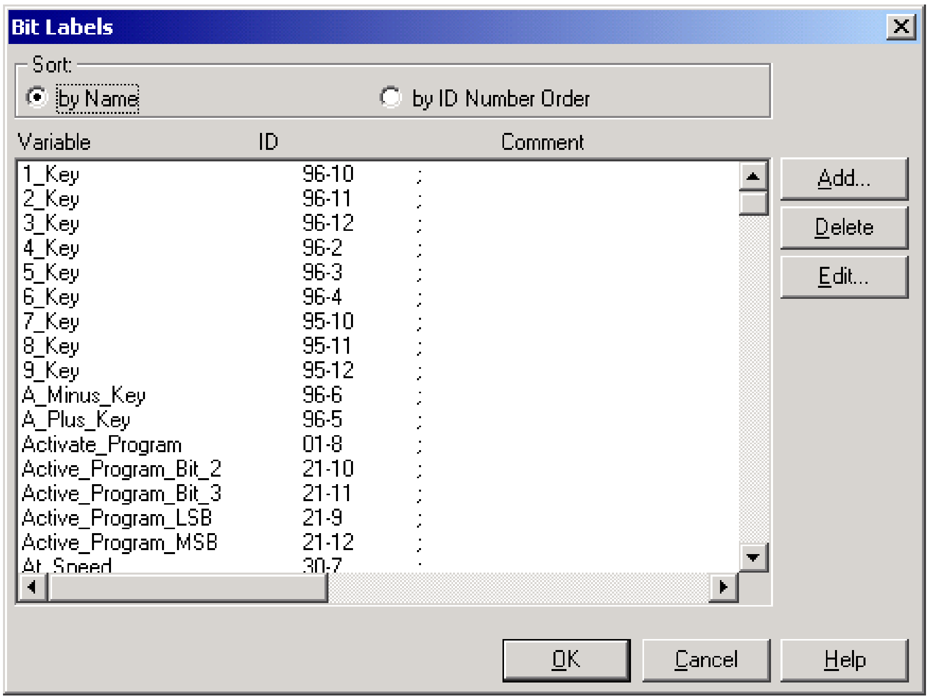


Fig. 10-51: "Bit Labels" Window

To sort the list alphabetically by label, select the radio box **by Name**. To sort the list alphabetically by ID, select the radio box **by ID Number Order**.

After bit labels are assigned and the program is saved, the labels are embedded in the motion program (the .str file) and will not be lost if the program is later transferred to a different Rexroth VisualMotion system. New programs are loaded with the default bit names.

To add or edit a bit label:

1. Click **Add…** or select the desired bit and click **Edit…** .

   An **Add/Edit Bit Label** window opens. The **Name** must be explicitly entered. (When editing an existing label, the current **Name**, **Reg-Bit Number** and **Comment** are automatically filled in.)

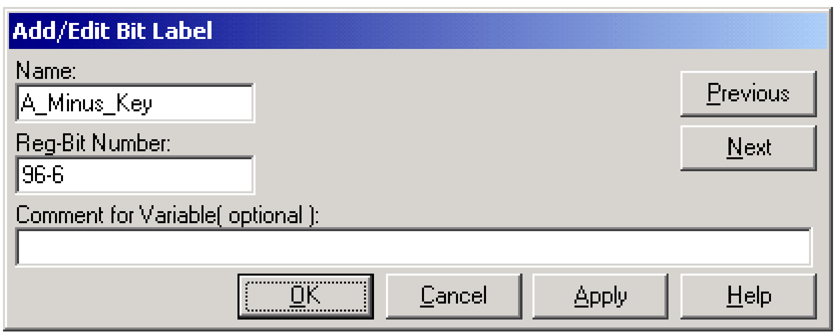


Fig. 10-52: "Add/Edit Bit Label" Window

Rexroth
Bosch Group

2. Enter the desired **Name**, **Reg-Bit Number** and **Comment** (up to 80 characters).
3. Click **Apply** or **OK**.
   The **Previous** and **Next** buttons scroll through the defined labels.

I/O Bit Function Labels

Selecting ***Edit*** ⇒ ***Edit Labels*** ⇒ ***I/O Bit Function Labels*** opens the **Global Integer Bit Labels** window, used to provide default ASCII names for I/O Mapper functions.

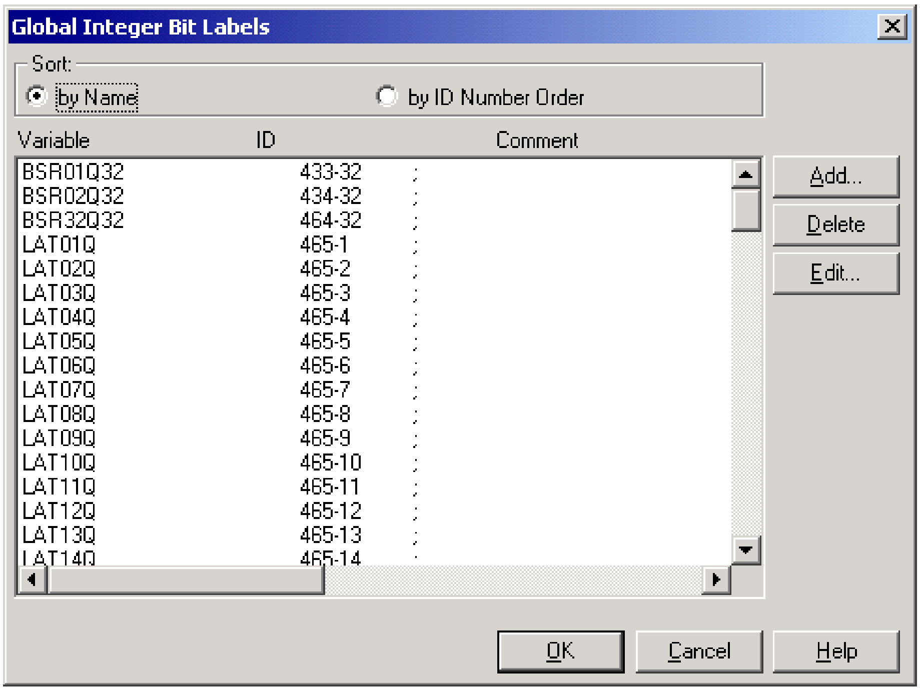


Fig. 10-53: “Global Integer Bit Labels” Window

To sort the list alphabetically by label, select the radio box **by Name**. To sort the list alphabetically by ID, select the radio box **by ID Number Order**.

After bit labels are assigned and the program is saved, the labels are embedded in the motion program (the .str file) and will not be lost if the program is later transferred to a different Rexroth VisualMotion system. New programs are loaded with the default bit names. Refer to chapter 4, Registers, for default bit names.

To add or edit a bit label:

1. Click **Add…** or select the desired bit and click **Edit…** .

   An **Add/Edit Bit Label** window opens. The **Name** must be explicitly entered. (When editing an existing label, the current **Name**, **Reg-Bit Number** and **Comment** are automatically filled in.)

2. Enter the desired **Name**, **Reg-Bit Number** and **Comment** (up to 80 characters).
3. Click **Apply** or **OK**.

10.9 File Formats

A project file contains all the data required for a system, including icon program files and their associated data files. A project may contain one of each of the following files:

VisualMotion Related File Extensions

File Extension	Description
*.ACC	text file that ACAM utility converts to a .csv file
*.CMD	Command file used when launching IndraLogic
*.CSV	comma-separated-variable type file used to store cam profiles
*.DAT	data type files for common data types such as event, points, floats, integers, zones, pls, etc.
*.EX1 – 9, *.EXA	archived program uploaded from the control displaying the order for each program slot
*.EXB	backup compiled project file that is uploaded from the control. It is ready to run and contains project data.
*.EXC	compiled project file that is downloaded to and executed by the control
*.EXP	Label file exported for use by IndraLogic as SysLibDirect variables
*.LSS	text files where VisualMotion stores register and bit labels used by the .str file
*.LST	text file that is referred to for registers and bit labels when the registers on the control are viewed
*.MAP	file used by the “Show Program Flow” function to trace the flow of the project while it is executing
*.PAR	storage format for all parameter files using the Sercos, ASCII (DriveTop compatible) format standard
*.POS	text file that PCAM utility converts to a .csv file
*.PVA	database parameter files. These files are used as offline databases
*.STR	icon program file
*.SYM	VM_%.sym is a symbol file used by VI Composer
*.TBL	text file of points created by the control's “Oscilloscope” function
*.TXT	text files used for functions such as Xferlog and Xxferlist
*.VEL	text file that PCAM utility converts to a .csv file
*.VMJ	VisualMotion project file
*.VTR	new variable file
*.VZP	compressed icon program for archiving on control

Table 10-10: VisualMotion Related Files

IndraLogic Related File Extensions

File Extension	Description
*.BIN	all retain data captured at the end of the PLC cycle
*.CI	Compile information for IndraLogic project
*.CHK	IndraLogic project code checksum
*.DAT	IndraLogic source file
*.LIB	IndraLogic target library files
*.LOG	IndraLogic project log file
*.PRG	IndraLogic boot project file containing all information required to load PLC program after booting
*.PRO	IndraLogic project file downloaded to the control
*.RI	IndraLogic download/reference information
*.SDB	Boot project symbol file of loaded boot project
*.SDK	IndraLogic symbol file
*.STS	Status file containing the state of the runtime system
*.SYM	IndraLogic symbol file
*.TCF	Trace configuration file
*.TNF	target information in binary format for all installed targets
*.TRG	target files in binary format for all installed targets

Table 10-11: IndraLogic Related Files

11 Error Reaction

11.1 Overview

Errors in the VisualMotion system can have different root causes like hardware and communication failure, operator input error or a crash in the machine which could obstruct the movement of an axis.

Note: This chapter only describes the error reactions associated with the control's motion functionality.

When an error occurs, the error type and associated control and drive parameters determine the system's error reaction. The basic function of the error reaction is for the control to determine which user task(s) A-D should be shutdown. Drives, associated with a user task that is shutdown, are stopped based on the error reaction of the drive. User tasks and drives can be parameterized to react differently to errors.

The following are some examples:

- synchronized stop of several axes in ELS or coordinated motion
- switching axes to torque free mode
- moving axes to a safe position

Errors in the VisualMotion system can originate from the control (system and tasks errors) or from the drives, as illustrated in the Fig. 11-1.

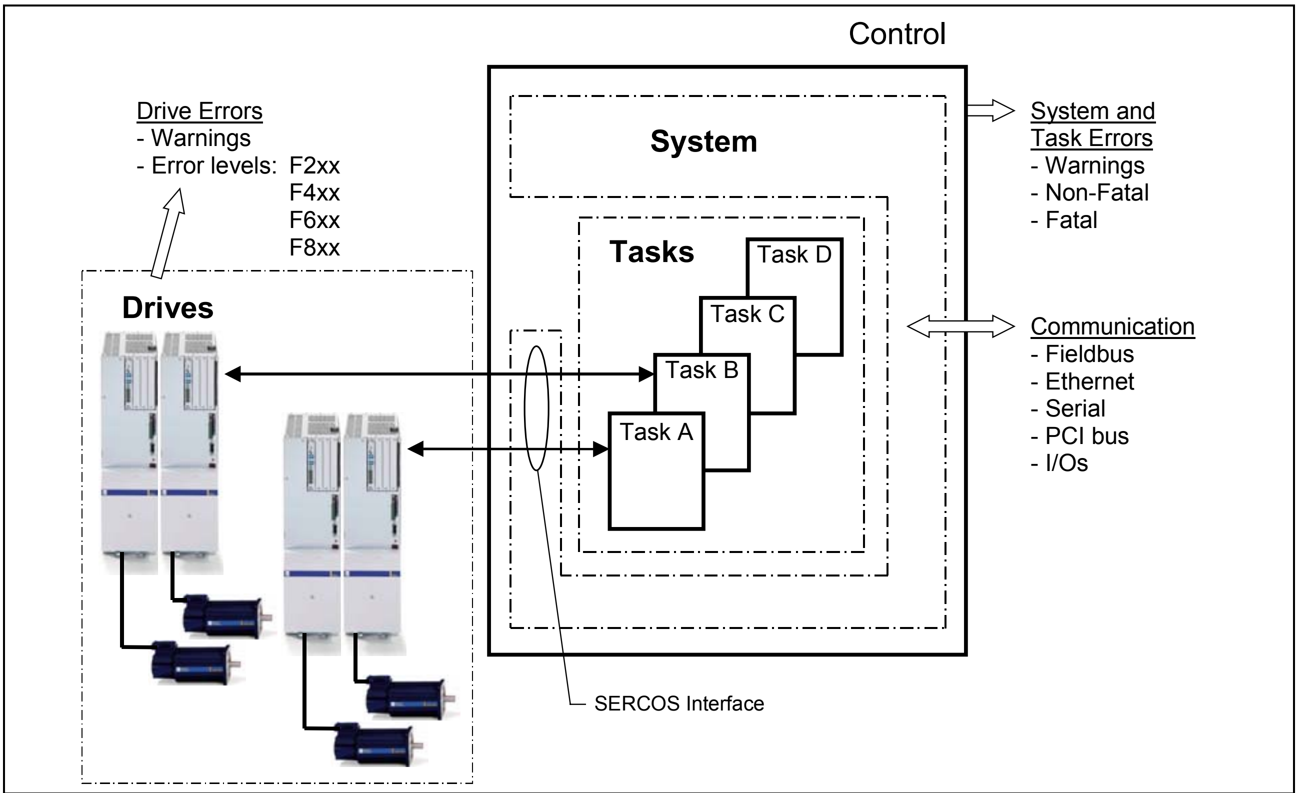


Fig. 11-1: System Overview

Error Types

VisualMotion's error system distinguishes between system, task, and drive errors and identifies them with diagnostic error codes.

System Errors

System errors affect all tasks that are not set to run during errors. These errors include for example, fieldbus errors, PLC interface errors or Option Card PLS errors.

Special Case System Error

When error "409 SERCOS Disconnect Error" occurs, the SERCOS communication stops and the control switches to SERCOS phase 0. Drives react to the loss of SERCOS communication by setting a F4xx interface error and stopping based on the error reaction of the drive.

User tasks that are configured to run during errors will continue to run. However, motion-related functionality such as the path planner for coordinated motion, or the ELS system stops when SERCOS communication is not in SERCOS phase 4.

Hardware and Fatal Operating System Errors

If the control stops operating because of a hardware failure or because of a fatal operating system error, the SERCOS communication will also stop. When a fatal operating system error occurs, the PPC-R displays two dots '**. .**' and the PPC-P11.1 displays a single dot '**.**' on the H1 display.

Task Errors

Task errors occur as the result of a user program instruction. These errors include for example, parameter transfer error, calculation instruction error, etc.. A task error sets bit 5 (Task_Error) in the relevant task status register.

Drive Errors and Warnings

A drive error sets bit 14 (Shutdown_Error) of the relevant axis status register and the control issues the error "420 Drive D Shutdown Error".

A drive warning sets bit 13 (Class_2_Warning) in the relevant axis status register. The default reaction of the control to a warning is to issue the error "498 Drive Shutdown Warning".

Drive warnings will be ignored by the control when bit 13 of C-0-0010 is set to 1.

Drive errors and warnings are associated with a task. A drive is associated with a task in the Axis or ELS Group icon.

Error Levels

The control distinguishes 3 levels of errors: fatal errors, non-fatal errors and warnings. The following flowchart shows how the control will respond to fatal and non-fatal errors:

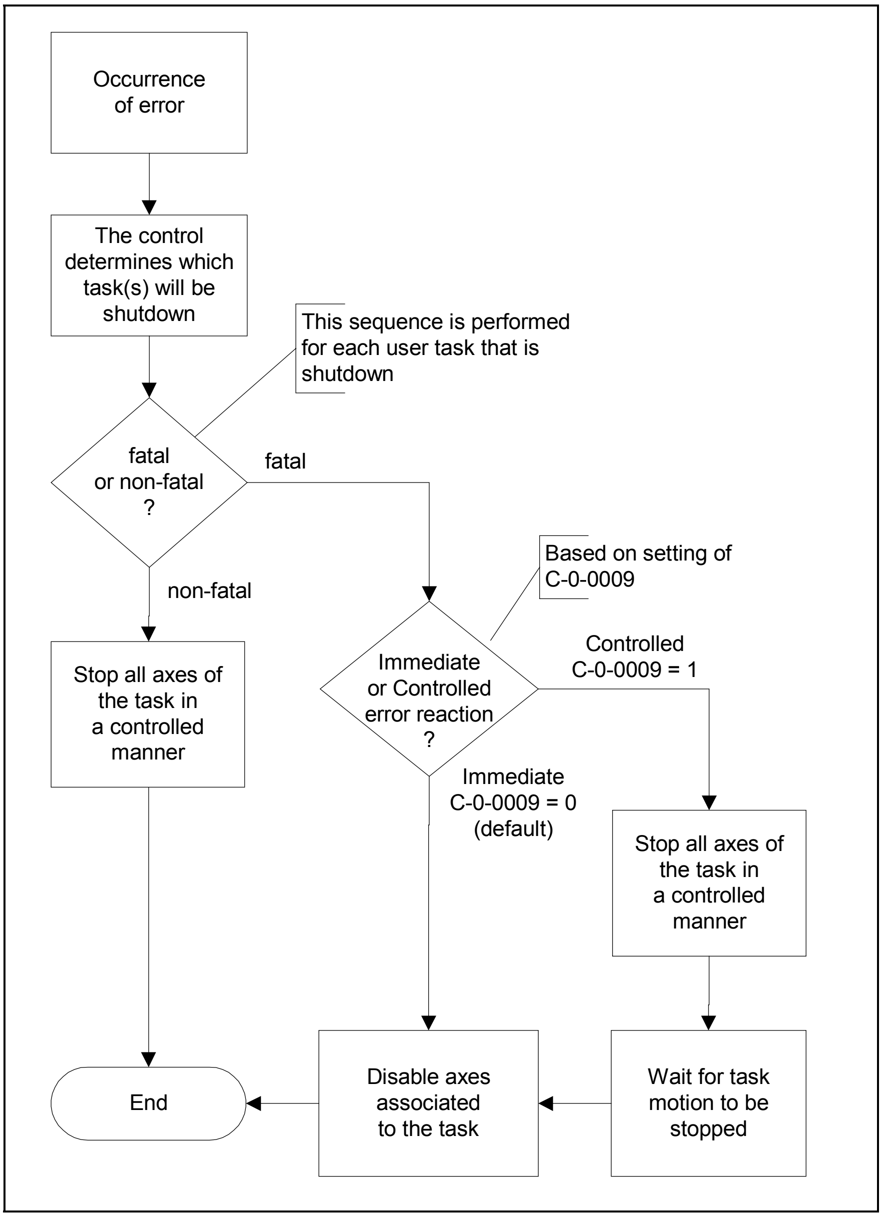


Fig. 11-2: Error Reaction Sequence

In the case of an error, moving axes can be disabled by the control and the error reaction of the drives can determine how the axes are stopped. Refer to *Drive Error Reaction* on page 11-12 for details.

Note: An axis can also be manually disabled by setting bit 1 of the relevant axis control register to 1.

Axes in ELS mode or coordinated motion mode can also be stopped in a controlled manner. This means that the ELS system master(s) or the coordinated motion path planner is ramped down to a standstill with the axes following in a synchronized/coordinated fashion. Refer to *Motion Type Error Reaction* on page 11-10 for details.

Rexroth Bosch Group

Fatal Errors

Fatal errors are considered the most severe. By default, the control reacts to fatal errors by shutting down the tasks and disabling the axes associated with the tasks when bit 1 of T-0-0002 is set to 0.

Some examples for fatal errors are:

- SERCOS disconnect error
- real-time operating system error (417 System Error)
- Emergency Stop (treated as a fatal error)

The error reaction for a fatal error depends on the parameterization of C-0-0009 (Error Reaction Mode).

Immediate Error Reaction for Fatal Errors (C-0-0009 = 0)

By default, all axes, regardless of their operating mode, are immediately disabled and stopped by the drive when C-0-0009 is set to 0. Refer to *Drive Error Reaction* on page 11-12 for details.

Virtual Masters in ELS mode and the path planner in coordinated motion mode are stopped instantaneously. Refer to *Motion Type Error Reaction* on page 11-10 for details.

Controlled Error Reaction for Fatal Errors (C-0-0009 = 1)

All motion comes to a controlled stop before the axes are disabled when C-0-0009 is set to 1. ELS axes stop synchronized and coordinated motion axes stop on path. Axes in single axis mode or velocity mode are switched to drive halt (AH). Refer to Table 11-4 and Table 11-5 for details.

Non-Fatal Errors

By default, the control reacts to non-fatal errors by stopping the ELS system and/or coordinated motion path planner in a controlled manner. Axes in single axis mode or velocity mode are switched to drive halt (AH). The task(s) are stopped, but the axes remain enabled when bit 1 of T-0-0002 is set to 0.

Some examples for non-fatal errors are:

- path planner errors
- program instruction errors (example: divide by zero in Calc icon)
- axis target position out of bounds

Warnings

A warning does not affect tasks and axes. User tasks remain in operation and all associated axes remain enabled. It is up to the user program to respond to warnings. For certain errors, the control can be configured to respond with an error or a warning, for example:

- fieldbus
- slip monitoring

Configurable Error Reaction

The following functions have a configurable error reaction. The following table lists the different settings:

Function	Parameter	Setting	Associated Error Code
RECO I/O Error Reaction	C-0-0010, bits 6-7	bit 6 bit 7 0 0 = ignore 1 0 = warning x 1 = fatal error (default)	544 (fatal error) 215 (warning)
Response to a drive warning	C-0-0010, bit 13	0 = drive warning issues an error (default) 1 = ignore drive warnings	498 (non-fatal error)
Axis	C-0-0010, bit 14	0 = error if axis is not ready (default) 1 = no error is issued if axis is not ready	426 (non-fatal error) fatal error for coord. motion
	A-0-0006, bit 1	0 = ignore (default) 1 = issue error when drive is enabled but not referenced	500 (non-fatal error)
Fieldbus Interface	C-0-2635	In case of fieldbus errors 0x0000 = shutdown (default) 0x0001 = warning 0x0002 = ignore	519 (fatal error) 520 (fatal error) 208 (warning) 209 (warning)
Position Monitoring Groups	C-0-32x6, bit 2	0 = warning (default) 1 = fatal error	554 (fatal error) 220 (warning)
Parameter Transfer	T-0-0002, bit 3	0 = non-fatal error (default) 1 = warning	422 (non-fatal error) 205 (warning)
ELS Slip Monitoring	ELS_MSTR_CONFIG, bit 32	0 = fatal error (default) 1 = warning	552 (fatal error) 221 (warning)

Table 11-1: Functions with Configurable Error Reaction

Error Diagnostics and Logging

VisualMotion errors, which are written to C-0-0122, are also stored in the diagnostic log list parameter C-0-2020.

Diagnostic Parameters

Errors are identified by a specific error code number. Some similar errors share the same error code. For these errors, additional information is provided as an extended diagnostic message.

The following parameters are used for diagnostic messages, error codes, and extended diagnostic messages:

Parameter	Description
C-0-0122	Current diagnostic message (displayed on control's H1 display)
C-0-0123	Current diagnostic code
C-0-0124	Extended diagnostic message
T-0-0122	Task diagnostic message
C-0-2020	Diagnostic log

Table 11-2: Diagnostic Parameters

Refer to the *VisualMotion 11 Trouble Shooting Guide* for a complete listing of error codes.

Rexroth Bosch Group

Accessing Error Codes

Diagnostic parameters and registers can be accessed via the serial, Ethernet or PCI interface using any of the following devices:

- VisualMotion Toolkit
- HMI
- PLC

Diagnostic Status Registers

VisualMotion provides the following status register bits to indicate that an error or warning has occurred. These read-only status register bits are set to 1, when an error or warning condition is present.

Register	Bit	Description
021: System Status	5	*Error* - set when any error or warning is present in the system
	6	*Error Active* - set when the current diagnostic is a fatal or non-fatal error
	7	*Warning Active* - set when the current diagnostic is a warning
022-025: Task Status	5	*Task Error* - set when the current diagnostic is a task error
031-038: Axis 1-8 Status 309-340: Axis 9-40 Status	13	Class 2 Warning
	14	Drive Shutdown Error

Table 11-3: Diagnostic Status Registers

Diagnostic Log

Errors are listed in the diagnostic log in the order in which they occurred. The most recent error is listed first. The diagnostic log can contain up to 100 of the most recent errors.

Diagnostic Priority

A set of rules governs the priority with which diagnostic messages are displayed and logged. A non-fatal error will overwrite a warning and a fatal error will overwrite a non-fatal error or a warning.

11.2 Task Association and Setup

Axes and path planner are associated with user tasks in the Axis or ELS Group icon.

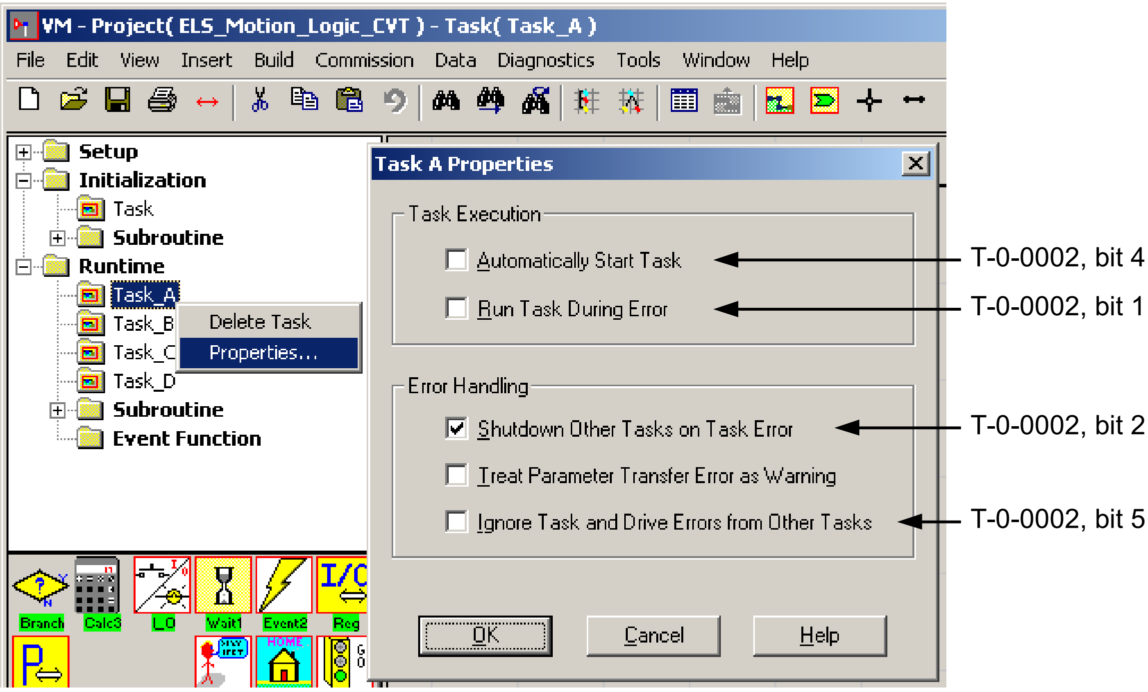
The error reaction for a given operating mode, such as single axis, or coordinated motion is processed through the associated user task. The ELS system is the only exception. The Virtual Master(s) and ELS Groups are associated with task A. If task A stops running, due to an error, so will the ELS system. The ELS slave axes can be associated with a different task than that of the ELS system. The association of an axis to a user task is performed in the Axis or ELS Group icons.

Task Error Reaction Setup

Each task in a VisualMotion project can be configured to react differently to errors. The programmer must determine which tasks should run and which tasks should be stopped when an error occurs.

Task Options

Task options are set in bits 1, 2, and 5 of task parameter T-0-0002. These bits can be set in VisualMotion Toolkit by right clicking on the desired task letter in the Project Navigator window and selecting ***Properties...***. The system must in SERCOS phase 2 (parameter mode) for changes to take effect.



TaskProperties.tif

Fig. 11-3: Task Properties

Default Task Error Reaction

When bits 1, 2, and 5 of T-0-0002 are set to 0, any error that occurs will shutdown the task. For Fig. 11-4 to Fig. 11-7, the "Other Tasks" are configured with the default task error reaction.

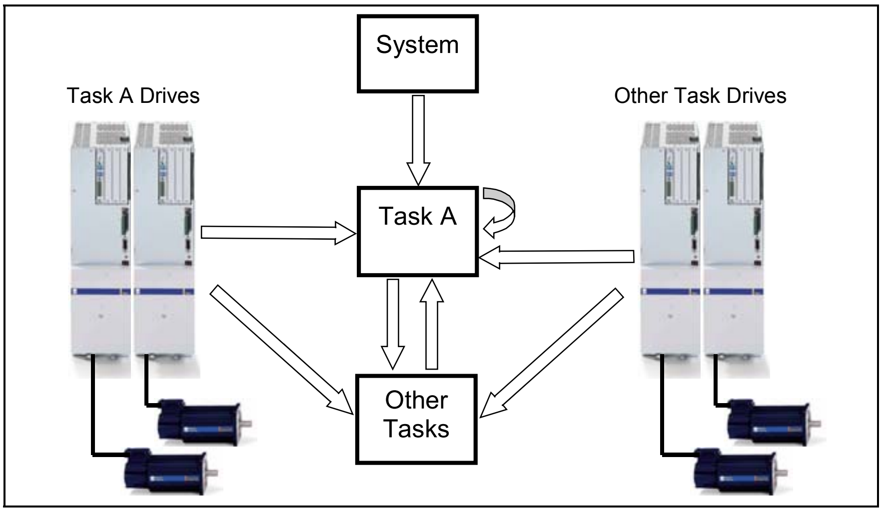


Fig. 11-4: Default Task Error Reaction

Run Task During Error (T-0-0002, Bit 1)

In Fig. 11-5, when bit 1 of T-0-0002 is set to 1, and bits 2 and 5 are set to 0, task A will run during errors unless an error occurs within task A. System errors, drive errors, and task errors of other tasks (B, C, D) will not shutdown task A. However, task A can shutdown other tasks.

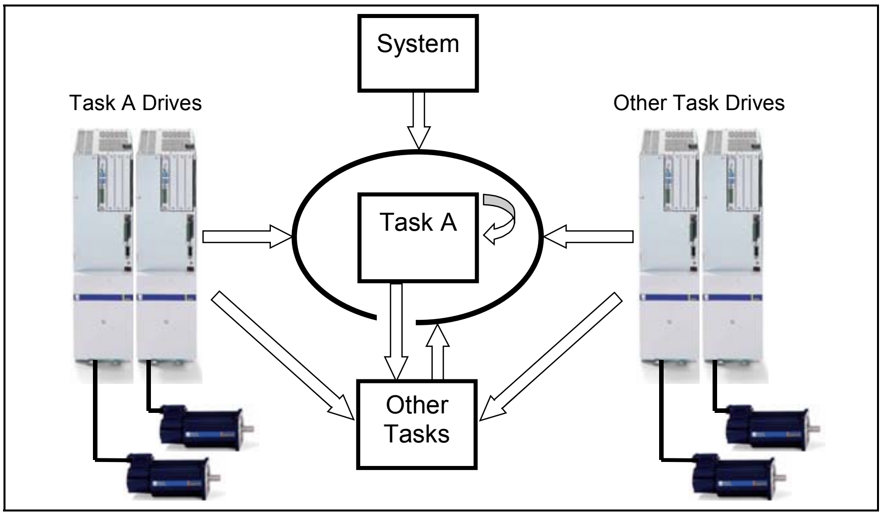


Fig. 11-5: Run Task During Error (T-0-0002, Bit 1)

Shutdown Other Tasks on Task Error (T-0-0002, Bit 2)

In Fig. 11-6, when bit 2 of T-0-0002 is set to 1 and bits 1 and 5 are set to 0, errors that occur within task A do not shutdown other tasks. System errors, drive errors, errors within task A, and task errors from other tasks will shutdown task A.

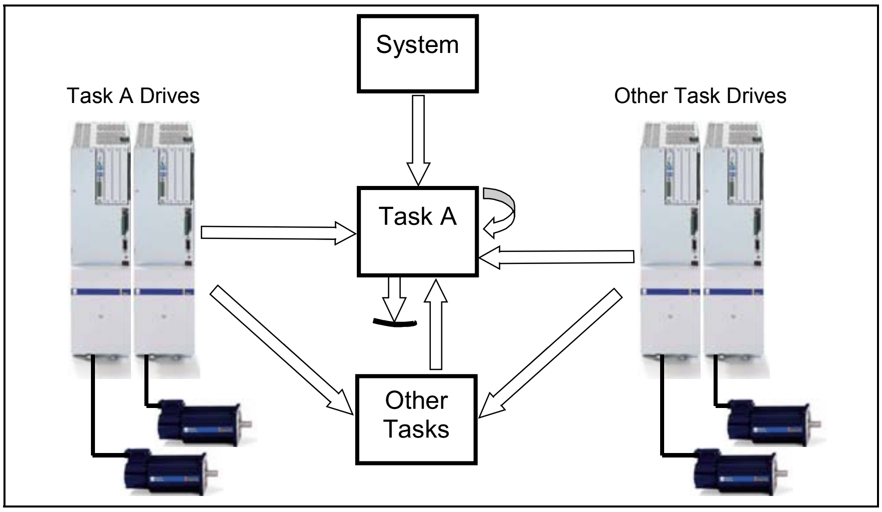


Fig. 11-6: Shutdown Other Tasks on Task Error (T-0-0002, Bit 2)

Ignore Task and Drive Errors from Other Tasks (T-0-0002, Bit 5)

In Fig. 11-7, when bit 5 of T-0-0002 is set to 1 and bits 1 and 2 are set to 0, task A will run during task errors that occur in other tasks, as well as drive errors from drives associated with other tasks. System errors, associated drive errors, and errors within task A will shutdown task A. However, task A can shutdown other tasks.

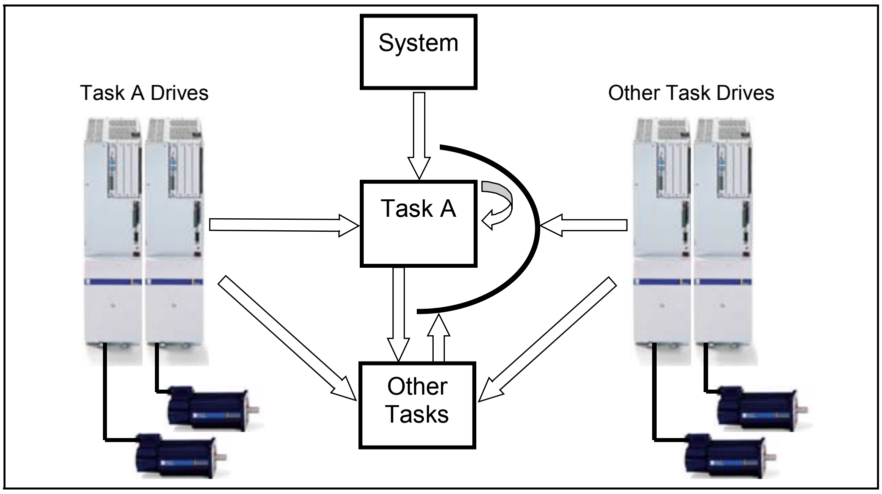


Fig. 11-7: Ignore Task and Drive Errors from Other Tasks

Task Shutdown Sequence

When a fatal or non-fatal error occurs, the control determines if the error is a system, task, or drive error. As the next step, the settings in task parameter T-0-0002 are checked to determine if the task should be shutdown. Axes, associated with the task that is shutdown, are stopped based on whether the error is fatal or non-fatal, their current mode of operation, and the parameterization of the error reaction of the drives. Refer to *Motion Type Error Reaction* on page 11-10 for details.

11.3 Motion Type Error Reaction

The error reaction varies based on the following factors:

- the current operating mode of the axis
- whether the error is non-fatal or fatal
- and the setting of control parameter C-0-0009

The following table lists the error reaction for non-fatal and fatal errors based on operating mode.

Operating Mode	**Non-Fatal Error**	**Fatal Error** Immediate (C-0-0009 = 0)	**Fatal Error** Controlled (C-0-0009 = 1)	**User Task Association**
Torque Mode	No error reaction	Axis is disabled immediately **	Axis is disabled after all task motion has been stopped	Configured in Axis Icon
Velocity Mode	Axis is switched to drive halt *	Axis is disabled immediately **	Axis is switched to drive halt *	Configured in Axis Icon
Single Axis Mode	Axis is switched to drive halt *	Axis is disabled immediately **	Axis is switched to drive halt *	Configured in Axis Icon
Coordinated Motion	Follows path planner The axis remains enabled	Axis is disabled immediately **	Follows path planner The axis is disabled after all task motion has been stopped	Configured in Axis Icon
Path Planner	Decelerates using maximum deceleration in T-0-0022	Stops instantaneously	Decelerates using maximum deceleration in T-0-0022	Configured in Axis Icon
Virtual Master	Decelerates using float VM#_CMD_DECEL	Stops instantaneously	Decelerates using float VM#_E_STOP_DECEL	Task A
ELS Group	If Virtual Master is the current master, than it continues to follow it to a stop, otherwise it switches to local mode and decelerates using G#_STOP_DECEL	If the Virtual Master is the current master, than it stops instantaneously, otherwise it switches to local mode and decelerates using G#_STOP_DECEL	If Virtual Master is the current master, than it continues to follow it to a stop, otherwise it switches to local mode and decelerates using G#_STOP_DECEL	Task A
Drive ELS	Follows ELS Group The axis remains enabled	Axis is disabled immediately **	Follows ELS Group, then The axis is disabled after all task motion has been stopped	Configured in Axis Icon
Control CAM	Follows ELS Group The axis remains enabled	Axis is disabled immediately **	Follows ELS Group The axis is disabled after all task motion has been stopped	Configured in Axis Icon
Ratio	Follows master or external axis The axis remains enabled	Axis is disabled immediately **	Follows master or external axis, The axis is disabled after all task motion has been stopped	Configured in Axis Icon
Coordinated Articulation	Follows ELS Group The axis remains enabled	Axis is disabled immediately **	Follows ELS Group, then after all task motion has stopped, the axis is disabled	Configured in Axis Icon

* Refer to *Drive Halt Setup (AH)* on page 11-13 for details.

** See parameterization of A-0-0004 and P-0-0119

Table 11-4: Motion Type Error Reaction

Special Case Error Reactions for the ELS system

A special case is the error reaction for when an ELS slave axis is associated with tasks B, C or D. The Virtual Masters and ELS Groups are always associated with task A.

ELS Slave Axis Task Shutdown

The following error reaction occurs when a task, with an associated ELS slave axis, shuts down but task A continues to run:

- In the case of a non-fatal error, the ELS slave axis remains in its operation mode and follows its ELS Group
- In the case of a fatal error, the ELS slave axis is disabled and brought to a stop using the drive's error reaction

ELS Group Task Shutdown

The following error reaction occurs when task A shuts down, but the task with an associated ELS slave axis continues to run:

- The ELS slave axis remains in its operating mode and follows its ELS Group to a stop

Link Ring Error Reaction

The default error reaction for an ELS Group following a Link Ring master is to switch to Local Mode. The ELS Group decelerate to a stop using the rate in program variable G#_STOP_DECEL.

If this is not the desired error reaction, task A can be set to run during errors and the user can program an error reaction. Therefore, the user program should monitor the Node_#_Data_Valid bits in status registers 41 and 42 and run a custom error reaction when the relevant bit goes low.

For example, an ELS Group could perform an "On-the-Fly" switch to a free Virtual Master. The Virtual Master could then move the ELS Group to a safe position.

When a Link Ring master position is used as the input of an ELS Group, the validity of the Link Ring master position is monitored continuously by the control. Error “575 ELS master for ELS Group # is invalid” occurs when the current ELS master is considered to be invalid. The Link Ring master is only updated and valid when the control, providing the Link Ring master position, is in SERCOS phase 4.

Rexroth Bosch Group

11.4 Drive Error Reaction

The error reaction of the drive depends on the error level and the settings in the following drive parameters:

- P-0-0117, NC reaction on error
- P-0-0119, Best possible deceleration

Error Level	Diagnostic Code	Drive Response
Fatal	F8 xx	Drive is switched torque free. The parameterization of P-0-0117 and P-0-0119 is ignored because a drive reaction is impossible.
Travel Range	F6 xx	The velocity command value is immediately set to zero, regardless of the parameterization of P-0-0117 and P-0-0119.
Interface	F4 xx	The drive stops as defined in P-0-0119. A response from the control is impossible, since communication has stopped.
Non-Fatal	F3 xx F2 xx	P-0-0117, Bit 0 = 0, the drive stops as defined in P-0-0119. P-0-0117, Bit 0 = 1, the drive continues to follow the commands of the control for 30 seconds after the error occurred. The control then brings the axis to a controlled stop. Afterwards, the drive stops as parameterized in P-0-0119.

Table 11-5: Drive Error Levels

Power Supply Error Reaction

For applications where a modular drive concept is used (e.g., DIAX04), the user has a choice of how to configure the response of the power supply in the event of a drive error. In the case of an error and default configuration, a signal is sent to the power supply and the power supply removes the bus power from all drives. The user has the option to suppress or delay this in bits 0 and 2 of P-0-0118 (Power off on error). Refer to the relevant *Digital Drive* manual for details.

Disable Axis Command (Ab)

An axis is disabled by the control in the event of a fatal error. The default reaction (A-0-0004, bit 12 = 0) is to stop according to the parameterization of P-0-0119 and then the drive will remove torque from the motor.

In the case of a fatal error, torque is immediately removed from the motor and it will coast to a stop when bit 12 of A-0-0004 is set to 1.

An axis is manually disabled when bit 1 of the relevant axis control register is set to 1.

The following axis and drive parameters determine how the drive will react when the axis is disabled.

Parameter	Description
A-0-0004	bit 12: 0 = Stop axis according to P-0-0119 (Default) 1 = Torque Free
P-0-0119	<u>value</u> 0 Velocity command value set to zero 1 Switch to torque-free state 2 Velocity command to zero with command ramp and filter 3 Return motion

Table 11-6: Drive Disable Method

Drive Halt Setup (AH)

When the drive issues a drive halt, the axis is brought to a stop using a predefined deceleration rate. The deceleration rate that is used varies based on the active operating mode of the axis and the drive firmware being used.

The following table lists the firmware/operating mode combinations:

Drive (Firmware)	Operating Mode	Drive Halt Deceleration Rate
ECODrive (SGP03) ECODrive Cs (MGP01)	Single Axis	S-0-0359 (positioning deceleration) S-0-0260 (positioning acceleration) is used when S-0-0359 = 0
	Velocity	The following conditions apply: • P-0-1211 (deceleration ramp 1) is used when the current velocity is less than P-0-1202 (final speed of ramp 1). If P-0-1211 = 0, P-0-1201(ramp 1 pitch) is used • P-0-1213 (deceleration ramp 2) is used when the current velocity is greater than P-0-1202 (final speed of ramp 1). If P-0-1213 = 0, P-0-1203 (ramp 2 pitch) is used Full torque is used if the active ramp pitch = 0
	All Others	S-0-0138 (bipolar acceleration limit value)
ECODrive (SGP01 or SMT02)	Single Axis	S-0-0260 (positioning Acceleration)
	Velocity	The following conditions apply: • P-0-1201 (ramp 1 pitch) is used when the current velocity is less than P-0-1202 • P-0-1203 (ramp 2 pitch) is used when the current velocity is greater than P-0-1202 Full torque is used if the active ramp pitch = 0
	All Others	S-0-0138 (bipolar acceleration limit value)
DIAX04 (ELS05)	All Modes	S-0-0138 (bipolar acceleration limit value)
DIAX04 (SSE03)	Single Axis	S-0-0260 (positioning acceleration)
	Velocity	P-0-1201 (velocity control, deceleration ramp 1)
	All Others	S-0-0138 (bipolar acceleration limit value)
IndraDrive	Single Axis	S-0-0349 (jerk limit bipolar) S-0-0372 (Drive Halt acceleration bipolar)
	Velocity	S-0-0372 (Drive Halt acceleration bipolar)
	All Others	S-0-0138 (bipolar acceleration limit value)

Table 11-7: Drive Halt Deceleration Rate based on Firmware/Operation Mode

Index

Rexroth
Bosch Group

P

R

S

T

U

V

W

Z

Service & Support

Helpdesk

Unser Kundendienst-Helpdesk im Hauptwerk Lohr am Main steht Ihnen mit Rat und Tat zur Seite. Sie erreichen uns

Our service helpdesk at our headquarters in Lohr am Main, Germany can assist you in all kinds of inquiries. Contact us

- telefonisch - by phone: **+49 (0) 9352 40 50 60**
  über Service Call Entry Center Mo-Fr 07:00-18:00 Central European Time
  - via Service Call Entry Center Mo-Fr 7:00 am - 6:00 pm CET
- per Fax - by fax: **+49 (0) 9352 40 49 41**
- per e-Mail - by e-mail: **service.svc@boschrexroth.de**

Service-Hotline

Außerhalb der Helpdesk-Zeiten ist der Service Deutschland direkt ansprechbar unter

After helpdesk hours, contact the German service experts directly at

+49 (0) 171 333 88 26

oder - or **+49 (0) 172 660 04 06**

Hotline-Rufnummern anderer Länder entnehmen Sie bitte den Adressen in den jeweiligen Regionen.

Hotline numbers of other countries to be seen in the addresses of each region.

Internet

Unter **www.boschrexroth.com** finden Sie ergänzende Hinweise zu Service, Reparatur und Training sowie die **aktuellen** Adressen *) unserer auf den folgenden Seiten aufgeführten Vertriebs- und Servicebüros.

☐ Verkaufsniederlassungen
☐ Niederlassungen mit Kundendienst

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit unserem für Sie nächstgelegenen Ansprechpartner auf.

*) Die Angaben in der vorliegenden Dokumentation können seit Drucklegung überholt sein.

At **www.boschrexroth.com** you can find additional notes about service, repairs and training. The current addresses *) for our sales and service locations around the world are on the following pages.

☐ Sales agencies
☐ Sales agencies providing service

Please contact our sales / service office in your area first.

*) Data in the present documentation may have become obsolete since printing.

Vor der Kontaktaufnahme... - Before contacting us...

Wir können Ihnen schnell und effizient helfen wenn Sie folgende Informationen bereithalten:

1. Detaillierte Beschreibung der Störung und der Umstände.
2. Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern.
3. Tel.-/Faxnummern und e-Mail-Adresse, unter denen Sie für Rückfragen zu erreichen sind.

For quick and efficient help, please have the following information ready:

1. Detailed description of the failure and circumstances.
2. Information on the nameplate of the affected products, especially typecodes and serial numbers.
3. Your phone/fax numbers and e-mail address, so we can contact you in case of questions.

Kundenbetreuungsstellen - Sales & Service Facilities

Deutschland – Germany

vom Ausland: (0) nach Landeskennziffer weglassen!
from abroad: don't dial (0) after country code!

Vertriebsgebiet Mitte Germany Centre Bosch Rexroth Electrice Drives and Controls GmbH Bgm.-Dr.-Nebel-Str. 2 / Postf. 1357 97816 Lohr am Main / 97803 Lohr **Kompetenz-Zentrum Europa** Tel.: +49 (0)9352 40-0 Fax: +49 (0)9352 40-4885	SERVICE AUTOMATION CALL ENTRY CENTER **Helpdesk** **MO – FR** **von 07:00 - 18:00 Uhr** **from 7 am – 6 pm** **Tel. +49 (0) 9352 40 50 60** **Fax +49 (0) 9352 40 49 41** service.svc@boschrexroth.de	SERVICE AUTOMATION **HOTLINE 24 / 7 / 365** **außerhalb der Helpdesk-Zeit** **out of helpdesk hours** **Tel.: +49 (0)172 660 04 06** oder / or **Tel.: +49 (0)171 333 88 26**	SERVICE AUTOMATION **ERSATZTEILE / SPARES** verlängerte Ansprechzeit - extended office time - ♦ nur an Werktagen - only on working days - ♦ von 07:00 - 18:00 Uhr - from 7 am - 6 pm - **Tel. +49 (0) 9352 40 42 22**
Vertriebsgebiet Süd Germany South Bosch Rexroth AG Landshuter Allee 8-10 80637 München Tel.: +49 (0)89 127 14-0 Fax: +49 (0)89 127 14-490	Vertriebsgebiet West Germany West Bosch Rexroth AG Regionalzentrum West Borsigstrasse 15 40880 Ratingen Tel.: +49 (0)2102 409-0 Fax: +49 (0)2102 409-406 +49 (0)2102 409-430	Gebiet Südwest Germany South-West Bosch Rexroth AG Service-Regionalzentrum Süd-West Siemensstr. 1 70736 Fellbach Tel.: +49 (0)711 51046–0 Fax: +49 (0)711 51046–248	
Vertriebsgebiet Nord Germany North Bosch Rexroth AG Walsroder Str. 93 30853 Langenhagen Tel.: +49 (0) 511 72 66 57-0 Service: +49 (0) 511 72 66 57-256 Fax: +49 (0) 511 72 66 57-93 Service: +49 (0) 511 72 66 57-783	Vertriebsgebiet Mitte Germany Centre Bosch Rexroth AG Regionalzentrum Mitte Waldecker Straße 13 64546 Mörfelden-Walldorf Tel.: +49 (0) 61 05 702-3 Fax: +49 (0) 61 05 702-444	Vertriebsgebiet Ost Germany East Bosch Rexroth AG Beckerstraße 31 09120 Chemnitz Tel.: +49 (0)371 35 55-0 Fax: +49 (0)371 35 55-333	Vertriebsgebiet Ost Germany East Bosch Rexroth AG Regionalzentrum Ost Walter-Köhn-Str. 4d 04356 Leipzig Tel.: +49 (0)341 25 61-0 Fax: +49 (0)341 25 61-111

Europa (West) - Europe (West)

vom Ausland: (0) nach Landeskennziffer weglassen, Italien: 0 nach Landeskennziffer mitwählen
from abroad: don't dial (0) after country code, Italy: dial 0 after country code

Austria - Österreich Bosch Rexroth GmbH Electric Drives & Controls Stachegasse 13 1120 Wien Tel.: +43 (0) 1 985 25 40 Fax: +43 (0) 1 985 25 40-1459	Austria – Österreich Bosch Rexroth GmbH Electric Drives & Controls Industriepark 18 4061 Pasching Tel.: +43 (0)7221 605-0 Fax: +43 (0)7221 605-1220	Belgium - Belgien Bosch Rexroth NV/SA Henri Genessestraat 1 1070 Bruxelles Tel: +32 (0) 2 451 26 08 Fax: +32 (0) 2 451 27 90 info@boschrexroth.be service@boschrexroth.be	Denmark - Dänemark BEC A/S Zinkvej 6 8900 Randers Tel.: +45 87 11 90 60 Fax: +45 87 11 90 61
Denmark - Dänemark Bosch Rexroth A/S Engelsholmvej 26 8900 Randers Tel.: +45 36 77 44 66 Fax: +45 70 10 03 20 tj@boschrexroth.dk	Great Britain – Großbritannien Bosch Rexroth Ltd. Electric Drives & Controls Broadway Lane, South Cerney Cirencester, Glos GL7 5UH Tel.: +44 (0)1285 863-000 Fax: +44 (0)1285 863-030 sales@boschrexroth.co.uk service@boschrexroth.co.uk	Finland - Finnland Bosch Rexroth Oy Electric Drives & Controls Ansatie 6 017 40 Vantaa Tel.: +358 (0)9 84 91-11 Fax: +358 (0)9 84 91-13 60	France - Frankreich Bosch Rexroth SAS Electric Drives & Controls Avenue de la Trentaine (BP. 74) 77503 Chelles Cedex Tel.: +33 (0)164 72-63 22 Fax: +33 (0)164 72-63 20 **Hotline: +33 (0)608 33 43 28**
France - Frankreich Bosch Rexroth SAS Electric Drives & Controls ZI de Thibaud, 20 bd. Thibaud (BP. 1751) 31084 Toulouse Tel.: +33 (0)5 61 43 61 87 Fax: +33 (0)5 61 43 94 12	France – Frankreich Bosch Rexroth SAS Electric Drives & Controls 91, Bd. Irène Joliot-Curie 69634 Vénissieux – Cedex Tel.: +33 (0)4 78 78 53 65 Fax: +33 (0)4 78 78 53 62	France – Frankreich **Tightening & Press-fit:** Globe Techniques Nouvelles 143, Av. du Général de Gaulle 92252 La Garenne Colombes Tel.: +33 (0)1 41 19 33 33	Italy - Italien Bosch Rexroth S.p.A. Strada Statale Padana Superiore 11, no. 41 20063 Cernusco S/N.MI **Hotline: +39 02 92 365 563** Tel.: +39 02 92 365 1 Service: +39 02 92 365 300 Fax: +39 02 92 365 500 Service: +39 02 92 365 516
Italy - Italien Bosch Rexroth S.p.A. Via Paolo Veronesi, 250 10148 Torino Tel.: +39 011 224 88 11 Fax: +39 011 224 88 30	Italy - Italien Bosch Rexroth S.p.A. Via Mascia, 1 80053 Castellamare di Stabia NA Tel.: +39 081 8 71 57 00 Fax: +39 081 8 71 68 85	Italy - Italien Bosch Rexroth S.p.A. Via del Progresso, 16 (Zona Ind.) 35020 Padova Tel.: +39 049 8 70 13 70 Fax: +39 049 8 70 13 77	Italy - Italien Bosch Rexroth S.p.A. Via Isonzo, 61 40033 Casalecchio di Reno (Bo) Tel.: +39 051 29 86 430 Fax: +39 051 29 86 490
Italy - Italien **Tightening & Press-fit:** TEMA S.p.A. Automazione Via Juker, 28 20025 Legnano Tel.: +39 0 331 4671	Netherlands – Niederlande/Holland Bosch Rexroth B.V. Kruisbroeksestraat 1 (P.O. Box 32) 5281 RV Boxtel Tel.: +31 (0) 411 65 16 40 Fax: +31 (0) 411 65 14 83 www.boschrexroth.nl	Netherlands - Niederlande/Holland Bosch Rexroth Services B.V. Technical Services Kruisbroeksestraat 1 (P.O. Box 32) 5281 RV Boxtel Tel.: +31 (0) 411 65 19 51 Fax: +31 (0) 411 67 78 14 **Hotline: +31 (0) 411 65 19 51** services@boschrexroth.nl	Norway - Norwegen Bosch Rexroth AS Electric Drives & Controls Berghagan 1 or: Box 3007 1405 Ski-Langhus 1402 Ski Tel.: +47 64 86 41 00 Fax: +47 64 86 90 62 **Hotline: +47 64 86 94 82** jul.ruud@rexroth.no
Spain – Spanien Goimendi Automation S.L. Parque Empresarial Zuatzu C/ Francisco Grandmontagne no.2 20018 San Sebastian Tel.: +34 9 43 31 84 21 - service: +34 9 43 31 84 56 Fax: +34 9 43 31 84 27 - service: +34 9 43 31 84 60 sat.indramat@goimendi.es	Spain - Spanien Bosch Rexroth S.A. Electric Drives & Controls Centro Industrial Santiga Obradors 14-16 08130 Santa Perpetua de Mogoda Barcelona Tel.: +34 9 37 47 94-00 Fax: +34 9 37 47 94-01	Spain - Spanien Bosch Rexroth S.A. Electric Drives & Controls c/ Almazara, 9 28760 Tres Cantos (Madrid) Tel.: +34 91 806 24 79 Fax: +34 91 806 24 72 fernando.bariego@boschrexroth.es	Sweden - Schweden Bosch Rexroth AB Electric Drives & Controls - Varuvägen 7 (Service: Konsumentvägen 4, Älfsjö) 125 81 Stockholm Tel.: +46 (0) 8 727 92 00 Fax: +46 (0) 8 647 32 77
Sweden - Schweden Bosch Rexroth AB Electric Drives & Controls Ekvändan 7 254 67 Helsingborg Tel.: +46 (0) 4 238 88 -50 Fax: +46 (0) 4 238 88 -74	Switzerland East - Schweiz Ost Bosch Rexroth Schweiz AG Electric Drives & Controls Hemrietstrasse 2 8863 Buttikon Tel. +41 (0) 55 46 46 111 Fax +41 (0) 55 46 46 222	Switzerland West - Schweiz West Bosch Rexroth Suisse SA Av. Général Guisan 26 1800 Vevey 1 Tel.: +41 (0)21 632 84 20 Fax: +41 (0)21 632 84 21	

Rexroth
Bosch Group

Europa (Ost) - Europe (East)

vom Ausland: (0) nach Landeskennziffer weglassen
from abroad: don't dial (0) after country code

<table>
<tr><td>Czech Republic - Tschechien
Bosch -Rexroth, spol.s.r.o.
Hviezdoslavova 5
627 00 Brno
Tel.: +420 (0)5 48 126 358
Fax: +420 (0)5 48 126 112</td><td>Czech Republic - Tschechien
Tightening & Press-fit:
Bosch -Rexroth, spol.s.r.o.
Stetkova 18
140 68 Praha 4
Tel.: +420 (0)241 406 675</td><td>Hungary - Ungarn
Bosch Rexroth Kft.
Angol utca 34
1149 Budapest
Tel.: +36 (1) 422 3200
Fax: +36 (1) 422 3201</td><td>Poland – Polen
Bosch Rexroth Sp.zo.o.
ul. Staszica 1
05-800 Pruszków
Tel.: +48 (0) 22 738 18 00
– service: +48 (0) 22 738 18 46
Fax: +48 (0) 22 758 87 35
– service: +48 (0) 22 738 18 42</td></tr>
<tr><td>Poland – Polen
Bosch Rexroth Sp.zo.o.
Biuro Poznan
ul. Dabrowskiego 81/85
60-529 Poznan
Tel.: +48 061 847 64 62 /-63
Fax: +48 061 847 64 02</td><td>Romania - Rumänien
East Electric S.R.L.
Bdul Basarabia no.250, sector 3
73429 Bucuresti
Tel./Fax:: +40 (0)21 255 35 07
+40 (0)21 255 77 13
Fax: +40 (0)21 725 61 21
eastel@rdsnet.ro</td><td>Romania - Rumänien
Bosch Rexroth Sp.zo.o.
Str. Drobety nr. 4-10, app. 14
70258 Bucuresti, Sector 2
Tel.: +40 (0)1 210 48 25
+40 (0)1 210 29 50
Fax: +40 (0)1 210 29 52</td><td>Russia - Russland
Bosch Rexroth OOO
Tschschjolkowskoje Chaussee 100
105523 Moskau
Tel.: +7-495-783 30 60 + 64
Fax: +7-495 783 30 68 + 69
brcschrexroth.ru</td></tr>
<tr><td>Turkey - Türkei
Bosch Rexroth Otomasyon
San & Tic. A..S.
Fevzi Cakmak Cad No. 3
34295 Sefaköy Istanbul
Tel.: +90 212 411-13 00
Fax: +90 212 411-13 17
www.boschrexroth.com.tr</td><td>Turkey - Türkei
Servo Kontrol Ltd. Sti.
Perpa Ticaret Merkezi B Blok
Kat: 11 No: 1609
80270 Okmeydani-Istanbul
Tel: +90 212 320 30 80
Fax: +90 212 320 30 81
remzi.sali@servokontrol.com
www.servokontrol.com</td><td>Slowakia - Slowakei
Tightening & Press-fit:
MTS, spol .s.r.o.
02755 Kriva 53
Tel.: +421 43 5819 161</td><td>Slowenia - Slowenien
DOMEL
Otoki 21
64 228 Zelezniki
Tel.: +386 5 5117 152
Fax: +386 5 5117 225
brane.ozebek@domel.si</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

Australien, Süd-Afrika - Australia, South Africa

<table>
<tr><td>Australia - Australien
AIMS - Australian Industrial
Machinery Services Pty. Ltd.
28 Westside Drive
Laverton North Vic 3026
Melbourne
Tel.: +61 3 93 14 3321
Fax: +61 3 93 14 3329
Hotlines: +61 3 93 14 3321
+61 4 19 369 195
enquires@aimservices.com.au</td><td>Australia - Australien
Bosch Rexroth Pty. Ltd.
No. 7, Endeavour Way
Braeside Victoria, 31 95
Melbourne
Tel.: +61 3 95 80 39 33
Fax: +61 3 95 80 17 33
mel@rexroth.com.au</td><td>South Africa - Südafrika
TECTRA Automation (Pty) Ltd.
100 Newton Road, Meadowdale
Edenvale 1609
Tel.: +27 11 971 94 00
Fax: +27 11 971 94 40
Hotline: +27 82 903 29 23
georgv@tectra.co.za</td><td>South Africa - Südafrika
Tightening & Press-fit:
Jendamark Automation
76A York Road, North End
6000 Port Elizabeth
Tel.: +27 41 391 4735</td></tr>
</table>

Asien - Asia (incl. Pacific Rim)

China Shanghai Bosch Rexroth Hydraulics & Automation Ltd. No.122, Fu Te Dong Yi Road Waigaoqiao, Free Trade Zone Shanghai 200131 - P.R.China Tel.: +86 21 58 66 30 30 Fax: +86 21 58 66 55 23 brcservice@boschrexroth.com.cn	China Shanghai Bosch Rexroth Hydraulics & Automation Ltd. 4/f, Marine Tower No.1, Pudong Avenue Shanghai 200120 - P.R.China Tel: +86 21 68 86 15 88 Fax: +86 21 68 86 05 99 +86 21 58 40 65 77 y.wu@boschrexroth.com.cn	China Bosch Rexroth (China) Ltd. Satellite Service Office Changchun Rm. 1910, Guangming Building No.336 Xi'an Rd., Chao Yang Distr. Changchun 130061 - P.R.China Tel.+Fax: +86 431 898 1129 Mobile: +86 139 431 92 659 shuhong.wang@boschrexroth.com.cn	China Bosch Rexroth (China) Ltd. Satellite Service Office Wuhan No. 22, Pinglanju, Milanyuan, Golden Harbor No. 236 Longyang Avenue Economic & Technology Development Zone Wuhan 430056 - P.R.China Tel.+Fax: +86 27 84 23 23 92 Mobile: +86 139 71 58 89 67 ym.zhu@boschrexroth.com.cn
China Bosch Rexroth (China) Ltd. Beijing Representative Office Xi San Qi Dong, De Sheng Mei Wai Hai Dian District Beijing 100096, P.R.China Tel.: +86 10 82 91 22 29 Fax: +86 10 82 91 81 09 yan.zhang@boschrexroth.com.cn	China Bosch Rexroth (China) Ltd. Guangzhou Repres. Office Room 3710-3716, Metro Plaza, Tian He District, 183 Tian He Bei Rd Guangzhou 510075, P.R.China Tel.: +86 20 87 55 00 30 +86 20 87 55 00 11 Fax: +86 20 87 55 23 87 andrew.wang@boschrexroth.com.cn	China Bosch Rexroth (China) Ltd. Dalian Representative Office Room 2005,Pearl River Int. Building No.99 Xin Kai Rd., Xi Gang District Dalian, 116011, P.R.China Tel.: +86 411 83 68 26 02 Fax: +86 411 83 68 27 02 jason.tan@boschrexroth.com.cn	China **Tightening & Press-fit:** C. Melchers GmbH & Co Shanghai Representation 13 Floor Est Ocean Centre No.588 Yanan Rd. East 65 Yanan Rd. West Shanghai 200001 Tel.: +86 21 63 52 88 48 Fax: +86 21 63 51 31 38 shermanxia@sh.melchers.com.cn
Hongkong Bosch Rexroth (China) Ltd. 6th Floor, Yeung Yiu Chung No.6 Ind Bldg. 19 Cheung Shun Street Cheung Sha Wan, Kowloon, Hongkong Tel.: +852 27 86 46 32 Fax: +852 27 42 60 57 Paul.li@boschrexroth.com.cn	India - Indien Bosch Rexroth (India) Ltd. Electric Drives & Controls Plot. No.96, Phase III Peenya Industrial Area Bangalore – 560058 Tel.: +91 80 51 17 0-211...-218 Fax: +91 80 83 94 345 +91 80 83 97 374 mohanvelu.t@boschrexroth.co.in	India - Indien Bosch Rexroth (India) Ltd. Electric Drives & Controls Advance House, II Floor Ark Industrial Compound Narol Naka, Makwana Road Andheri (East), Mumbai - 400 059 Tel.: +91 22 28 56 32 90 +91 22 28 56 33 18 Fax: +91 22 28 56 32 93 singh.op@boschrexroth.co.in	India - Indien **Tightening & Press-fit:** MICO Hosur Road Adugodi 560 030 Bangalore Karnataki Tel.: +91 80 22 99 28 86
India - Indien Bosch Rexroth (India) Ltd. S-10, Green Park Extension New Delhi – 110016 Tel.: +91 11 26 56 65 25 +91 11 26 56 65 27 Fax: +91 11 26 56 68 87 koul.rp@boschrexroth.co.in	Indonesia - Indonesien PT. Bosch Rexroth Building # 202, Cilandak Commercial Estate Jl. Cilandak KKO, Jakarta 12560 Tel.: +62 21 7891169 (5 lines) Fax: +62 21 7891170 - 71 rudy.karimun@boschrexroth.co.id	Japan Bosch Rexroth Automation Corp. Service Center Japan Yutakagaoka 1810, Meito-ku, NAGOYA 465-0035, Japan Tel.: +81 52 777 88 41 +81 52 777 88 53 +81 52 777 88 79 Fax: +81 52 777 89 01	Japan Bosch Rexroth Automation Corp. Electric Drives & Controls 2F, I.R. Building Nakamachidai 4-26-44, Tsuzuki-ku YOKOHAMA 224-0041, Japan Tel.: +81 45 942 72 10 Fax: +81 45 942 03 41
Korea Bosch Rexroth-Korea Ltd. Electric Drives & Controls 1515-14 Dadae-Dong, Saha-gu Pusan Metropolitan City, 604-050 Tel.: +82 51 26 00 741 Fax: +82 51 26 00 747 eunkyong.kim@boschrexroth.co.kr	Korea Bosch Rexroth-Korea Ltd. Electric Drives and Controls Bongwoo Bldg. 7FL, 31-7, 1Ga Jangchoong-dong, Jung-gu Seoul, 100-391 Tel.: +82 234 061 813 Fax: +82 222 641 295	Korea Bosch Rexroth-Korea Ltd. Electric Drives & Controls 1515-14 Dadae-Dong, Saha-gu Ulsan, 680-010 Tel.: +82 52 256-0734 Fax: +82 52 256-0738 keonhyun.jeong@boschrexroth.co.kr	Korea **Tightening & Press-fit:** KVT Co., Ltd. 901, Daeryung Techno Town 8 481-11 Gasan-Dong Geumcheon-Gu Seoul, 153-775 Tel.: +82 2 2163 0231 9
Malaysia Bosch Rexroth Sdn.Bhd. 11, Jalan U8/82, Seksyen U8 40150 Shah Alam Selangor, Malaysia Tel.: +60 3 78 44 80 00 Fax: +60 3 78 45 48 00 hhlim@boschrexroth.com.my rexroth1@tm.net.my	Singapore - Singapur Bosch Rexroth Pte Ltd 15D Tuas Road Singapore 638520 Tel.: +65 68 61 87 33 Fax: +65 68 61 18 25 lai.ts@boschrexroth.com.sg	Taiwan Bosch Rexroth Co., Ltd. Taichung Industrial Area No.19, 38 Road Taichung, Taiwan 407, R.O.C. Tel : +886 - 4 -235 08 383 Fax: +886 - 4 -235 08 586 jim.lin@boschrexroth.com.tw david.lai@boschrexroth.com.tw	Taiwan Bosch Rexroth Co., Ltd. Tainan Branch No. 17, Alley 24, Lane 737 Chung Cheng N.Rd. Yungkang Tainan Hsien, Taiwan, R.O.C. Tel : +886 - 6 –253 6565 Fax: +886 - 6 –253 4754 charlie.chen@boschrexroth.com.tw
Thailand NC Advance Technology Co. Ltd. 59/76 Moo 9 Ramintra road 34 Tharang, Bangkhen, Bangkok 10230 Tel.: +66 2 943 70 62 +66 2 943 71 21 Fax: +66 2 509 23 62 Hotline +66 1 984 61 52 sonkawin@hotmail.com			

Rexroth
Bosch Group

Nordamerika – North America

USA Headquarters - Hauptniederlassung Bosch Rexroth Corporation Electric Drives & Controls 5150 Prairie Stone Parkway Hoffman Estates, IL 60192-3707 Tel.: +1 847 645-3600 Fax: +1 847 645-6201 servicebrc@boschrexroth-us.com repairbrc@boschrexroth-us.com	USA Central Region - Mitte Bosch Rexroth Corporation Electric Drives & Controls 1701 Harmon Road Auburn Hills, MI 48326 Tel.: +1 248 393-3330 Fax: +1 248 393-2906	USA Southeast Region - Südost Bosch Rexroth Corporation Electric Drives & Controls 2810 Premiere Parkway, Suite 500 Duluth, GA 30097 Tel.: +1 678 957-4050 Fax: +1 678 417-6637	**USA SERVICE-HOTLINE** - 7 days x 24hrs - **+1-800-REXROTH** **+1 800 739-7684**
USA Northeast Region – Nordost Bosch Rexroth Corporation Electric Drives & Controls 99 Rainbow Road East Granby, CT 06026 Tel.: +1 860 844-8377 Fax: +1 860 844-8595	USA West Region – West Bosch Rexroth Corporation Electric Drives & Controls 7901 Stoneridge Drive, Suite 220 Pleasanton, CA 94588 Tel.: +1 925 227-1084 Fax: +1 925 227-1081	Canada East - Kanada Ost Bosch Rexroth Canada Corporation 5345 Outer Drive unit 5 Windsor, Ontario Canada N9A 6J3 Tel.: +1 519 737 7393 Fax.: +1 519 737 9999	Canada East - Kanada Ost Bosch Rexroth Canada Corporation Automation Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8 Tel.: +1 905 335 5511 Fax: +1 905 335 4184 (Main) +1 905 335 9803 (Serv.) automation.service@boschrexroth.ca automation.repair@boschrexroth.ca
Canada West - Kanada West Bosch Rexroth Canada Corporation 5345 Goring St. Burnaby, British Columbia Canada V7J 1R1 Tel. +1 604 205 5777 Fax +1 604 205 6944 automation.service@boschrexroth.ca automation.repair@boschrexroth.ca	**Canada SERVICE-HOTLINE** - 7 days x 24hrs - **+1 905 335 5511**	Mexico Bosch Rexroth Mexico S.A. de C.V. Calle Neptuno 72 Unidad Ind. Vallejo 07700 Mexico, D.F. Tel.: +52 55 57 54 17 11 Fax: +52 55 57 54 50 73 mario.francioli@boschrexroth.com.mx	Mexico Bosch Rexroth S.A. de C.V. Calle Argentina No 3913 Fracc. las Torres 64930 Monterrey, N.L. Tel.: +52 81 83 49 80 91 +52 81 83 49 80 92 +52 81 83 49 80 93 Fax: +52 81 83 65 52 80

Südamerika – South America

Argentina - Argentinien Bosch Rexroth S.A.I.C. "The Drive & Control Company" Rosario 2302 B1606DLD Carapachay Provincia de Buenos Aires Tel.: +54 11 4756 01 40 +54 11 4756 02 40 +54 11 4756 03 40 +54 11 4756 04 40 Fax: +54 11 4756 01 36 +54 11 4721 91 53 victor.jabif@boschrexroth.com.ar	Argentina - Argentinien NAKASE SRL Servicio Tecnico CNC Calle 49, No. 5764/66 B1653AOX Villa Balester Provincia de Buenos Aires Tel.: +54 11 4768 36 43 Fax: +54 11 4768 24 13 **Hotline: +54 11 155 307 6781** nakase@usa.net nakase@nakase.com gerencia@nakase.com (Service)	Brazil - Brasilien Bosch Rexroth Ltda. Av. Tégula, 888 Ponte Alta, Atibaia SP CEP 12942-440 Tel.: +55 11 4414 -56 92 +55 11 4414 -56 84 Fax sales: +55 11 4414 -57 07 Fax serv.: +55 11 4414 -56 86 alexandre.wittwer@rexroth.com.br	Brazil - Brasilien Bosch Rexroth Ltda. R. Dr.Humberto Pinheiro Vieira, 100 Distrito Industrial [Caixa Postal 1273] 89220-390 Joinville - SC Tel./Fax: +55 47 473 58 33 Mobil: +55 47 9974 6645 sergio.prochnow@boschrexroth.com.br
Columbia - Kolumbien Reflutec de Colombia Ltda. Calle 37 No. 22-31 Santafé de Bogotá, D.C. Colombia Tel.: +57 1 208 65 55 Fax: +57 1 269 97 37 reflutec@etb.net.co			

Rexroth
Bosch Group

Bosch Rexroth AG
Electric Drives and Controls
P.O. Box 13 57
97803 Lohr, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Germany
Phone +49 (0)93 52-40-50 60
Fax +49 (0)93 52-40-49 41
service.svc@boschrexroth.de
www.boschrexroth.com

R911309162

Printed in Germany
DOK-VISMOT-VM*-11VRS**-FK01-EN-P