CL350 / CL400 / CL500

MMI-MADAP for Programmers and Project Designers Software manual

Edition 101

CL350 / CL400 / CL500

MMI-MADAP for Programmers and Project Designers Software manual

1070 072 168-101 (98.04) GB

1 Safety Instructions

Before you start working with the MMI-MADAP, we recommend that you thoroughly familiarize yourself with the contents of this manual. Keep this manual in a place where it is always accessible to all users.

1.1 Proper use

This instruction manual presents a comprehensive set of instructions and information required for the standard operation of the described products.

The products described hereunder

- were developed, manufactured, tested and documented in accordance with the relevant safety standards. In standard operation, and provided that the specifications and safety instructions relating to the project phase, installation and correct operation of the product are followed, there should arise no risk of danger to personnel or property.
- are certified to be in full compliance with the requirements of the
  - COUNCIL DIRECTIVE 89/336/EEC of May 3rd 1989 on the approximation of the laws of the Member States relating to electromagnetic compatibility, 93/68/EEC (amendments of Directives), and 93/44/EEC (relating to machinery)
  - COUNCIL DIRECTIVE 73/23/EEC (electrical equipment designed for use within certain voltage limits)
  - Harmonized standards EN 50081–2 and EN 50082–2
- are designed for operation in an industrial environment (Class A emissions). The following restrictions apply:
  - No direct connection to the public low–voltage power supply is permitted.
  - Connection to the medium and/or high–voltage system must be provided via transformer.

  The following applies for application within a personal residence, in business areas, on retail premises or in a small–industry setting:
  - Installation in a control cabinet or housing with high shield attenuation.
  - Cables that exit the screened area must be provided with filtering or screening measures.
  - The user will be required to obtain a single operating license issued by the appropriate national authority or approval body. In Germany, this is the Federal Institute for Posts and Telecommunications, and/or its local branch offices.

⇒ **This is a Class A device. In a residential area, this device may cause radio interference. In such case, the user may be required to introduce suitable countermeasures, and to bear the cost of the same.**

Proper transport, handling and storage, placement and installation of the product are indispensable prerequisites for its subsequent flawless service and safe operation.

1.2 Qualified personnel

This instruction manual is designed for specially trained personnel. The relevant requirements are based on the job specifications as outlined by the ZVEI and VDMA professional associations in Germany. Please refer to the following German–Language publication:
Weiterbildung in der Automatisierungstechnik
Publishers: ZVEI and VDMA Maschinenbau Verlag
Postfach 71 08 64
60498 Frankfurt/Germany

Interventions in the hardware and software of our products not described in this instruction manual may only be performed by our skilled personnel.

Unqualified interventions in the hardware or software or non–compliance with the warnings listed in this instruction manual or indicated on the product may result in serious personal injury or damage to property.

Installation and maintenance of the products described hereunder is the exclusive domain of trained electricians as per IEV 826–09–01 (modified) who are familiar with the contents of this manual.

Trained electricians are persons of whom the following is true:

- They are capable, due to their professional training, skills and expertise, and based upon their knowledge of and familiarity with applicable technical standards, of assessing the work to be carried out, and of recognizing possible dangers.
- They possess, subsequent to several years' experience in a comparable field of endeavour, a level of knowledge and skills that may be deemed commensurate with that attainable in the course of a formal professional education.

With regard to the foregoing, please read the information about our comprehensive training program. The professional staff at our training centre will be pleased to provide detailed information. You may contact the centre by telephone at (+49) 6062 78–258.

1.3 Safety markings on components

DANGER! High voltage!

DANGER! Corrosive battery acid!

CAUTION! Electrostatically sensitive components!

Disconnect mains power before opening!

Lug for connecting PE conductor only!

Functional earthing or low–noise earth only!

Screened conductor only!

1.4 Safety instructions in this manual

DANGEROUS ELECTRICAL VOLTAGE
This symbol warns of the presence of a **dangerous electrical voltage**. Insufficient of lacking compliance with this warning can result in **personal injury.**

DANGER
This symbol is used wherever insufficient or lacking observance of this instruction can result in **personal injury.**

CAUTION
This symbol is used wherever insufficient or lacking observance of instructions can result in **damage to equipment or data files.**

⟹ This symbol is used to alert the user to an item of special interest.

1.5 Safety instructions for the described product

DANGER
Fatal injury hazard through ineffective Emergency–OFF devices! Emergency–OFF safety devices must remain effective and accessible during all operating modes of the system. The release of functional locks imposed by Emergency–OFF devices must never be allowed to cause an uncontrolled system restart! Before restoring power to the system, test the Emergency–OFF sequence!

DANGER
Danger to persons and equipment!
Test every new program before operating the system!

DANGER
Retrofits or modifications may interfere with the safety of the products described hereunder!

The consequences may be severe personal injury or damage to equipment or the environment. Therefore, any system retrofitting or modification utilizing equipment components from other manufacturers will require express approval by Bosch.

DANGEROUS ELECTRICAL VOLTAGE
Unless described otherwise, maintenance procedures must always be carried out only while the system is isolated from the power supply. During this process, the system must be blocked to prevent an unauthorized or inadvertent restart.

If measuring or testing procedures must be carried out on the active system, these must be carried out by trained electricians.

CAUTION
Danger to the module!
Do not insert or remove the module while the controller is switched ON! This may destroy the module. Prior to inserting or removing the module, switch OFF or remove the power supply module of the controller, external power supply and signal voltage!

CAUTION
Only Bosch–approved spare parts may be used!

CAUTION
Danger to the module!
All ESD protection measures must be observed when using the module! Prevent electrostatic discharges!

Observe the following protective measures for electrostatically endangered modules (EEM)!

- The Employees responsible for storage, transport and handling must be trained in ESD protection.
- EEMs must be stored and transported in the protective packaging specified.
- Out of principle, EEMs may be handled only at special ESD work stations equipped for this particular purpose.
- Employees, work surfaces and all devices and tools that could come into contact with EEMs must be on the same potential (e.g. earthed).
- An approved earthing wrist strap must be worn. It must be connected to the work surface via a cable with integrated 1 MW resistor.
- EEMs may under no circumstances come into contact with objects susceptible to accumulating an electrostatic charge. Most items made of plastic belong to this category.
- When installing EEMs in or removing them from an electronic device, the power supply of the device must be switched OFF.

1.6 Trademarks

All trademarks referring to software that is installed on Bosch products when shipped from the factory represent the property of their respective owners.

At the time of shipment from the factory, all installed software is protected by copyright. Software may therefore be duplicated only with the prior permission of the respective manufacturer or copyright owner.

MS–DOS® and Windows™ are registered trademarks of Microsoft Corporation.

PROFIBUS® is a registered trademark of the PROFIBUS Nutzerorganisation e.V. (user organization).

2 Introduction

This manual is designed to support the MMI-MADAP software user with all activities related to project design, programming and system start-up. The manual discusses handling procedures for the MMI-MADAP software with regard to the programmable logic controller being used, including the *control panel*. Due to the additional hardware features and operating functions which go beyond those of a mere control panel, the term *operator terminal* appears appropriate to describe the BT100 discussed throughout these pages.

Fig. 2-1 Base Screen MMI-MADAP Software

Subjects related to the operation and programming of the MMI-MADAP software are addressed in the supplementary manual indicated below:

☞ **BOSCH documentation reference**

MMI-MADAP for System or Machine Operators — Software Manual	Part no. 1070 072 167

2.1 Overview of Functions

To illustrate the powerful features of the MMI-MADAP software, the implemented functions are listed individually in the following sections.

General information and functions featured in each screen

- System time and date display
- Controller ID (user ID)
- PLC type and central processing unit ID
- Two permanently displayed alarm lines (message bars)
- Processing Unit Stopped message
- Print Screen function
- Permanently available Help windows

Project handling and backup management

- Multilevel access privileges via password system
- User-configurable Save, Load and Delet functions
- Selectable user language

Power-up screens providing overview of the machine start prerequisites

- 6 power-up screens with 32 power-up conditions each
- Screen titles and function key labels, plus bit variable and text for each power-up condition can be defined on-line on the BT100 operator terminal.
- Lamp test function

User screens for visualization and parameter selection for machines and systems

- 8 groups with 8 user screens each (total of 64 screens)
- On the BT100 operator terminal, on-line definition of group titles, function key labels, screen matrix
- Comfortable editor for screens and variables, with vector graphics and bitmap level, plus access to additional tools, such as alarm system, recipe management, math worksheet, trend functions, protocol/record system, scheduler, Unisoft language, etc.

Movement screens for manual machine operation and setup

- Step-programmable, user-defined manual conditions
- Movement initiation through activation of step programmed for this movement
- Diagnostics also for all manual movements
- 8 groups with 8 movement screens each, each featuring 8 movement pairs: = 1024 movements
- Definable on-line on BT100: Group titles, screen titles, function key labels, screen matrix, link with user screen
- For each movement, definable on-line on BT100: cascade/step, movement text, two variables (actuator and end positions), incl. descriptive text
- The executability of a given movement is indicated on the screen
- The statuses of the actuators and their end positions are visible on the screen.
- Immediate motion stop upon release of key
- Event-controlled movement inhibition centrally possible for all movement screens, and selectively for individual movement screens.

Status screens to support start-up and maintenance functions

- Status display for all PLC operands:
- I/EI, O/EO, M/SM/S, T/C, DF/DP, all available DMs
- Data module list of all available DMs
- Display of equipped system modules
- Display of version identifiers of relevant function modules
- Indication of active input/output bytes
- Information about PLC cycle times and communication interruptions
- Display of current Time/OM declarations
- Current PLC warnings / messages / information statuses
- Setting time and date

Machine usage

- Production statistics, current / actual or historical
- Data recording for 3 work shifts with 6 breaks each
- Standardized recording parameters for:
  - Machine On
  - Production
  - Fault
  - No Parts
  - Buffer Full
  - Standstill
  - Parts actual /Parts desired
- Bar graph
- Overview of production day
- Overview of individual work shifts
- Plotting of curves for current or selectable historical period
- Exportable Trend data

Machine cycle times

48 machine cycle times with on-line text definition on the BT100 operator terminal

Message systems and protocol record

Current messages, long-time protocols, fault statistics

- 5 message systems for current messages / alarms:
  - First-value errors in cascade diagnostics
  - PLC system messages
  - PROFIBUS-DP diagnostics
  - Bus and bus station faults *
  - Status messages, 128 (user) *
  - Serial messages, 511 (user) *
  - * = On-line alarm text definition on BT100
- Protocol record with selectable life cycles for:
  - First-value errors in cascade diagnostics
  - PLC system messages
  - PROFIBUS-DP diagnostics
  - User messages
- Display of protocol record for selectable time periods
- Statistics for first-value errors in cascade diagnostics:
  - Resolution into detailed error patterns
  - Bar graph display of error frequency distribution of most major error patterns

Diagnostics for rapid error detection and troubleshooting

- Self-teaching diagnostics for process sequences
- Automatic adoption of symbols, symbol comments, step text and cascade text from PLC project
- Automatic first-value diagnostics
- Manual diagnostics of each cascade with the current step
- Diagnostic display in the form of instruction list (IL) or ladder diagram (LD)
- PROFIBUS-DP diagnostics, bus and bus station errors
- Serial message system containing 511 messages, coming / going
- Parallel message system encompassing 128 statuses

Synchronization for automatic restart without control reset

3 Suitable Controllers

Introduction

MMI-MADAP comprises a software system capable of controlling the CL350, CL400 and CL500 Bosch-proprietary PLC controllers.

Hardware concept with PROFIBUS-FMS networking

This is an MMI-MADAP hardware concept that provides for PC control panels to be operated on a CL400 or CL500 (but **not** CL350) programmable logic controller via a PC PROFIBUS-FMS card, with the respective controller being connected to the PROFIBUS-FMS via the Bosch-proprietary R500P or COM-P interface card.

MMI-MADAP is capable of managing, on one CL400 or CL500, up to four operator terminals (per central processing unit). A CL500 can accommodate up to four central processing units.

A maximum of eight MMI-MADAP operator terminals can be connected to a single R500P or COM-PPROFIBUS-FMS card.

The entire PROFIBUS-FMS management is handled by the MMI-MADAP operator terminals. On the side of the programmable logic controller, no PROFIBUS-FMS software modules are required.

Hardware concept as a point-to-point connection with the Bosch BUEP19E transmission protocol

This is an MMI-MADAP hardware concept that provides for PC control panels to be operated either via the central processing units and/or the R500 interface card on a Bosch CL350, CL400 or CL500 programmable logic controller.

MMI-MADAP is capable of managing, on one CL400 or CL500, up to four operator terminals (per central processing unit). A CL500 can accommodate up to four central processing units.

In the case of the CL350, the operator terminal can be connected only to the central processing unit. As it is not possible to install an R500 card, this controller does not permit the use of several operator terminals.

Signals generated by hardware operating elements (e.g. Start key switch) can be transferred via the PROFIBUS-DP decentralized bus system.

For details about the hardware configuration of PC operating panels, please refer to »System Requirements for MMI-MADAP Operator Terminal « on page 6-2 of this manual.

Hardware concept with PROFIBUS-FMS networking

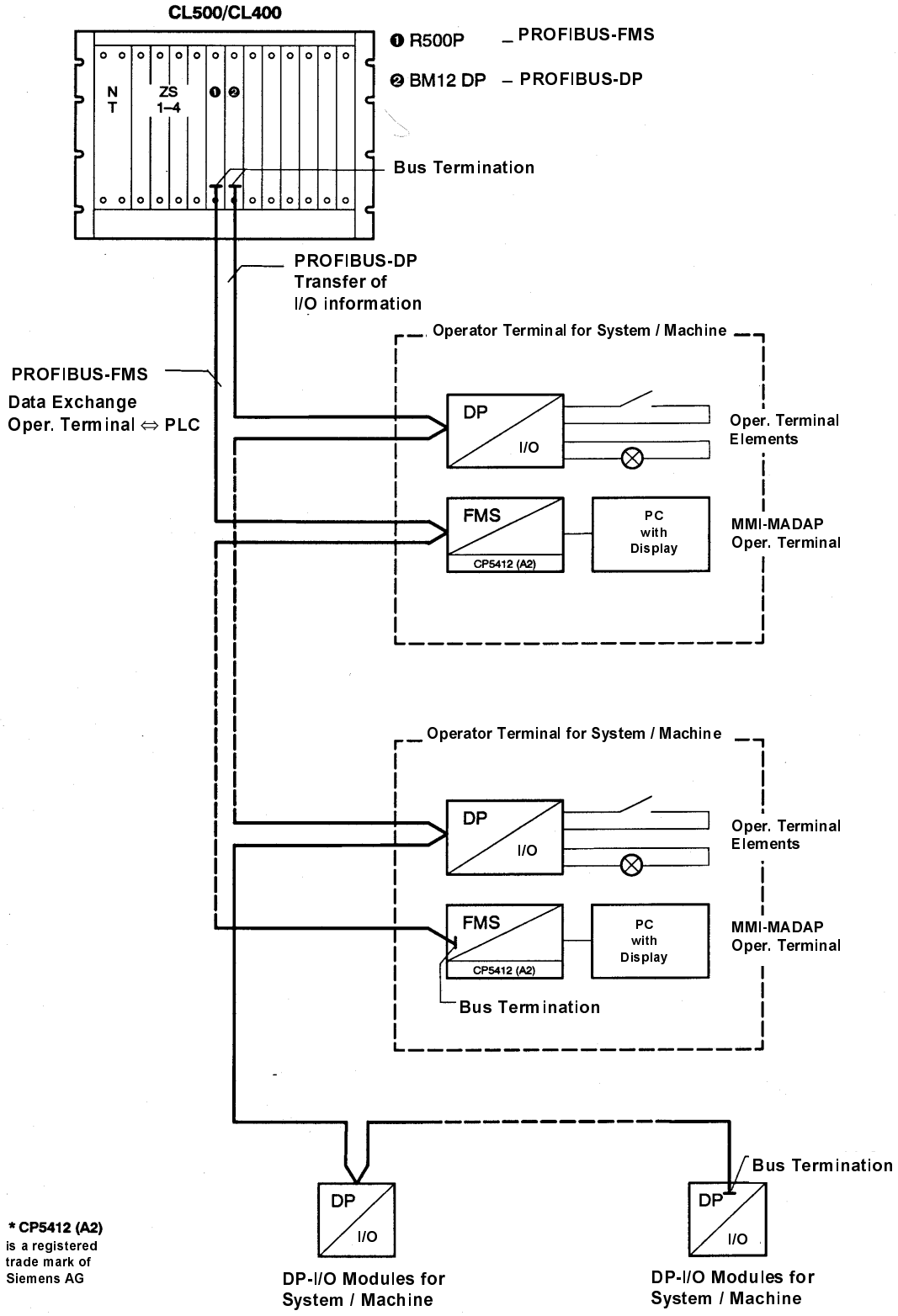


Fig. 3-1 Hardware Configuration, MMI-MADAP with PROFIBUS-FMS

4 MMI-MADAP PLC Software

4.1 Introduction

The MMI-PLC software consists of the following function units:

- Control
- Operating / Monitoring
- Machine usage
- Machine cycle time recording

The referred function units are capable of standalone operation, and operate independently of each other.

Control

Performance features:

- Control of a maximum of 64 cascades operating in parallel, with up to 128 steps per cascade.
- Synchronization of control sequences to current machine status.
- Management of system operating modes.
- Linear or branched sequence organization.
- Command output.

Operating / Monitoring

Performance features:

- Execution of 1024 movements directly from operator terminal.
- Display of system fault conditions and machine statuses.
- Display of PLC errors and fault statuses.
- Monitoring and reporting of sequential faults.
- Monitoring and reporting of PROFIBUS-DP faults.

Machine usage

Recording of production data for the following parameters:

- Machine On
- Production
- Fault
- Standstill
- Buffer Full
- No parts
- Parts Actual

Cycle time recording

Recording / logging of 48 machine cycle times.

Data interface

A number of data modules are defined to serve as the interface between the PLC controller and the operator terminal. Predefined data word ranges are declared as PROFIBUS-FMS objects. These are directly read and/or written to by the operator terminal.

Differences between PROFI and WinSPS PLC programming software

☞ **For the purpose of representing and expressing constants and program modules, this documentation uses the conventions of the PROFI software designed for the programming device. For this reason, they may appear dissimilar when using the WinSPS programming software.**

Differences in programming and representation of word constants

Data type		PLC Utility programs	
Explanation	**Representation**	**PROFI**	**WinSPS**
UINT (unsigned integer)	Binary / Dual	K00000000 00000000B K11111111 11111111B	2#0000000000000000 2#1111111111111111
	Decimal, Word	K00000D - K63535D	00000 - 65535
	Decimal, Byte/Byte	K000/000 - K255/255	not defined in IEC1131 Teil 3
	Hexadecimal	K0000H - KFFFFH	16#0000 - 16#FFFF
INT (integer)	Decimal, Word	K-32768 - K+32767 K-32768D - K+32767D	-32768 - +32767
Text, STRING(2)	ASCII	K'AB'	'AB'
Time value, TVALUE	Time value (+time base r) r: 0=10ms, 1=100ms 2=1s, 3=10s	K0.r - K1023.r	T#10ms - T#10230s T#0.r - T#1023.r

Differences in programming and representation of program module calls

	PLC Utility programs	
	PROFI	**WinSPS**
Program module/Function call (IEC1131/3)	CM PM	CM FC

Differences in programming and representation of jump instructions

	PLC Utility programs	
	PROFI	**WinSPS**
Jump instruction	JPx -label	JPx label
Branch destination	-label	label:

4.2 Software Installation

4.2.1 Supplied Software

4.2.1.1 DISK.FB.MMIMADAP Diskette

This diskette contains the following software files:

Organization modules

OM1	Administration module
OM2	Definition module
OM5	Start-up module, subsequent to Power-ON
OM7	Start-up module, subsequent to STOP/RUN
OM9	Error module
OM18 - OM25	Time-controlled modules 1 through 8

Open program modules (can be displayed and modified)

SCHRK1	Kette 1 transition program
KETTE1	Call-up for "KETTE" PM and command processing for Kette 1
MMIDESI	PROFIBUS_DP_Diagnose call-up module

Library modules (can neither be displayed nor modified)

MMIMADAP	Data processing for screen displays
KETTE	Control sequence management
DIAGMMI	Control sequence diagnostics
R5INIT	R500P initialization
BT100DEC	BT100 key decoding
MMIAUSL	Machine usage administration module
MMISTAT	Statistical data
MMISTCK	Actual piece counts
MMIPROZ	Percentage data
MMISCHT	Work shift information
MMILOGIK	Logging/recording parameters
MMI_TZ	Cycle time evaluation for 48 cycle times
MMI_T1S	Freerunning 1-s cycle
MMI_T01S	Freerunning 0.1-s cycle
DPSTATUS	DESI-DP data processing
FIFODM1	DESI-DP data processing

The OM5 and OM7 organization modules (start-up modules) contain the call for the R5INIT standard module. This module is used for initialization of the status utility up to and including version 6 of the R500P module (refer to front panel labelling). This utility generates the CPU Stopped and Communication Failure system messages.
Beginning with version 7 of the R500P, the R5INIT module is no longer required.
The R5INIT module is supplied on the diskette containing function modules for CL400 / CL500 standard interfaces. If required, it can be ordered as part number 069065.

MMI-MADAP Standard symbol file

Standard data modules (for WinSPS only)

Preconfigured SC table.

PROFIBUS-FMS Configuration files for R500P

4.2.2 Installing MMI-MADAP PLC Software

The diskette labelled DISK.FB.MMIMADAP contains the following directories:

- **ProfiSPS** (PLC files for PROFI software),
- **PROFIBUS** (PROFIBUS-FMS files for the R500P).
- **WinSPS** (PLC files for WinSPS software),

```
A:\
  +---PROFIBUS          PROFIBUS configuration files
  ¦   +---COM-P.400     for use with CL400
  ¦   +---COM-P.500     for use with CL500
  ¦   +---R500P.400     for use with CL400
  ¦   +---R500P.500     for use with CL500
  ¦       +---00122232      Configuration 1
  ¦       :           (ZS0:0BF,ZS1:2BF,ZS2:2BF, ZS3:2BF)
  ¦       :
  ¦       +---04142030      Configuration 21
  ¦               (ZS0:4BF,ZS1:4BF,ZS2:0BF, ZS3:0BF)
  ¦
  +---ProfiSPS        PLC files for PROFI
  ¦    +---BOSCH.BIB        Library modules for CL350 / 400 / 500
  ¦    +---MMIMADAP.500
  ¦    ¦    +---SK          SC table for CL500
  ¦    ¦    +---ZS0         "open" modules for CL500
  ¦    +---MMIMADAP.400
  ¦        +---SK          SC tables for CL350 / 400
  ¦        +---ZS0         "open" modules for CL350 / 400
  ¦
  +---WinSPS        PLC files for WinSPS
      +---BIBCL4_5      Library modules for CL350 / 400 / 500
      +---MMIMADAP.PRJ
          +---CL500.500
              +---SK        SC table for CL500
              ¦   +--SK3_4  SC tables for CL350 / 400
              +---ZS0       "open" modules for CL350 /400 /500
```

Fig. 4-2 Directory Structures on DISK.FB.MMIMADAP Diskette

The program modules listed below are supplied in the form of library modules on the diskette labelled DISK.FB.MMIMADAP. They are located in the following library directories:

- winsps\bibcl4_5 (WinSPS), and/or
- profisps\bosch.bib (PROFI).

MMI-MADAP library modules

MMIMADAP Data processing for screen displays

KETTE Control sequence management

DIAGMMI Control sequence diagnostics

BT100DEC BT100 key decoding

MMIAUSL Machine usage administration

MMISTAT	Statistical data
MMISTCK	Actual piece counts
MMIPROZ	Percentage data
MMISCHT	Work shift information
MMILOGIK	Logging/recording parameters
MMI_TZ	Cycle time evaluation for 48 cycle times
MMI_T1S	Freerunning 1-s cycle
MMI_T01S	Freerunning 0.1-s cycle
DPSTATUS	DESI-DP data processing
FIFODM1	DESI-DP data processing

Please copy these files as required from the diskette to your applicable library directory or directories.

.pxl filename extension	WinSPS files for CL400 / CL500
.p5l filename extension	PROFI files for CL500
.pcl filename extension	PROFI files for CL400

As a next step, copy all required files from the applicable directory listed below to the respective project directory:

winsps\mmimadap.prj\cl500.500\zs0
(CL500 modules, WinSPS software),

profisps\mmimadap.400\zs0
(CL400 modules, PROFI software), OR

profisps\mmimadap.500\zs0
(CL500 modules, PROFI software)

☞ **If you are using the CL350 or CL400 with the WinSPS software, copy the listed CL500 modules into your project directory. This will cause the modules to be processed automatically upon start-up.**

☞ **Please be careful to prevent accidental overwriting of files in the project directory that you have created yourself.**

In the event that you have not yet created a current file for your system co-ordinator (SC table), you can copy the required file from the corresponding directory ending with ...\sk into your current SK project directory.

4.2.3 Linking Standard Symbol File and Standard Modules

Subsequent to the completed installation, the symbol file will be located in the current project directory.

☞ **Filename: Mmimadap.sxs for WinSPS version**
Filename: Mmimadap.s5s for CL500 PROFI version
Filename: Mmimadap.scs for CL400 PROFI version

When starting a new project, please use this symbol file.

If you are already using a symbol file, enter the module names specified in Section 3.2.1.1 and, if required, the R5INIT module, into your symbol file.

If you want to integrate the MMI-MADAP software into an existing project, you will be required to open all data modules named DM230 through DM255, and enter them in the symbol file. If you want to use also the machine usage times and machine cycle time functions, you must open data modules DM217 through DM221 also.

In this case, copy the data modules from the standard symbol file into your current symbol file (including comments and default values).

☞ **In the case of the WinSPS version, the installation procedures will copy all standard data modules into the current project directory.**

☞ **Up to and including version 6 of the R500P PROFIBUS-FMS card, the R5INIT module is required for initializing the status utility that generates the CPU Stopped and Communication Failure system messages.**
Beginning with version 7 of the R500P, the R5INIT module is no longer required.

4.2.4 Configuring Com-P and R500P PROFIBUS Cards

For the purpose of configuring the PROFIBUS-FMS, ready-to-use configuration files are provided on the DISK.FB.MMIMADAP standard diskette. The filenames are as follows:

- 001SPS.KBL
- 001SPS.OV
- XXXSPS.BUS

4.2.4.1 Com-P Card

CL400 Controller

For use with the CL400, the directory named Com-P.400 contains a fully prepared configuration for 4 operator terminals.

CL500 Controller

For use with the CL500, the directory named Com-P.500 contains a fully prepared configuration for 4 central processing units with 4 operator terminals each.

4.2.4.2 R500P Card

CL500 Controller

As the R500P is capable of managing only 100 objects, it is not possible to provide a ready-to-use configuration for the maximum hardware configuration of the MMI-MADAP concept onboard the module. For this reason, various hardware configurations for the CL500 were provided on the MMI-SPS diskette. As a result, configuration files are available for all useful configuration variants (combination between the numbers of operator terminals and central processing units).

The different variants are located in 21 different subdirectories. The type of combination contained in a given subdirectory can be directly recognized from the respective directory name. This means that the specified combination of numerals must always be read in pairs, with the first numeral indicating the number of the central processing unit ("ZS number"), and the second numeral indicating the number of operator terminals (BT) assigned to this central processing unit.

☞ **NOTE:**
The abbreviations "BT" and "BF" (from the German --> *Bedien-Terminal* and *Bedienfeld*) **shall have the meaning of "operator terminal" throughout this manual.**

Example 1:

Directory pathname: r500p.500\02*12***2232***

- 02: ZS0 operates 2 BT
- *12*: ZS1 operates 2 BT

22: ZS2 operates 2 BT

32: ZS3 operates 2 BT

Example 2:

Directory pathname: r500p.500\03*13***2230***

03: ZS0 operates 3 BT

13: ZS1 operates 3 BT

22: ZS2 operates 2 BT

30: ZS3 operates 0 BT

CL400 Controller

For use with the CL400, the directory named r500p.400 contains a fully prepared configuration for 4 operator terminals.

4.2.4.3 Installation Procedure

From the respective directory, e.g.
r500p.500\02122232,
copy the PROFIBUS files to the applicable project directory, e.g.
c:\pg\mmimadap.500\profibus.

Use the PROFI programming device software to load the PROFIBUS files into the PLC controller.

☞ **Ensure that the R500P is set to PROFIBUS-FMS station address no. "1."**

☞ **BOSCH documentation reference**

Communication Module for CL400/CL500 COM-P (Manual)	in preparation
CL 500/ R500P Computer Module (Manual)	No. 1070 072 138
PROFI Programming Software (Manual)	No. 1070 072 129

4.3 PLC Program Components

4.3.1 Organization Modules

	Symbolic name	Function
OM1	OM1	Administration module with program module calls for basic functions
OM2	OM2	Definition module
OM5	OM5	Start-up module subsequent to Power-ON
OM7	OM7	Start-up module subsequent to STOP/RUN
OM9	OM9	Error module
OM18	OM18	Time-controlled module 1
OM19	OM19	Time-controlled module 2
OM20	OM20	Time-controlled module 3
OM21	OM21	Time-controlled module 4
OM22	OM22	Time-controlled module 5
OM23	OM23	Time-controlled module 6
OM24	OM24	Time-controlled module 7
OM25	OM25	Time-controlled module 8

Fig. 4-3 List of Organization Modules

4.3.2 Program Modules

	Symbolic name	Function
PM1	SCHRK1	Kette 1 transition program
PM100	KETTE1	Call-up of KETTE PM, and command processing for Kette 1
PM200	MMIMADAP	Data processing for screen displays
PM201	KETTE	Control sequence management
PM202	DIAGMMI	Control sequence diagnostics
PM203	R5INIT	Initialization of status utility, R500P or COM-P (not required for R500P, version 7 and up)
PM205	BT100DEC	BT100 key decoding
PM207	MMIAUSL	Machine usage administration module
PM208	MMISTAT	Statistical data (Call-up via MMIAUSL)
PM209	MMISTCK	Actual piece counts (Call-up via MMIAUSL)
PM210	MMIPROZ	Percentage data (Call-up via MMIAUSL)
PM211	MMISCHT	Work shift information (Call-up via MMIAUSL)
PM212	MMILOGIK	Logging/recording parameters (Call-up via MMIAUSL)
PM214	MMI_TZ	Cycle time evaluation for 48 cycle times
PM215	MMI_T1S	Freerunning 1.0-sec cycle (Call-up via MMI_TZ)
PM216	MMI_T01S	Freerunning 0.1-sec cycle (Call-up via MMI_TZ)
PM218	MMIDESI	Call-up module for PROFIBUS-DP diagnostics
PM219	DPSTATUS	DESI-DP data processing (Call-up via MMIDESI)
PM220	FIFODM1	DESI-DP data processing (Call-up via DPSTATUS)

Fig. 4-4 List of Program Modules

4.3.3 Data Modules

DM no.	Name	Function	R/E	Length
DM 1	DB_K01	Kette 1 cascade information	R	100
:	:	:	:	:
DM 64	DB_K64	Kette 64 cascade information	R	100
DM 217	Taktzeit	Cycle time recording	R	230
DM 218	Auslast1	Machine usage data 1	R	512
DM 219	Auslast2	Machine usage data 2	R	512
DM 220	DP_Daten	DESI-DP data	R	512
DM 221	DP_Komm	DESI-DP communication channel 1	R	320
DM 222	Diag_St5	Station 5 diagnostic data (optional)	R	512
DM 223	Diag_St6	Station 6 diagnostic data (optional)	R	512
DM 224	Diag_St7	Station 7 diagnostic data (optional)	R	512
DM 225	Diag_St8	Station 8 diagnostic data (optional)	R	512
DM 226				
DM 227				
DM 228				
DM 229				
DM 230	BF1_DB	Communication / display data	R	512
DM 231	BF1_Stat	OPD status	R	512
DM 232	BF1_Sper	Movement lock functions	R	512
DM 233	BF1_Anw	Communication DM, 1 Object of 220 bytes	R	512
DM 234	BF1_Diag	Diagnostic result for BT1 = Station 1	R	512
DM 235	BF2_DB	Communication / display data	R	512
DM 236	BF2_Stat	Operand status	R	512
DM 237	BF2_Sper	Movement lock functions	R	512
DM 238	BF2_Anw	Communication DM, 1 Object of 220 bytes	R	512
DM 239	BF2_Diag	Diagnostic result for BT2 = Station 2	R	512
DM 240	BF3_DB	Communication / display data	R	512
DM 241	BF3_Stat	Operand status	R	512
DM 242	BF3_Sper	Movement lock functions	R	512
DM 241	BF3_Anw	Communication DM, 1 Object of 220 bytes	R	512
DM 244	BF3_Diag	Diagnostic result for BT3 = Station 3	R	512
DM 245	BF4_DB	Communication / display data	R	512
DM 246	BF4_Stat	Operand status	R	512
DM 247	BF4_Sper	Movement lock functions	R	512
DM 246	BF4_Anw	Communication DM, 1 Object of 220 bytes	R	512
DM 249	BF4_Diag	Diagnostic result for BT4 = Station 4	R	512
DM 250	BF1_4Anw	DM250 for all BT, 2 objects of 220 bytes ea.	R	512
DM 251	ResKObj1	Reserved for future communication objects	R	
DM 252	ResKObj2	Reserved for future communication objects	R	
DM 253	EA_SK	I/O assignment and SC table	R	512
DM 254	AL_DP_D	Display of machine usage & DP diagnostics	R	512
DM 255	BF_Globa	DM, valid for all BTs	R	512

Fig. 4-5 List of Data Modules

4.3.4 System Configuration Table for CL500

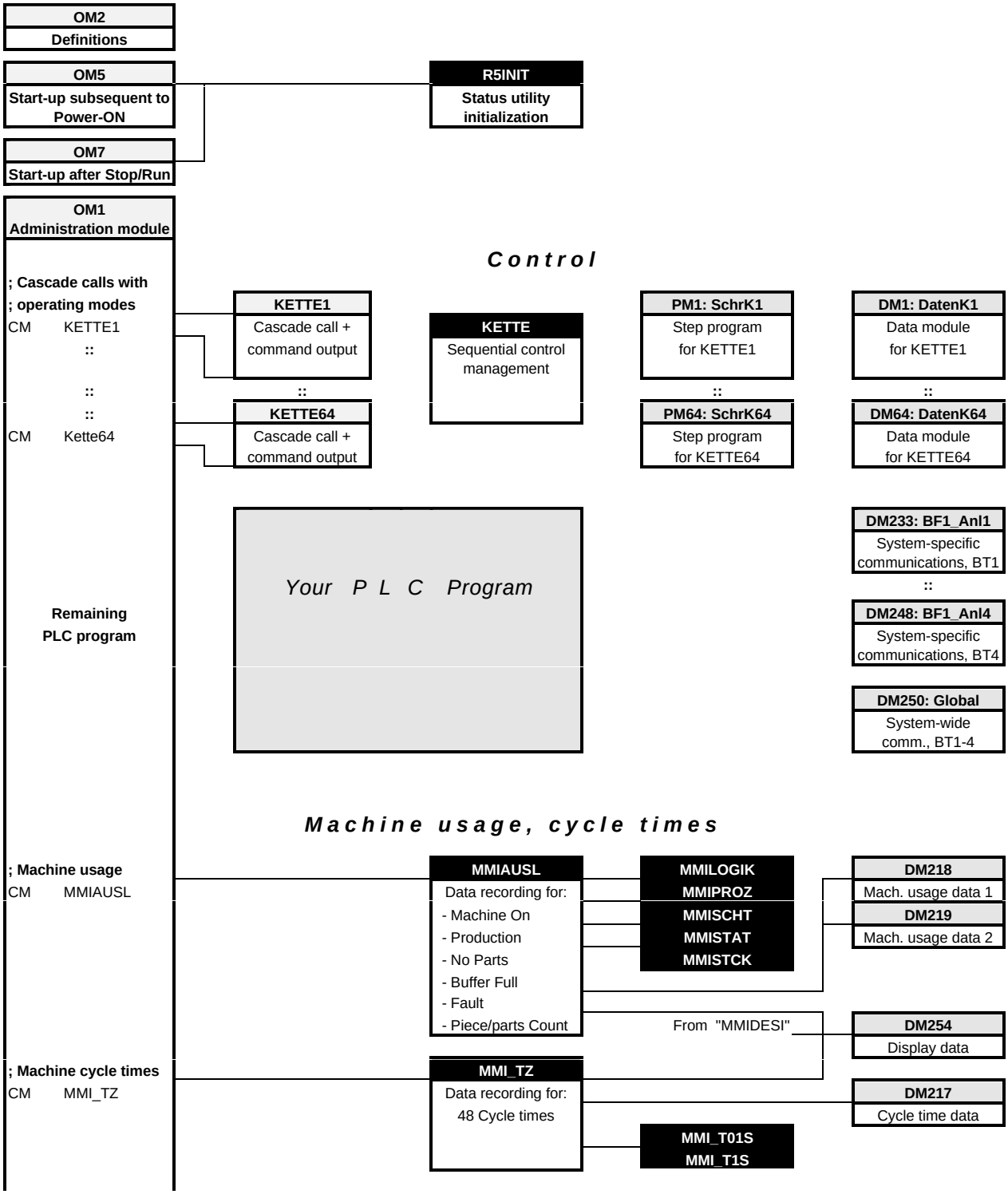
The system configuration described below comprises a suggestion, the scope of which may be expanded or otherwise modified. With the use of a CL400, all ZS entries will be deleted.

NOTE:
If the R500P is entered in another row of the list, this will cause the module number to change. As a consequence, the parameterization of the R5INIT standard PM (if in use) must also be suitably adapted in the OM5 and OM7 start-up modules.

Module no.	Modules	In Rack	Block Addr.	Synch. method	Rem. STOP	IR module	Periph. Addr.	I/O Length	EI/EO Length
1	ZS510	J	0	00000000	N	0	0	64	64
2	ZS510	N	8	00000000	N	0	64	64	64
3	ZS510	N	16	00000000	N	0	128	64	64
4	ZS510	N	24	00000000	N	0	192	64	64
5									
6									
7	R500P	J	40						
8									
9									
10									

Fig. 4-6 SC Table

4.4 Principal Program Structure



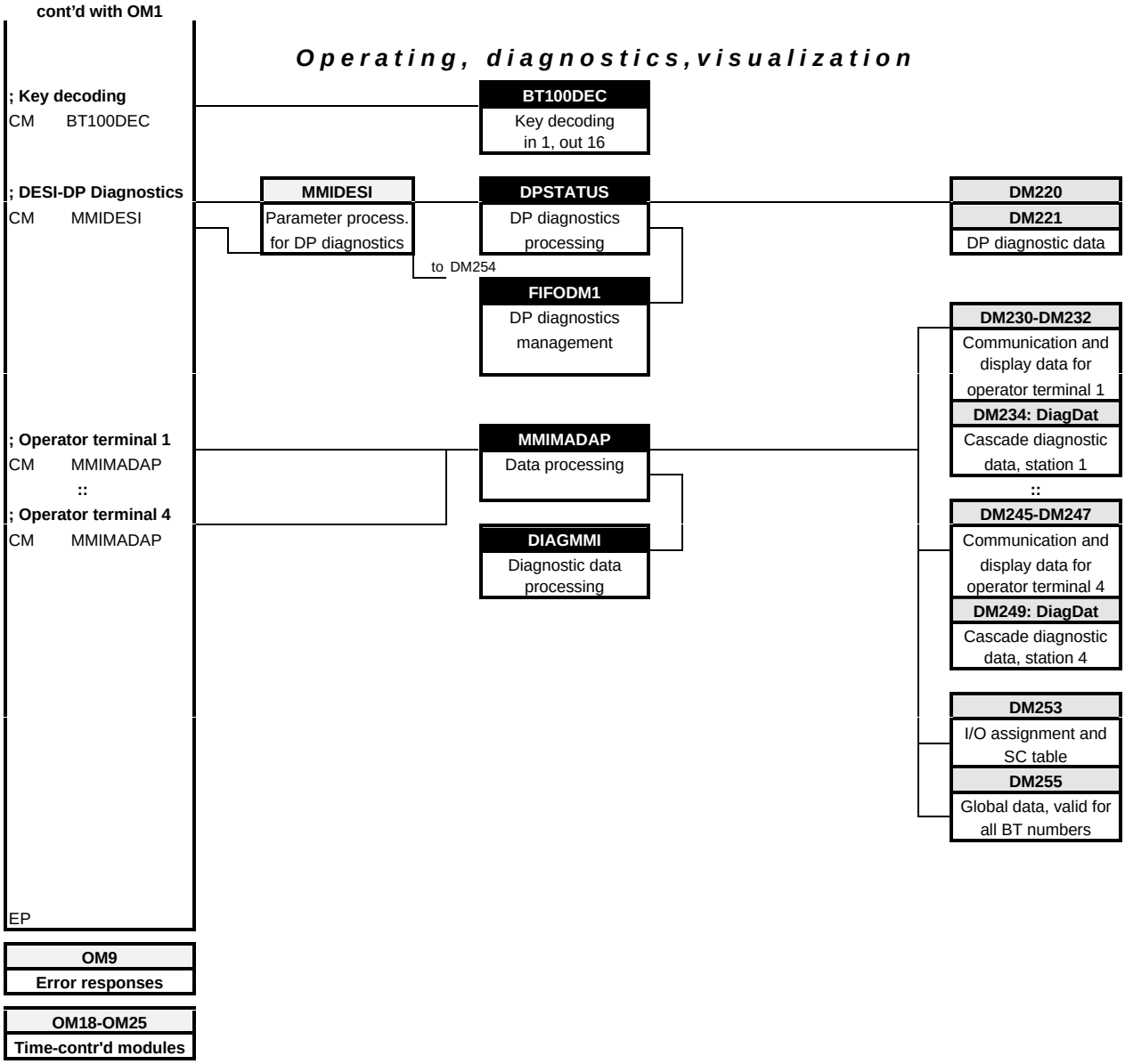


Fig. 4-7 MMI-MADAP Software — Overall Program Structure

OM2 Definition module

The OM1 comprises a system initialization table containing default definitions governing the operation of the PLC. It is essential to note that it is not permitted to insert rows (lines) into or delete rows from the table. Instead, the existing values may be merely modified, i.e., overwritten.

Entries enabling recognition of a cycle time error

```
;DW 2: Initialization flag (entries permitted)
;---------------------------------
;               Entry 0 = DO NOT verify or execute function
;               Entry 1 = Verify or execute function
;
DEFW  W       K0000000000000100bin

;               Bit 1 = Check assignment list
;               Bit 2 = Check nominal cycle time (not effective for ZS500 < version 201!!)
;               Bit 9 = Copy data module in data buffer
;


;DW 5: Maximum cycle time (not effective for ZS500 < version 201!!)
;-------------------------------------
;               Entry to be a multiple of 10 ms time base of K1D and K200D
;               10 ms through 2000 ms) for cycle time monitoring
;               Execution of function when bit 1 of DW2 = 1.
;
DEFW  W       K200D
;
```

Entries ensuring time-controlled processing steps

```
;               Definition of time OMs (entries permitted)
;               ========================
;               Entries to constitute multipliers of 10 ms basic time of K1D through K65535D
;               e.g. K0D  = NO time-controlled processing
;                    K11D = 11 x 10 ms = 110 ms processing time interval
;
;DW 11:         Time OM18
;-----------------------------------
DEFW  W       K0D

;DW 12:         Time OM19
;-----------------------------------
DEFW  W       K0D

;DW 13:         Time OM20
;-----------------------------------
DEFW  W       K0D

;DW 14: Time OM21
;-------------------------
DEFW  W       K0D
```

```
;*****************************************************************************
; In the case of the ZS500 < version 201, the time-controlled OMs, i.e.,
; OM22 through OM25, must first be released by the program.
;*****************************************************************************

;DW 15:          Time OM22      (for ZS500 < version 201, release by program required)
;--------------------------------
DEFW  W         K0D

;DW 16:          Time OM23      (for ZS500 < version 201, release by program required)
;--------------------------------
DEFW  W         K0D

;DW 17:          Time OM24      (for ZS500 < version 201, release by program required)
;--------------------------------
DEFW  W         K0D

;DW 18:          Time OM25      (for ZS500 < version 201, release by program required)
;---------------------------------
DEFW  W         K0D
```

OM5 and OM7 Start-up modules

The purpose of the R5INIT module call in the OM5 and OM7 organization modules is to initialize the status utility for the R500P (through version 6, check front panel marking), and to generate the CPU Stopped and Communication Failure system messages for display on the operator terminal.

Effective with version 7 of the R500P (refer to front panel marking), the linking of the R5INIT module, and of the OM5 and OM7 is no longer required because the status utility is now supported directly.

The OM5 start-up module is processed subsequent to each Power-On, and the OM7 after each STOP/RUN command.

As the preprogrammed contents are identical for both modules, they are shown here only once.

```
;**** OB5 / OB7 Start-up module after Power-On / STOP/RUN ****

DEF     xxxx,-InitErg; any operand except M230-M254

; Initialization of R500P computer interface module for processing
; the status utility for evaluating the CPU Stopped and Communication
; Failure system messages.

; Upon initialization, the module number (row number in SC Table)
; of the R500P must be specified. Parameterization is restricted to
; modules that are physically present. A module that is not physically
; present must be represented by a parameterization of KFFFFH.

CM      -R5INIT,6
P0   W  K7D          ; R500P module number (row 7 of SC table)
P1   W  KFFFFH       ; No module present
P2   W  KFFFFH       ; No module present
P3   W  KFFFFH       ; No module present
P4   W  KFFFFH       ; No module present
P5   W  -InitErg     ; Result of initialization

;Error response
L    W  -InitErg,A   ; Read result
CPLA W  K0,A         ; Fault-free initialization
JPZ     -Init_iO
HLT                  ; Halt on error
     -Init_iO

; additional application program

EM
```

Fig. 4-8 OB5 / OB7 Start-up Modules

OM9 Error module

The OM9* represents an error module within the PLC. The following program part is required as a standard function:

```
;**** OM9, Data processing for displaying causes of PLC STOP conditions ****

; MMI-MADAP Error return upon controller standstill
;================================================
DEF     DB255,-GlobalDB
DEF     SM28.0,-AdrFehl    ; Addressing error
DEF     SM28.1,-PBparaF    ; PM parameter error
DEF     SM28.2,-n_extBst   ; Non-existent PM was called
DEF     SM28.3,-BstStack   ; Module stack error
DEF     SM28.4,-AST_U      ; Application stack underrun error
DEF     SM28.5,-AST_Ue     ; Application stack overrun error
DEF     SM29.2,-SBparaF    ; Parameter error in system commands
DEF     SM29.5,-noDB       ; No DM active at this time
DEF     SM29.7,-ZyklZF     ; Cycle time error
DEF     D86,-SPS_Z_F

CM      -GlobalDB          DB255
L    W  -SPS_Z_F,A         D86
A    B  -AdrFehl           SM28.0   Addressing error
=OM  B  A.8
A    B  -PBparaF           SM28.1   PM parameter error
=OM  B  A.9
A    B  -n_extBst          SM28.2   Non-existent PM was called
=OM  B  A.10
A    B  -BstStack          SM28.3   Module stack error
=OM  B  A.11
A    B  -AST_U             SM28.4   Application stack underrun error
O    B  -AST_Ue            SM28.5   Application stack overrun error
=OM  B  A.12
A    B  -SBparaF           SM29.2   Parameter error in system commands
=OM  B  A.13
A    B  -noDB              SM29.5   No DM active at this time
=OM  B  A.14
A    B  -ZyklZF            SM29.7   Cycle time error
=OM  B  A.15
T    W  A,-SPS_Z_F         D86

; if applicable, additional error response program

EM
```

Fig. 4-9 OB9 Error Module

✉ **BOSCH documentation reference**

CL400 / CL500 Operations List, Software Manual	No. 1070 072 127

OM18 through OM25 time-controlled processing modules

The modules for time-controlled processing are supplied merely in a preparatory condition, and require programming and, if applicable, activation within OM2.

```
; OM18 - OM25 Time-controlled modules
;=====================================

; Application program

EM
```

Fig. 4-10 OM18 through OM25 Time-controlled Modules

4.5 Process Control

4.5.1 Definitions

Introduction

This section describes the process control management as well as its operating modes, process management and command output. The appropriate sequential control, diagnostics and display program modules are provided.

Specifications

To guarantee orderly processing sequences and/or unambiguous diagnostic results, the MMI-MADAP software packet shall be subject to the following specifications:

Modules

The KETTE program module manages up to

- 64 process sequences, encompassing
- 128 steps each,
- with one active step in each cycle.

The following are permanently assigned to the referred process sequences:

– Program modules	PM1 through PM64
	as process sequences
– Data modules	DM1 through DM64
	as cascade data modules

The respective PM and DM numbers correspond to the cascade numbers.

Markers

Within the marker range, the markers listed below are assigned a permanent function.

Symbol	Address	Function
BEFA	M255.0	Assigned to command output
WSB	M255.1	Assigned to step-on functions
STOEM	M255.2	Fault marker; STOEM = LOW indicates fault present
HALBAUTO	M255.4	Step-on in inching mode even without S+1 transition
WZT_HLT	M255.5	Wait time halt
WZT	M255.6	Wait time statusLOW: Wait time running HIGH: Wait time expired
UEKONTR	M255.7	Monitoring time check If UEKONTR = HIGH, expiration of monitoring time will not trigger diagnostics.
VERZW	M242	Branch address (word) Step number within KETTE program module to which branching will take place if WSB = 1_{bin}.

Fig. 4-11 Control Marker Definitions

4.5.2 Programming

Program structure

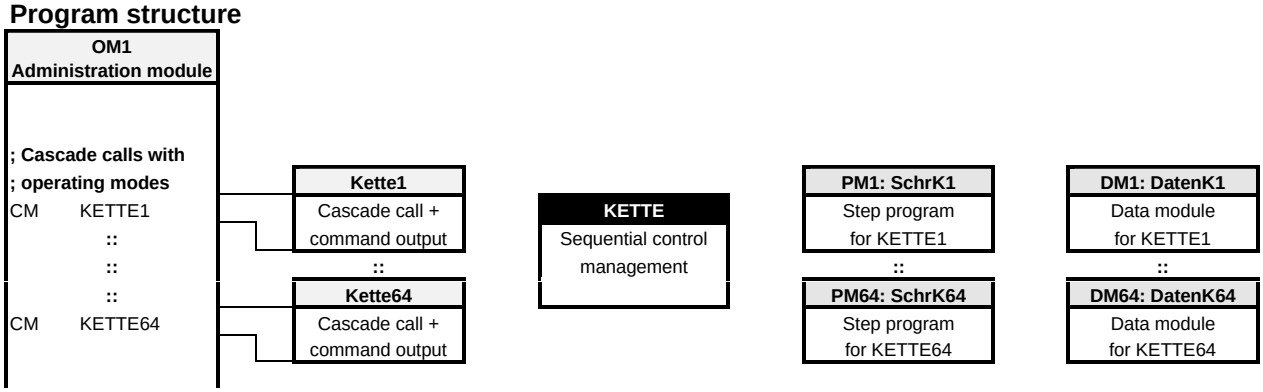


Fig. 4-12 Program Structure — Sequential Control

Module call

The KETTE cascade sequence management module is called within the KETTE1 through KETTE64 program module and subsequently services the defined cascades.

The operating mode information is transferred to the corresponding parameters of the KETTE module. In order to enable the formation of functional cascade groups, it is possible to leave gaps when assembling up cascade data modules.

All cascades that are not called up in the OM1 administration module, will be skipped, and processing will continue with the subsequent cascade. In the event that step modules are to be excluded from processing, the associated cascade module must be declared as a comment upon module call.

The module call-up is programmed as follows:

```
CM      -KETTE,4
;                    +---+
P0   W  -PB/DB     ; <   !  Cascade and DM number (1≤n≤64)
P1   W  -BETR      ; <   !  Operating mode selection
P2   W  -KUE       ; <   !  Time value of monitoring time
P3   W  -KWA       ; <   !  Time value of monitoring time
;                    +---+
```

Fig. 4-13 Module Call for KETTE Program Module

Refer to Chapter 4, "Interfaces — PLC <–> Operator Terminals," for the following information:
Data module contents DM1 through DM64, i.e. DB_K01 - DB_K64: cascade information for KETTE1 through KETTE64

Example

The supplied MMI-MADAP standard software contains a programming example for the module call in the form of a network in the OM1 You can copy the example from there into your program.

Parameter description

☞ **Scratch markers must not be used in place of parameters!**

KETTE

P0 W -PB/DB

Parameter P0 supplies the KETTE cascade sequence management module with the current cascade number **n** for the following modules:

- Number of SCHRK**n** step module
- Number of DM**n** cascade data module

Each processed cascade **n** utilizes the cascade data module **n** to store variable data, e.g. current step, monitoring time and wait time.

P1 W -BETR

Parameter P1 supplies the KETTE module with the operating mode selection.

- P1.0 Manual mode
- P1.1 Inching mode
- P1.2 Automatic mode
- P1.3 Start
- P1.4 S + 1
- P1.5 Set Step
- P1.6 Halt acknowledgement
- P1.7 Reset
- P1.8 Halt
- P1.9 Synchronizing
- P1.10
- P1.11 Generate cascade data module
- P1.12 Step-on condition in same cycle
- P1.13
- P1.14 WSB does not reset command output
- P1.15 Fault acknowledgement

The KETTE program module writes the operating mode data into data word D6 (nnBAWAHL) of the associated cascade data module. In Manual mode, Start and Set Step will be adopted by D6 only subsequent to pressing the movement keys on the operator terminal.

☞ **In cascade data modules, MMI-MADAP sets data bit D24.2 (MADAP identifier bit). Thus in Manual mode, the Start operating mode bit on parameter P1.3 will be AND-linked with the MADAP Start operating mode bit D30.3, and only then returned in data word D6 (selected operating mode).**

P2 W -KUE

Parameter P2 supplies the KETTE module with the monitoring time value. This will remain identical for every step unless it is newly defined in the individual steps.
The time base is always 100 ms. Thus an example default:

- P2 W $K20_{dez}$

will result in a monitoring time of 20 x 100ms = 2 s

The maximum monitoring time is 109 minutes.
The monitoring time is transferred to data word D20 of the associated cascade data module.

P3 W -KWA

Parameter P3 supplies the KETTE module with the wait time value. This will remain identical for every step unless it is newly defined in the individual steps.
The time base is always 100 ms. Thus an example default:

- P3 W $K15_{dez}$

will result in a wait time of 15 x 100ms = 1.5 s

The maximum monitoring time is 109 minutes.
The wait time is transferred to data word D18 of the associated cascade data module.

Register contents

Neither the PLC registers A, B, C and D nor the control flags (e.g. RES, Carry) will be retained beyond the module call. Upon returning from the KETTE program module to the calling module, the registers are given the following defined contents:

KETTE

Reg.	Contents
A	Version number of KETTE module
B	Error codes, if cascade data module not generated
C	No relevance
D	No relevance

Fig. 4-14 KETTE Register Contents

Error codes in register B

Accu B contents	Troubleshooting
FFFFhex	Cascade data module not available
0001hex	P0 of KETTE module is 0
0002hex	P0 of KETTE module is > 64_{dez}
0004hex	Cascade data module too short < 96_{dez}

Fig. 4-15 Error Code in KETTE Register B

4.5.3 Operating Modes

The operating mode is transferred via parameter P0 to the KETTE module, from where it is written into data word D6 of the associated cascade data module.

Manual mode/Setup

Function

Manual operation of steps occurs in accordance with the conditions valid for the manual branch.
The step is entered in data word D14 of the associated cascade data module, and adopted as a current step by means of Set Step (D6.5=HIGH) in D12.

Command output occurs if the following applies:

- The conditions of the manual branch are met (BEFA command output = HIGH
  and WSB step-on condition = LOW)

AND

- The Start (D6.3) = HIGH

The command output occurs via the data bits of data words D80 through D94 of the associated cascade data module (refer to cascade data module).

No **step-on** will occur.

Both the Manual mode and Start bits are set HIGH.
In the event that a movement function is effected at the operator terminal, the following will occur automatically:

- Activation of cascade number of the associated cascade data module,
- Entry of step number in data word D14,
- Set Step (D6.5),
- Start (D6.3) is set to HIGH.

Diagnostics

Display in cascade information, **H** for Manual mode, in the respective cascade.

Display of all criteria of manual branch, i.e., either of BEFA command output that was not met, or of WSB step-on branch.

Monitoring and wait time values are loaded along with the default values but are not started.

No fault message is returned.

Inching mode/Single-step mode

Function

Step-by-step processing of steps in accordance with conditions applying to automatic branch.

Command output occurs if the following applies:

- The conditions of the manual branch are met (BEFA command output = HIGH
  and WSB step-on condition = LOW)

AND

- The Start (D6.3) = HIGH

The command output occurs via the data bits of data words D80 through D94 of the associated cascade data module (refer to cascade data module).

No automatic step-on will occur.

With a positive transition on S+1 (P1.4), and WSB step-on condition being met, the advance to the next step will occur.

Programming

Inching mode bit is set HIGH. When Start = HIGH, the current step will be processed, and a positive transition at S+1 will cause an advance to the next step.

Diagnostics

Display in cascade information, **T** for Inching mode, in the respective cascade.

Display of all criteria of automatic branch, i.e., either of BEFA command output that was not met, or of WSB step-on branch.

Indication of faulty cascade.

Monitoring time and wait time elapse on the basis of the default values.

Semi-automatic mode

Function

Semi-automatic processing of steps in accordance with conditions applying to automatic branches.

Command output occurs if the following applies:

- The conditions of the manual branch are met (BEFA command output = HIGH
  and WSB step-on condition = LOW)

AND

- The Start (D6.3) = HIGH

The command output occurs via the data bits of data words D80 through D94 of the associated cascade data module (refer to cascade data module).

Step-on occurs automatically when WSB step-on condition is met, and when -HALBAUTO marker (M255.4) is set. The step-on will end on the step on which the -HALBAUTO marker is = LOW, or when the WSB step-on condition has not been met.

With a positive transition on S+1 (P1.4), and with WSB step-on conditions met, the next program sequence will be processed up to the reset -HALBAUTO marker.

Programming

Inching mode bit is set HIGH. When Start = HIGH, the current step will be processed, and a positive transition at S+1 will cause Semi-automatic mode to be started.

Diagnostics

Display in cascade information, **T** for Inching mode, in the respective cascade.

Display of all criteria of automatic branch, i.e., either of BEFA command output that was not met, or of WSB step-on branch.

Indication of faulty cascade.

Monitoring time and wait time elapse on the basis of the default values.

Automatic mode

Function

Automatic processing of steps in accordance with conditions applying to automatic branch.

Command output occurs if the following applies:

- The conditions of the manual branch are met (BEFA command output = HIGH
  and WSB step-on condition = LOW)

AND

- The Start (D6.3) = HIGH

The command output occurs via the data bits of data words D80 through D94 of the associated cascade data module (refer to cascade data module).

If WSB step-on condition = HIGH, **automatic step-on** will occur.

Programming

Set Automatic mode (D6.2) and Start (D6.3) bits to HIGH.

Diagnostics

Display in cascade information, **A** for Automatic mode, in the respective cascade.

Display of all criteria of automatic branch, i.e., either of BEFA command output that was not met, or of WSB step-on branch.

Indication of faulty cascade.

Monitoring time and wait time elapse on the basis of the default values.

D6, Selected operating mode

In addition to the operating modes discussed in the preceding section, data word D6 of the cascade data module contains further operating mode selection information.

D6, nnBaWahl

This data word is written to via the -BETR (P1) parameter of the KETTE module.

D6.0	Manual mode **H**
D6.1	Inching mode **T**
D6.2	Automatic mode **A**
D6.3*****	Start **s**
D6.4	S + 1
D6.5*****	Set Step
D6.6	Fault acknowledgement
D6.7	Reset **r**
D7.0	Halt **h**
D7.1*****	Synchronization
D7.2	
D7.3	Cascade data module generation
D7.4	No step-on in same cycle
D7.5	
D7.6	WSB does not reset BEFA (Manual mode only)
D7.7	Fault requires acknowledgement
*****	Observe bit description

Fig. 4-16 D6 — Selected Operating Mode

Bit description

D6.0, Manual mode

Select Manual mode

D6.1, Inching mode

Select Inching mode

D6.2, Automatic mode

Select Automatic mode

D6.3, Start

Start/Command enable

The bit is valid for all operating modes, and is statically transferred to parameter P1 of the KETTE module.
If Start = LOW, the following actions will occur:

- BEFA command output is deleted,
- the monitoring time is stopped, and
- the wait time continues to elapse.

Prior to generating the cascade data modules, and for the purpose of synchronizing in Automatic mode, the Start must be deleted.

☞ **In cascade data modules, MMI-MADAP sets data bit D24.2 (MADAP identifier bit). Thus in Manual mode, the Start operating mode bit on parameter P1.3 will be AND-linked with the MADAP Start operating mode bit D30.3, and only then returned in data word D6 (selected operating mode).**

D6.4, S+1

Executing the next step

When in Inching mode, a positive transition of this bit and satisfied WSB step-on condition (WSB = HIGH), the next step will be executed.

When in Semi-automatic mode, a positive transition of this bit and satisfied WSB step-on condition (WSB = HIGH), and with
-HALBAUTO marker set, the next program sequence will be processed up to the reset -HALBAUTO marker.

D6.5, Set step

Adopting preselected step number

In Manual mode, the step prepared in D14 is adopted into the active step (D12), and then executed.

☞ **In cascade data modules, MMI-MADAP sets data bit D24.2 (MADAP identifier bit). Thus in Manual mode, the Set Step operating mode bit on parameter P1.5 will be OR-linked with the MADAP Set Step operating mode bit D30.5, and only then returned in data word D6 (selected operating mode).**

D6.6, Halt acknowledgement

Manual error acknowledgement

Effective only if D7.7 = HIGH (manual fault acknowledgement).

A positive transition on this bit acknowledges a fault (cascade stop) that was triggered by a monitoring time-out or by the reset fault marker.

Monitoring and wait time values are loaded along with the default values but are not started.

D6.7, Reset

Cascade reset

If D6.7 = HIGH, the following actions will occur:

- Deletion of active step,
- Reinitialization of cascade sequence,
- Recreation of cascade data module.

Subsequent to a Reset, step 1 is prepared.

D7.0, Halt

Halting cascade processing

If the bit is set, the cascade is halted, and the processing of the current step continues. If D7.0 = HIGH, the following will occur:

- BEFA command output is returned,
- Monitoring and wait times are halted.

D7.1, Synchronization

Synchronizing cascade

This action is possible in manual and Automatic mode. In the case of Automatic mode, D6.3 = LOW will be additionally required.

If this bit is set, the KETTE cascade management module will search the cascade for satisfied conditions, and subsequently synchronize the operating mode in accordance with the operating mode.

The conditions for an effective synchronization are as follows:

- BEFA command output = HIGH and
- WSB step-on condition = LOW

For all steps for which the synchronization requirements have been met, the corresponding bit in data block D48 through D62 is set.

In the event that, in Automatic mode, exactly 1 step is found for which the synchronization conditions are satisfied, this step will be prepared.

In the case of AND-links, because the cascades are examined independently of each other, there are limitations to the synchronization to Automatic mode.

☞ **In cascade data modules, MMI-MADAP sets data bit D24.2 (MADAP identifier bit). Thus in Manual mode, the Synchronization operating mode bit on parameter P1.9 will be OR-linked with the MADAP Synchronization operating mode bit D31.1, and only then returned in data word D6 (selected operating mode).**

Function description:

If a synchronization procedure is initiated via parameter P1.9 or D31.1, and if subsequently the D7.1 = HIGH, the synchronization result of this cascade (D9.1, No Synchronization Possible and/or D9.2, More Than One Synch Step) can already be interpreted. A subsequent synchronization initiation will be interpreted only if the system undergoes another – transition-controlled – LOW-to-HIGH change.

D7.3, Learning

Generating cascade data module

When bit D7.3 is set, subsequent to loading the program, and following a Power-On or Reset command, the KETTE cascade management module will generate the data for the cascade modules. This cascade-specific data is determined on the basis of the corresponding step modules and of the parameterization in the KETTE module. As a prerequisite, D6.3 = LOW must be true.

D7.4, Step-on (when using WinSPS control language, D7.4 = 0 must be true!)

No step-on in same cycle

Automatic mode only

- When D7.4 = LOW is true, only one step will be processed in each cycle.
- When D7.4 = HIGH is true, and WSB step-on condition is met, the next step will be activated during the same cycle.

D7.6, WSB fails to reset BEFA

Manual operation only

- When D7.6 = HIGH is true, even a satisfied WSB step-on condition will not reset the corresponding BEFA command output.
- When D7.6 = LOW is true, a satisfied WSB step-on condition will reset the associated BEFA command output.

D7.7, Acknowledgement

Fault acknowledgement

If this bit is set, an occurring cascade fault must be acknowledged by bit D6.6.

With bit D7.7 reset, and WSB step-on condition met, the cascade will auto-acknowledge.

Priority ranking of operating mode bits

In the event that several operating mode bits are simultaneously selected in data word D6 of the cascade data module, the processing will be subject to the following priority ranking:

1. Reset	highest priority
2. Halt	↓
3. Start	↓
4. Manual	↓
5. Inching	↓
6. Automatic	lowest priority

Fig. 4-17 Operating Mode Priorities

D8, Confirmed operating mode

Subsequent to processing the KETTE proigram module, data word D8 of each cascade data module contains the return confirmation of the operating mode.

D8, nnBaMldg

D8.0	Manual mode **H**
D8.1	Inching mode **T**
D8.2	Automatic mode **A**
D8.3	Start, all operating modes **s**
D8.4	Automatic or Inching mode enabled
D8.5	
D8.6	Wait time active
D8.7	Reset, cascade reset **r**
D9.0	Halt, cascade halted **h**
D9.1	No synchronization possible
D9.2	More than one step with BEFA = HIGH, synchronizati-
D9.3	Cascade data module generated
D9.4	
D9.5	
D9.6	Pulse fault
D9.7	Static fault

Fig. 4-18 D8 — Confirmed Operating Mode

Bit description

D8.0, Manual mode

Cascade is in Manual mode.

Display **H** appears in cascade information of operator terminal.

D8.1, Inching mode

Cascade is in Inching mode.

Display **T** appears in cascade information of operator terminal.

D8.2, Automatic mode

Cascade is in Automatic mode.

Display **A** appears in cascade information of operator terminal.

D8.3, Start

Cascade has received the start bit.

Display **s** appears in cascade information of operator terminal.

D8.4, Automatic/Inching

Cascade is in Automatic or Inching mode.

This bit is used to select whether the Manual or Automatic branch is to be processed in the cascade module.

If D8.4 = HIGH is true, Automatic or Inching mode is enabled.

If D8.4 = LOW is true, Manual mode is enabled.

D8.6, Wait time active

If this bit is set, this indicates that the wait time for this step has expired.

Prior to calling this step, the cascade management module writes the status of this bit into the WZT wait time marker (M255.6), thus making the wait time available to the steps as a diagnosable operand.
If the wait time has expired, the query:

- A B -WZT

will return HIGH

D8.7, Reset cascade

The cascade is reset, and the active step deleted (subsequent to Reset, D12 = HIGH is true).

D9.0, Cascade halted

Cascade is in Halt status.
This operating mode is activated by:

- D7.0 = HIGH (Halt)
- Reset of M255.2 fault marker, OR
- Fault in Automatic mode, with expired monitoring time (only with manual acknowledgement via D7.7 = HIGH).

D9.1, No Synchronization Possible

In none of the scanned steps, the

- BEFA command output = HIGH

and

- WSB step-on condition = LOW

synchronization conditions have been met.

D9.2, More Than One Synch Step

During synchronization in Automatic mode, more than one step was found in which BEFA = HIGH and WSB = LOW were true. The Auto Continue synchronized start of Automatic mode is not possible.

D9.3, Cascade data module generated

With D9.3 = HIGH being true, the learning or generating of the data module will be concluded.

D9.6, Pulse fault

Returns a pulse for a given PLC cycle in the event that a fault was recognized.

Fault criteria:

- Reset fault marker

OR

- Monitoring time expired.

D9.7, Static fault

Returns a static signal upon fault detection (criteria as for D9.6).

The bit is reset by:

- Fault acknowledgement

OR

- An action subsequent to a change in operating mode (e.g. Set Step in Manual mode).

4.5.4 Step module

For each cascade, a -SCHRKn ($1 \leq n \leq 64$) step module is generated.

The step module contains, for all controller types:

- the jump distributor to the active step

AND

- a maximum of 128 steps.

Step module processing always begins with the jump distributor line belonging to the active step. From here, the jump to the actual step conditions is effected. Only the active step is processed.

As a rule, the step consists of 2 independent program parts:

- the Manual part with the manual conditions, and
- the Automatic part with the Automatic and/or Inching conditions.

Both program parts must be individually concluded with an **EM** end-of-module instruction.

If the same conditions apply to Manual and Automatic mode, only one part will have to be programmed.

Both the Manual and Automatic part are again divide into:

- a BEFA command output branch

AND

- a step-on condition branch

An arbitrary number of BEFA command output and WSB step-on branches can be programmed. However, the instruction:

= - BEFA and/or
= - WSB

must be the last instruction in a given branch.

When diagnosing processing faults, the display always indicates the first BEFA branch for which the conditions are not met, starting from the start of the step. If conditions for all BEFA branches are met, the first non-satisfied WSB branch will be displayed.

Rules for step programming

- Only unconditional jumps may be programmed in the jump distributor! The jump sequence must match the sequence of jump destinations!
- Nor other instructions are permitted prior to the jump distributor. The jump distributor must contain
  SP [A]
  as its first instruction!
- The jump distributor may not be interrupted by other instructions!
- Only jump destinations or comments may be inserted between the command
  SP [A]
  and the first jump destination. There are no control characters for program segmentation permitted here!
- The number of jumps in the jump distributor must correspond to the number of scheduled steps!
  Otherwise, no diagnostics will be possible!
- A module call from within a -SCHRKn step module is not permitted!
- The jump distributor and/or the jump destinations in the SCHRKn step module must not be changed with the use of the Replace function!
- Non-bit instructions (e.g. links and time manipulation) must be programmed at the beginning of the branch.
- Scratch markers and special markers in step modules may cause unpredictable diagnostic results (wait time, too, is a scratch marker).
- In the event that different criteria are programmed for the Manual and Automatic branch in a given step, both Manual branch and Automatic branch must be concluded with an EM end-of-module instruction.

Example

The supplied standard software provides a programming example for the step module for KETTE1. This can be integrated into your own program and modified/expanded to suit your requirements.

Example of step module structure:

```
;Jump distributor to active step (mandatory at start of module)
JP      [A]               ; Reads active step, and commences jump
                          ; to step
JP      -Schritt1         ; to Step 1
JP      -Schritt2         ; to Step 2

     -Schritt1
DEF     D18,-WaZeit       ; Wait time value
DEF     D20,-UeZeit       ; Monitoring time value

; Changing wait time for this step
L    W  K80,A             ; 8s wait time (Wait time > Monitoring time: Fault)
T    W  A,-WaZeit         ; Wait time data word
; Changing monitoring time for this step
L    W  K30,A             ; 3s monitoring time (Wait time > Monit time: Fault)
T    W  A,-UeZeit         ; Monitoring time data word

; Automatic or Inching enabled?
L    W  D8,A              ; Operating mode confirmation from KETTE module
A    B  A.4               ;
JPC     -Auto1            ; at Automatic conditions of Step 1

; Manual conditions for Step 1
A    B  -K1S1HBef         E0.0    Kette 1, Step 1, BEFA Manual condition
=    B  -BEFA             M255.0  command output

A    B  -K1S1HWSB         E0.1    Kette 1, Step 1, WSB Manual condition
=    B  -WSB              M255.1  Step-on condition

EM                        ; =WSB/EM: Defined formalism at end of step.

     -Auto1
 ; Automatic conditions for Step 1
U    B  -K1S1ABef         E1.0    Kette 1, Step 1, BEFA Automatic condition
=    B  -BEFA             M255.0  Command output

U    B  -K1S1AWSB         E1.1    Kette 1, Step 1, WSB Automatic condition
O    B  -WZT              M255.6  Wait time marker status
=    B  -WSB              M255.1  Step-on condition

EM                        ; =WSB/EM: Defined formalism at end of step

     -Schritt2
; Automatic conditions for Step 2
A    B  -K1S2ABef         E1.2    Kette 1, Step 2, BEFA Automatic condition
=    B  -BEFA             M255.0  Command output

A    B  -K1S2AWSB         E1.3    Kette 1, Step 2, WSB Automatic condition
O    B  -WZT              M255.6  Wait time marker status
=    B  -WSB              M255.1  Step-on condition

EM                        ; =WSB/EM: Defined formalism at end of step.
```

Fig. 4-19 Example of Step Module

If a given step is identical with regard to Manual and Automatic mode movements, the operating mode selection may be omitted (as in Step 2 in this example).

4.5.5 Command Output

It makes good sense to arrange the command output immedately following the call-up of the KETTE cascade management module.

This purpose is served by the KETTE**n** module (with 1 ≤ **n** ≤ 64) which handle the command output subsequent to completed processing of the KETTE cascade management module.

This requires the corresponding data word (D80 through D94) for command output to be loaded.

When using the KETTE module with version 2.5 and higher, the command output can also handled via data word D16.

Example of utilization of data words D80 through D94

Command output for a cascade (KETTE1) containing two steps:

```
; Parameterization and module call
DEF     K1,-PB/DB
DEF     K50,-KUE
DEF     K40,-KWA

CM      -KETTE,4
;                   +---+
P0   W  -PB/DB    ; <   !  Cascade number
P1   W  -BETR     ; <   !  Operating mode
P2   W  -KUE      ; <   !  Monitoring time
P3   W  -KWA      ; <   !  Wait time
;                   +---+
;
; Command output programming
DEF     DB1,-DB-Kett1
DEF     D80,-BEFA1-16     ; Command execution bits, Step 1 thru 16
DEF     A0.0,-BEFAUS1     ; Command output Step 1
DEF     A0.1,-BEFAUS2     ; Command output Step 2

CM      -DB-Kett1

L    W  -BEFA1-16,A

; Step 1
A    B  A.0               ; BEFA command output Step 1
=    B  -BEFAUS1          ; Enable output

; Step 2
A    B  A.1               ; BEFA command output Step 2
=    B  -BEFAUS2          ; Enable output

EM
```

Fig. 4-20 Cascade Command Output via Data Words D80 through D94

Subsequent to calling the KETTE module, these command output program instructions must be repeated for all active cascades and for all steps in the corresponding KETTE1 through KETTE64 modules.

The supplied standard software contains a programming example for the cascade management module of Kette1. You can copy this example into your own program, and modify or expand it to suit your requirements.

Example of utilization of data word D16

This version is supported by the WinSPS programming device software with SFC, and requires KETTE program module, version 2.5 and higher.

Command output for a cascade (KETTE1) containing two steps:

```
; Parameterization and module call
DEF     K1,-PB/DB
DEF     K50,-KUE
DEF     K40,-KWA

CM      -KETTE,4
;                   +---+
P0   W  -PB/DB    ; <   !  Cascade number
P1   W  -BETR     ; <   !  Operating mode
P2   W  -KUE      ; <   !  Monitoring time
P3   W  -KWA      ; <   !  Wait time
;                   +---+
;
; Command output programming
DEF     DB1,-DB-Kett1
DEF     D16,-BEFAUSG      ; Command execution word for all steps
DEF     A0.0,-BEFAUS1     ; Command output Step 1
DEF     A0.1,-BEFAUS2     ; Command output Step 2

CM      -DB-Kett1

L    W  -BEFAUSG,A

; Step 1
CPLA W  K1,A
A    B  SM31.7            ; A=1: BEFA command output Step 1
=    B  -BEFAUS1          ; Enable output

; Schritt 2
VGLA W K2,A
A    B  SM31.7            ; A=2: BEFA command output Step 2
=    B  -BEFAUS2          ; Enable output

EM
```

Fig. 4-21 Cascade Command Output via Data Word D16

4.5.6 Cascade Data Module

The data modules DM1 through DM64 handle the sequential control, and furnish all data essential to system control. The data module must be created for each step module.

	Symbol	**Explanation**	**Data format**	Entry caused by: K: KETTE M: MMI-MADAP A: USER
D00	nnFehler	Error bits	binary	K
D02	nnKettNr	Cascade no. n (1-64)	decimal	K
D04	nnSchAnz	Number of steps in cascade	decimal	K
D06	nnBaWahl	Operating mode selection	binary	K
D08	nnBaMldg	Confirmed operating mode	binary	K
D10	nnSchr-1	Step number, preceding step	decimal	K
D12	nnSchr.	Step number, current step	decimal	K
D14	nnSchr.S	Step number, Set Step	decimal	M
D16	nnBEFA	Command output for all steps	decimal	K
D18	nn-KWA	Wait time, actual value	dec. x 100 ms	K/A
D20	nn-KUE	Monitoring time, actual value	dec. x 100 ms	K/A
D22	nnINT0	internal use		
D24	nnINT1	internal use Bit 2 MADAP active bit	binary	M
D26	nnINT2	internal use		
D28	nnINT3	internal use		
D30	nnBa_Ext	Operating mode selection for external operator terminals	binary	M
D32	nnINT4	internal use		
::	::	internal use		
D48	nnSyn16	Synchronization steps 1 thru 16	binary	K
::	::	::		K
D62	nnSyn128	Synchronization steps 113 thru 128	binary	K
D64	nnSch16	Steps 1 thru 16	binary	K
::	::	::		K
D78	nnSch128	Steps 113 thru 128	binary	K
D80	nnBef16	Command output, steps 1 through 16	binary	K
::	::	::		K
D94	nnBef128	Command output, steps 113 thru 128	binary	K

Fig. 4-22 Cascade Data Module

D0 Data word assignment

Bit	Error and/or Status message	Troubleshooting
15	Structural fault in cascade module	The structure of the jump distributor fails to correspond to the sequence of programmed steps. Cascade runs correctly but diagnostics are not possible. → Correct cascade structure.
14 thru 4		Reserved
3	Jump sequence error	The first instruction to appear in the step module must be the jump instruction: SP [A] Only jump instruction or comment lines may be inserted between above instruction and the first jump instruction, e.g.: -S1 **CAUTION: Program segmentation!** → Correct program.
2	Reference list	The existing module is faulty. → Recompile and load program.
1	Step module (PM) not available	The program module PMn assigned to a defined data module DMn is not available. → Ensure linking of required module.
0	Number of steps too high or zero	Number of steps must be between > 0 and ≤ 128. → Correct program.

Fig. 4-23 Error Word in Cascade Data Module

Interpretation of wait time and monitoring time (D18, D20)

Upon jump entry into a new step, the KETTE module verifies whether or not the actual values for wait time and monitoring time (D18, D20) were set by the application program. In the case of values unequal zero, these will be interpreted as program lines that are valid for the active step. Otherwise, the time defaults will be taken from parameters P2 and P3 of the KETTE cascade management module.

4.6 Machine Usage

4.6.1 Definitions

Introduction

The Machine Usage function is used for recording production data and for creating production statistics. As a contribution to this statistical logging function, the production times for the three-shift operating with 6 work breaks each can be entered directly at the BT100 operator terminal.

The data recording utilizes seven standardized recording parameters, as well as logic links.

Recording parameters:

- Machine On (Daily logging)
- Production (Shift-specific and daily logging)
- Parts Count (Shift-specific logging)
- Fault (Daily logging)
- No Parts (Daily logging)
- Buffer Full (Daily logging)
- Standstill (Daily logging)

Actual piece-count logging occurs in terms of absolute numbers as well as in terms of a percentage value of the nominal (or *setpoint*) piece count. A maximum of 65535 pieces/parts can be counted in a single work-shift period.

All other parameter values are logged on a time base and calculated in percentages. The value of 100 % forming the basis is calculated as follows:

Effective work time = shift length minus total breaks.

This means that production values in excess of 100 % can also occur in the event that the machine continues producing during break times.

The recorded data will be stored in the PLC for a period of 14 days. The data for the current day (today) and the previous day (yesterday) is transferred to the BT100 operator terminal, where it will be statistically recorded and evaluated. The time period for which data may be stored on the BT100 can be freely defined by the user, and is limited only by the unit's hard drive capacity.

- The operator terminal provides the following functions:
- Bar graph display of recorded data
- Display of current production day
- Display of individual work shifts.
- Curve plotting of current production process or encompassing a definable historic time period
- Trend data exportability to standard software applications

4.6.2 Programming

The Machine Usage function module comprises 6 program modules and 2 data modules.

Of the supplied modules, you will only be required to program the call-up and parameterization of the MMIAUSL module. All other modules are merely entered in the module list.

Basic modules:

- MMILOGIK Logical links for recording parameters
- MMISTCK Parts count recording
- MMIPROZ Percentage recording
- MMISCHT Work shift data recording
- MMISTAT Recording statistical data (e.g. calendar date)

A special feature is the basic MMILOGIK module. In the event that the logics as supplied do not meet your needs, you can adapt MMILOGIK to your requirements (refer also to the section entitled, "MMILOGIK – Machine usage recording logics" further on in this chapter).

Program structure

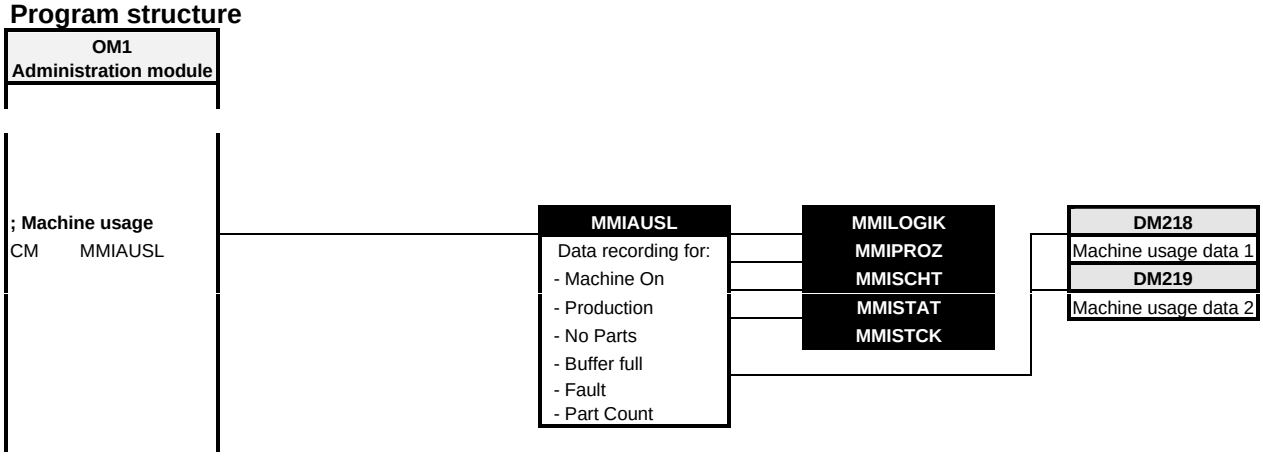


Fig. 4-24 Program Structure, Machine Usage Functions

Refer to Chapter 4, "Interfaces — PLC <–> Operator Terminals," for the following information:
Contents of DB218 data module "Auslast1" = Machine usage data 1
Contents of DB219 data module "Auslast2" = Machine usage data 2

Example

The supplied standard software contains, for the module call, a programming example in the form of linked program segments in the OM1. You can copy this example from the OM1 into your own program.

Module call

```
;Parameterization of "MMIAUSL" machine usage module
;=====================================================================
;
;Parameter P0: This parameter stores the recording bit statuses,
;              where individual bits have the following meaning:
;       - Bit 0: Machine On
;       - Bit 1: Production
;       - Bit 2: No Parts
;       - Bit 3: Buffer Full
;       - Bit 4: Fault
;       - Bit 5: unused
;       - Bit 6: Piece Count +1 (machine cycle counter)
;       - Bit 7: Delete all recorded data
;
;Parameter P1: Multiplier value if more than one piece is produced
;              per machine cycle (max. 255)
;
;The governing logics are programmed in the internally called MMILOGIK
;module, and can be adapted to your system
;while retaining the same filename.
;
;Parameters P2 through P6 denote basic modules,
;and should not be modified.
;

; Processing machine usage parameters for P= and P1
;------------------------------------------------------
DEF     Kxxx,-MULT        ; Count up xxx pieces per machine cycle
DEF     YYY,-AUSLDAT      ; Operand, recording parameters

; Prepare recording parameters
;    :
;    :
;    :
;
CM      -MMIAUSL,7
;                    +---+
P0   BY -AUSLDAT   ; <   ! Machine usage data reshaped by supplied logics.
P1   BY -MULT      ; <   ! Multiplier (n pieces/machine cycle)
P2   W  -MMILOGIK  ; <   ! Module for processing recorded data
P3   W  -MMISTCK   ; <   ! Module for recording piece count data
P4   W  -MMIPROZ   ; <   ! Module for recording percentage data
P5   W  -MMISCHT   ; <   ! Module for defining work shift data
P6   W  -MMISTAT   ; <   ! Module for recording statistical data (nominal count)
;                    +---+
;
```

Fig. 4-25 Module Call, MMIAUSL Program Module

Parameter description

In the DEF lines for the P0 and P1 module parameters preceding the module call, you enter the constants and/or operands that are designated for this function in your program.

P0	**BY**	**-AUSLDAT**	Here you enter the operand that contains the recording parameters. The recording parameters are provided by yourself, transformed according to logic supplied by MMILOGIK program module, and subsequently stored. **To facilitate recording of each counting pulse, Bit 6, handling the piece count function, must show LOW and/or HIGH status for at least 4 PLC cycles!** **Only a LOW byte is permitted as a parameter. The byte is read and transformed, and the result is loaded into the HIGH byte. This means that the HIGH byte may not be continued to be used for any other purpose!**
P1	**BY**	**-MULT**	Here you enter the factor (1-255) by which each piece count pulse (bit 6 of parameter P0) is to be multiplied. This will become effective whenever more than one piece per machine cycle is being produced.
P2	**W**	**-MMILOGIK**	Symbolic name of basic module. The entry must not be changed.
P3	**W**	**-MMISTCK**	Symbolic name of basic module. The entry must not be changed.
P4	**W**	**-MMIPROZ**	Symbolic name of basic module. The entry must not be changed.
P5	**W**	**-MMISCHT**	Symbolic name of basic module. The entry must not be changed.
P6	**W**	**-MMISTAT**	Symbolic name of basic module. The entry must not be changed.

Machine usage recording logic MMILOGIK

This module transforms the recording parameters applied to parameter P0 (LOW byte of an operand) of the MMIAUSL program module, and loads the result into the free HIGH byte of this operand (refer to description of MMIAUSL, parameter P0).

If another logic is to be processed, this module can be modified by yourself as required.

Module contents

```
+------------------------------------------------------------------------------+
|    1 |                                                                       |
+------------------------------------------------------------------------------+


     ; TRANSFORMATION OF MACHINE STATUSES INTO ENABLES FOR
     ; RECORDING MACHINE STATUSES IN ACCORDANCE WITH DEFAULT LOGIC
     ;===========00================================================

     ; This module is called from within the MMIAUSL module!

     ; READ MACHINE STATUSES AS DIGITAL INFORMATION

     ; Default bits
     DEF     M230.0,-Masch_e
     DEF     M230.1,-Produkt
     DEF     M230.2,-k_Teile
     DEF     M230.3,-Puffer_v
     DEF     M230.4,-Stoerung
     DEF     M230.5,-MStoe
     DEF     M230.6,-MZykl+1
     DEF     M230.7,-loesch
     ; Recording bits
     DEF     M231.0,-MASCH_E
     DEF     M231.1,-PRODUKT
     DEF     M231.2,-K_TEILE
     DEF     M231.3,-PUFFER_V
     DEF     M231.4,-STOERUNG
     DEF     M231.5,-STILLST:
     DEF     M231.6,-STUECK
     DEF     M231.7,-LOESCH


2    L    W  P0,A      ;INPUT INFORMATION
3    T    W  A,M230    ;ON SCRATCH MARKER



+------------------------------------------------------------------------------+
|    2 | Machine On                                                            |
+------------------------------------------------------------------------------+

             +-----+                                    +-----+               -MASCH_E  M231.0
   -Masch_e -+  &  +------------------------------------+-+  =  +--MASCH_E      -Masch_e  M230.0
             +-----+                                    +-----+


+------------------------------------------------------------------------------+
|    3 | Production                                                            |
+------------------------------------------------------------------------------+

             +-----+                                                           -Masch_e  M230.0
   -Produkt -+  &  |                                                           -PRODUKT  M231.1
   -Masch_e -+     |                                                           -Produkt  M230.1
   -k_Teile -O     |                                                           -Puffer_v M230.3
   -Puffer_v-O     |                                    +-----+                -STOERUNG M231.4
   -STOERUNG-O     +------------------------------------+-+  =  +--PRODUKT      -k_Teile  M230.2
             +-----+                                    +-----+
```

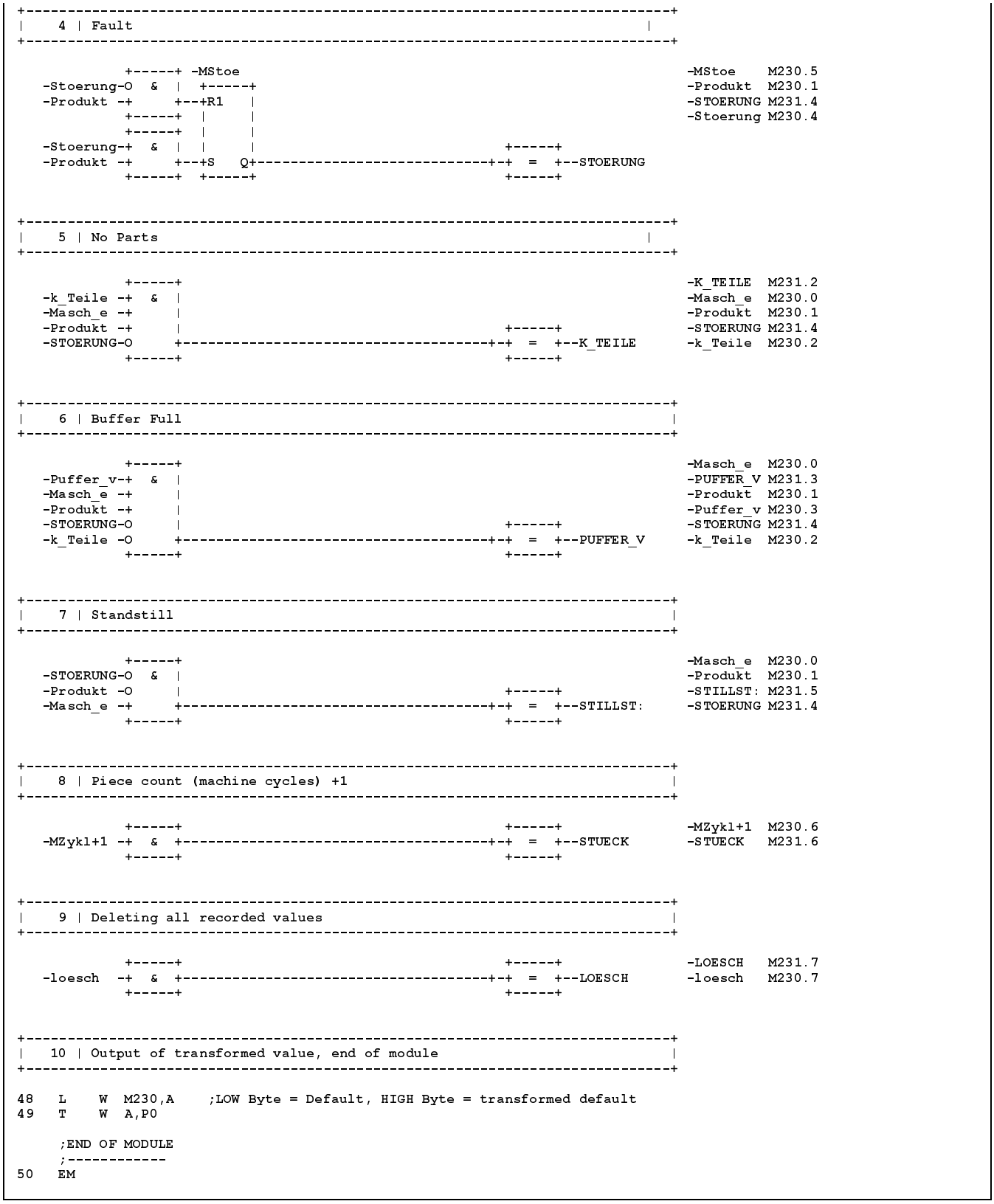


Fig. 4-26 MMILOGIK Module Contents

4.7 Machine Cycle Times

4.7.1 Definitions

The MMI-MADAP software provides you with the power to manage 48 machine cycle times. The timekeeping resolution can be set to either 0.1 or 1.0 sec. The respective measuring times are formed in basic program modules.

Measuring and storage of cycle times is handled by data module DM217.

The timers T117 (0.1 sec cycle) and T118 (1.0 sec cycle) are used for time determination. You may therefore not used these times.

To select the measuring accuracy (resolution), the DM217 data module defines one data bit for each cycle time.

The incrementation of times is effected by releasing and disabling of defined start bits. The readings thus determined are transferred to the associated display values via the stop bits. They are subsequently deleted in order to enable the transfer of new measurements.

You will be required to manage and control the program-specific treatment of the start and stop bits.

4.7.2 Programming

Program structure

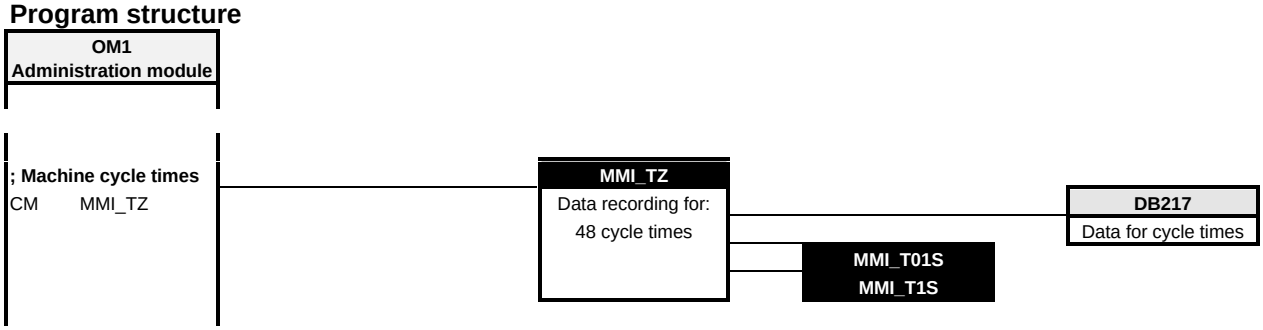


Fig. 4-27 Program Structure, Machine Cycle Times

Refer to Chapter 4, "Interfaces — PLC <–> Operator Terminals," for the following information: DM217 "Cycle Times" data module contents: Machine cycle times

Example

The supplied standard software contains the following programming example in the form of linked program segments in the OM1. You can copy this example from the OM1 into your own program, and modify it to suit your requirements.

Program example

This module calls submodules that merely require entry in the module list, and that do not expect any further activities.

Submodules: MMI_T01S and MMI_T1S.

```
;Module for cycle time management
;================================
;Can be used to generate up to 48 cycle times.
;
;In the event that cycle times with a time base of 1 sec and 0.1 sec are
;to be processed, the module must be called 2 times, with parameter P0
;defining the recording cycle. Contingent upon the definition of P0, only
;those times for which the time base bit was appropriately set are processed.
;
;The recording of a cycle time (incr. of measuring time) is controlled (started
;and/or stopped) by the corresponding start bit.
;
;The display of a cycle time is updated by setting the appropriate stop bit
;with simultaneous deletion of the measuring time.
;

; Controlling time recording for 2 cycle times (CT)
;-------------------------------------------------
CM      -Taktzeit
; Number of cycle times to be processed
DEF     D100,-TZ-Anz

L    W  K2,A              ; Process 2 cycle times
T    W  A,-TZ-Anz

; Time base bits
DEF     D114,-Bas01-16    ; Time base bits, cycle times 1 thru 16

L    W  K0002H,A          ; Bit 0 = 0 --> CT 1 reads exactly 0.1 sec
T    W  A,-Bas01-16       ; Bit 1 = 1 --> CT 2 reads exactly 1.0 sec

; Start bits
DEF     D102,-Sta01-16    ; Start measuring times, CT 1 thru 16
DEF     E14.0,-Sta01      ; Start bit, CT 1
DEF     E14.1,-Sta02      ; Start bit, CT 2

L    W  -Sta01-16,A
A    B  -Sta01
=OM  B  A.0
A    B  -Sta02
=OM  B  A.1
T    W  A,-Sta01-16

; Stop bits
DEF     D108,-Stp01-16    ; Write display value, delete measuring time 1 - 16
DEF     E15.0,-Stp01      ; Stop bit CT1
DEF     E15.1,-Stp02      ; Stop bit CT2
DEF     M15.0,-Fl01       ; Flank marker CT1
DEF     M15.1,-Fl02       ; Flank marker CT2

; ... delete cyclically
L    W  K0,A
T    W  A,-Stp01-16
```

```
;  Update CT1 for display
AN   B  -Stp01
R    B  -F101
AN   B  -Stp01
O    B  -F101
JPC     -nostp1
AN   B  -F101
S    B  -F101
L    W  -Stp01-16,A
SWM  B  A.0
T    W  A,-Stp01-16
     -nostp1

;  Update CT2 for display
AN   B  -Stp02
R    B  -F102
AN   B  -Stp02
O    B  -F102
JPC     -nostp2
AN   B  -F102
S    B  -F102
L    W  -Stp01-16,A
SWM  B  A.1
T    W  A,-Stp01-16
     -nostp2

DEF     SM31.1,-log1
DEF     SM30.3,-log0
DEF     E0.0,-loesch
DEF     M4.0,-Takt01s
DEF     M4.1,-Takt1s

;  Call-up for cycle time measurements with 1.0 sec time base
;-------------------------------------------------------------
BA      -MMI_TZ,5
;                  +---+
P0   B  -log1     ; <   ! Cycle definition: 0 = 0.1s, 1 = 1s
P1   B  -loesch   ; <   ! Delete all display and measured values
P2   B  -Takt1s   ; !   > Cycle output as pulse, as defined in P0
P3   W  -MMI_T01S ; <   ! Module name for cycle generation 0.1s (uses T118)
P4   W  -MMI_T1S  ; <   ! Module name for cycle generation 1.0s (uses T117)
;                  +---+
;
;  Call-up for cycle time measurements with 0.1 sec time base
;-------------------------------------------------------------
BA      -MMI_TZ,5
;                  +---+
P0   B  -log0     ; >   ! Cycle definition: 0 = 0.1s, 1 = 1s
P1   B  -loesch   ; >   ! Delete all display and measured values
P2   B  -Takt01s  ; !   > Cycle output as pulse, as defined in P0
P3   W  -MMI_T01S ; <   ! Module name for cycle generation 0.1s (uses T118)
P4   W  -MMI_T1S  ; <   ! Module name for cycle generation 1.0s (uses T117)
;                  +---+
```

Fig. 4-28 Program Example, Machine Cycle Times

4.8 Decoding BT100 Movement and Function Keys

4.8.1 Definitions

As the BT100 operator responds to the actuation of a movement or function key by sending an 8-bit code to the PLC, this code must be converted (8 bits in 1 of 16) in order to ensure fault-free operation of the PLC program.

4.8.2 Programming

Program structure

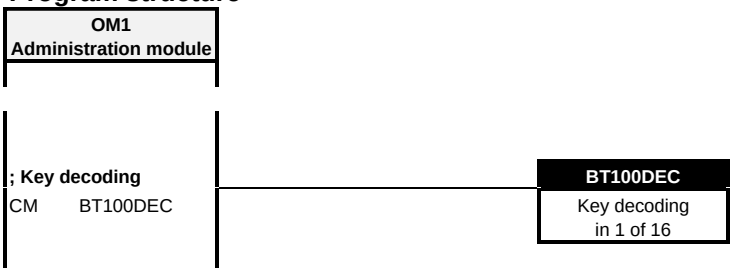


Fig. 4-29 Program Structure, Key Decoding

Example

The supplied standard software contains a programming example in the form of linked program segments in the OM1. You can copy this example from the OM1 into your own program.

Module call

```
;Decoding the codes for BT100 movement and function keys
;-------------------------------------------------------
;
DEF     XXX,-TastCode     ; Key code from terminal panel
DEF     YYY,-BewTast      ; decoded movement keys
DEF     D300,-FktTast     ; decoded function keys

DEF     DB230,-BF1_DB     ; Communication and display data for BT1
CM      -BF1_DB           ; Communication and display data for BT1

CM      -BT100DEC,3       ; BT100 key decoding
;                   +---+
P0   BY -TastCode   <   !  Key code from BT100
P1   W  -FktTast    !   >  actuated function key 1 of 16
P2   W  -BewTast    !   >  actuated movement key 1 of 16
;                   +---+
```

Fig. 4-30 Module Call for BT100DEC Program Module

Parameter description

In the DEF lines for the module parameters preceding the module call, you can enter the operands that are defined for this function in your program.

In the event that you want to operate several operator terminals on a single central processing unit, you will be required to call this module once for each operator terminal (BF). In this case the identifier for the data module assigned to the respective operator terminal must be entered.

```
DEF     DB230,-BF1_DB       ; Communication/display data
BA      -BF1_DB             ; ... for BF1

DEF     DB235,-BF2_DB       ; Communication/display data
BA      -BF2_DB             ; ... for BF2

DEF     DB230,-BF3_DB       ; Communication/display data
BA      -BF3_DB             ; ... for BF3

DEF     DB230,-BF4_DB       ; Communication/display data
BA      -BF4_DB             ; ... for BF4
```

P0 BY -TastCode

The parameter P0 will be assigned the operand to which the operator terminal will send the encoded 8-bit signal representing the actuation of function and/or movement key.

P1 W -FktTast

Parameter P1 outputs the decoded function keys (1 of 16). P1 is copied into data word D300 of the data module assigned to the selected operator terminal. This causes any movement to be blocked while a movement screen is being changed. In the context of MMI-MADAP standard software functions, the decoded function keys are not used for any other purpose.

NOTE: If you are using another type of decoding module, you must ensure that each actuation of a function key is signalled to data word D300 of the data module that is assigned to the selected operator terminal.

Definition for decoding (FK = function key):
FK1 = Bit 0 ... FK16 = Bit 15

P2 W -BewTast

Parameter P2 contains the decoded movement keys (1 of 16). To effect operator terminal linking, apply the movement keys to input parameter P3 of the MMI-MADAP program module described in the following section.

Definition for decoding (MovK = movement key):
MovK1 left = Bit 0 ... MovK8 left = Bit 7
MovK1 right = Bit 8 ... MovK8 right = Bit 15

4.9 PROFIBUS-DP Diagnostics Principle

4.9.1 Definitions

The PROFIBUS-DP provides a variety of diagnostic services among which the classified slave diagnostics represent the most important function. With the use of MMI-MADAP, the service is interpreted, displayed and stored in the protocol record. For this purpose, 3 data modules were defined; two of these handle data acquisition, and one provides display data. The program modules processing the PROFIBUS-DP diagnostics are designated DPSTATUS and FIFODM1. As these are library modules, you do not have to parameterize them. The calling module for the DPSTATUS module is MMIDESI. It processes and transfers the DPSTATUS parameters.

4.9.2 Programming

Program structure

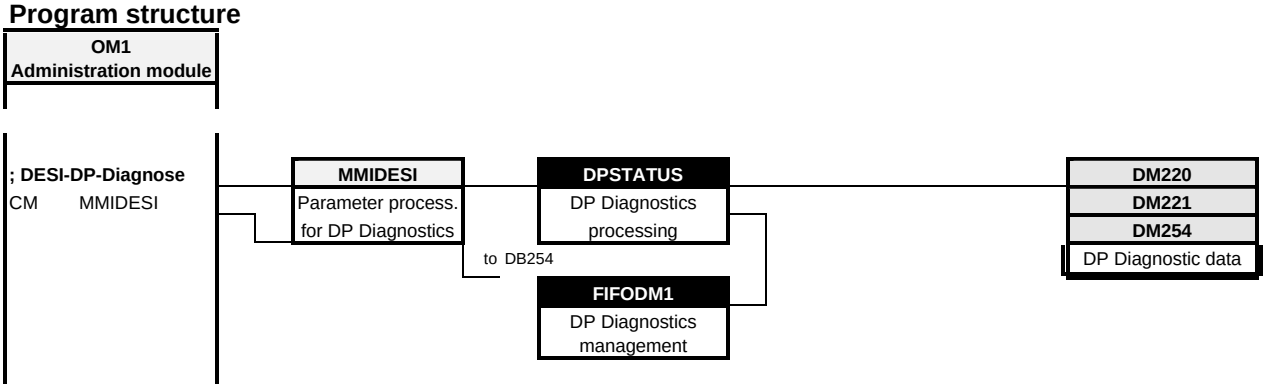


Fig. 4-31 Program Structure, PROFIBUS-DP Diagnostics

Refer to Chapter 4, ""Interfaces — PLC <–> Operator Terminals," for the following information:
DM220 "DP_Daten" data module contents: DESI-DP data
DM221 "DP_Komm" data module contents: DESI-DP communication channel 1
DM254 "AL_DP_D" data module contents: Usage display and DP-Diagnostics,

Example

The supplied standard software contains a programming example in the form of linked program segments in the OM1. You can copy this example from the OM1 into your own program.

Module call

```
;Module call, communication module for BM-DP12 PROFIBUS-DP bus master
;-------------------------------------------------------------------
;This module handles parameter processing for DPSTATUS program module and
;copies resulting diagnostic data into communication object 25 in DM254 for
;purposes of display and storage on the MMI-MADAP operator terminal.
;
;Definitions:
;
;DB220              Communication parameterization
;DB221              Communication data
;DB254 from D220    MMI display data
;
DEF     K2,-KF-Adr
;
CM      -MMIDESI,1
;                   +---+
P0   W  -KF-Adr   ; <   ! EI/EO switching matrix address
;                   +---+
```

Fig. 4-32 Module Call, MMIDESI Program Module

Parameter description

In the DEF lines for module parameter P0 preceding the module call, you enter the constant that corresponds to the switching matrix address at which you intend to operate the bus master.

P0	**W**	**- KF-Adr**	Switching matrix address at which the BM-DP12 is operated.

Module contents

In contrast to the other MMI-MADAP standard modules, the MMIDESI module is provided on the standard diskette in the form of an open module. Although you will link it to the program, you are not permitted to modify the module contents.

Major contents of DM254 data module "AL_DP_D": Usage display + DP diagnostics, as per EN 50170, part 2 (DP)

D220 Global status

Bit 0 : Bus master error
Bit 1 : Classified slave diagnostics (MMI-MADAP standard)
Bit 2 : System diagnostics
Bit 3 : reserved
Bit 3 : reserved
Bit 5 : DP-Bus STOP by DP bus master
Bit 6 : DP-Bus STOP by programming device
Bit 7 : DP-Busmaster 1=active, 0=not ready
Bit 8 : Slave(s) not reachable via DP bus
Bit 9 : Slave(s) report configuration fault
Bit 10 : Slave(s) report static diagnostics
Bit 11 : Slave(s) report extended diagnostics
Bit 12 : Slave(s) not ready for cyclical data exchange
Bit 13 : Slave(s) report slave error
Bit 14 : reserved
Bit 15 : reserved

Slave error messages

D238.0-D253.7 Slaves not reachable
D254.0-D269.7 Slaves report configuration fault
D270.0-D285.7 Slaves report static diagnostics
D286.0-D301.7 Slaves report extended diagnostics
D302.0-D317.7 Slaves not ready for cyclical data exchange
D318.0-D333.7 Slaves report slave error

These message blocks contain 128 bits each, with the respective LSB being assigned to the programming device (address 0), and the next higher bit assigned to the bus master (address 1).

Examples:

D238.1 = 1, Bus master not reachable

D256.0 = 1, Bus station 17 reports configuration fault

D317.3 = 1, Bus station 124 not ready for data exchange

4.10 Operator Terminal Connectivity

The purpose of the operator terminal connectivity module is to handle the communications between the PLC and the BT100 operator terminal. The communication data encompass the functions required for operation, sequential control diagnostics and visualization on the MMI-MADAP operator terminal.

4.10.1 Definitions

The operator terminal connectivity module contains the **MMIMADAP** program module, plus the secondary **DIAGMMI** program module which does not require user parameterization, as well all data modules for the data interface.

Without exception, the **MMIMADAP** program module is called for each MMI-MADAP operator terminal. Up to four MMI-MADAP operator terminals can be connected to each central processing unit.

A **cascade range-specific diagnostic routine** occurs in conjunction with the DIAGMMI program module, thus facilitating the management of four independent stations per central processing unit.

Display data for operator terminal

The following data groups are processed for the MMI-MADAP operator terminal:

- Power-up conditions
- Manual movements with execution messages and execution enable
- Status displays for all operands,
  for PLC configuration,
  for I/O assignment,
  for DM list,
  for PLC module version ID, as well as
  system date and time.
- Machine usage with machine cycle times
- Message systems for sequential processing faults,
  PROFIBUS-DP errors,
  PLC messages plus user messages.

Data interface

The data interface between the PLC and the BT100 operator terminal is handled by designated data modules.

For a detailed discussion of these data interfaces, refer to Chapter 4, "Data Interfaces – PLC <-> Operator Terminals."

4.10.2 Programming

Program structure

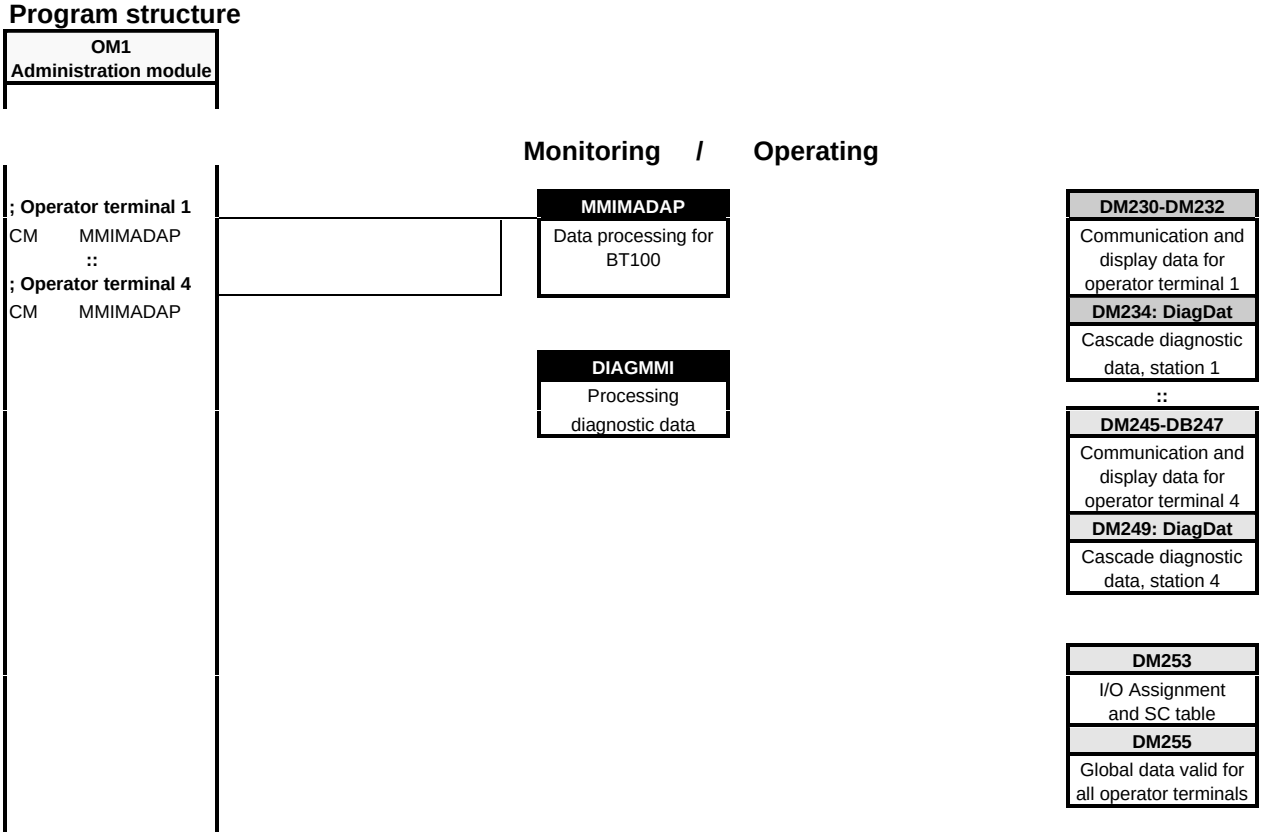


Fig. 4-33 Program Structure, Operator Terminal Connectivity

Example

The supplied standard software contains, for the module call, a programming example in the form of linked program segments in the OM1 administration module. You can copy this example from the OM1 into your own program.

Module call

Scratch markers must not be used in place of parameters!

The module call is made from within the OM1 administration module. The module must be called for each MMI MADAP operator terminal.

The module call is structured as follows:

```
; Module call, function module for operation of operator terminal
; In the event that several operator terminals are to be connected (max. 4),
; this module must be called for each connected terminal.
;-------------------------------------------------------------------------
;
DEF     K1,-BF-Nr
DEF     K4,-Kett_Anz
DEF     K1,-K_Start
DEF     E12.1,-BTsperr

CM      -MMIMADAP,7
;                   +---+
P0   W  -BF-Nr    ; <   ! Operator terminal no., starting with T1, no gaps
P1   W  -Kett_Anz ; <   ! Number of last cascade
P2   W  -K_Start  ; <   ! Number of first cascade
P3   W  -BewTast  ; <   ! Movement keys: HIGH byte=Right, LOW byte=Left
                  ; <   ! (T1=Bit0, T8=Bit7)
P4   W  -DIAGMMI  ; <   ! DIAGMMI program module
P5   W  -KETTE    ; <   ! Cascade module (for version ID only)
P6   B  -BTsperr  ; <   ! Remote locking of movement keys
;                   +---+
```

Fig. 4-34 Module Call, MMIMADAP Program Module

Parameter description

In the DEF lines for the module parameters preceding the module call, you can enter the operands that are defined for this function in your program.

In the event that you want to operate several operator terminals on a single central processing unit, you will be required to call this module once for each operator terminal (BF).

P0	**W**	**- BF-Nr1-4**	Parameter P0 provides the module with the number of the current MMI-MADAP operator terminal. (Value range: integers 1 - 4, starting with 1, no gaps). The parameter may be assigned a default constant.
P1	**W**	-**Kett_Anz**	Parameter P1 provides the module with the last cascade number that is valid for the MMI-MADAP operator terminal in parameter P0. (Value range: integers 1 - 64) The parameter P1/P2 will be required for the cascade-specific diagnostics function. The parameter may be assigned a default constant.
P2	**W**	-**K_Start**	Parameter P2 provides the module with the first cascade number that is valid for the MMI-MADAP operator terminal in parameter P0. (Value range: integers 1 - 64) The parameter may be assigned a default constant.
P3	**W**	-**BewTast**	Parameter P3 provides the module with the information regarding the 16 movement keys in the MMI-MADAP operator terminal. P3.0 Movement key 1 top left . P3.7 Movement key 8 bottom left P3.8 Movement key 1 top right . P3.15 Movement key 8 bottom right The manual movement keys of the MMI-MADAP operator terminal must be interpreted by the user, and must be made available to the MMI-MADAP module via parameter P3.

P4	**W**	**-DIAGMMI**	Parameter P4 provides the module with the symbolic module name of the program module handling the diagnostic function. (Value range: symbolic name, 8 characters) Default name: -DIAGMMI. The symbolic name must be entered in the symbol file. The parameter is transferred as a symbol name.
P5	**W**	**-KETTE**	Parameter P5 provides the module with the symbolic module name of the program module handling the sequential control. (Value range: symbolic name, 8 characters) Default name: -KETTE. The symbolic name must be entered in the symbol file. The parameter is transferred as a constant.
P6	**B**	**-BTsperr**	Parameter P6 provides the module with a locking instruction for all movement keys. P6.0 = LOW , movement keys enabled, parameter P3 P6.0 = HIGH , movement keys disabled, parameter P3 With the aid of this bit parameter, the user can cause, independent of parameter P3, a centralized movement lock for all manual movement functions. The locked/disabled status is displayed on the MMI-MADAP operator terminal as Code 99.

Example

Module call for MMIMADAP program module handling operator terminal 1. Cascades 1 through 12 are used for sequential control and diagnostics function.

```
; Module call, function module for operation of operator terminal
; In the event that several operator terminals are to be connected (max. 4),
; this module must be called for each connected terminal.
;-------------------------------------------------------------------------

;
DEF     M50,-BewTast       ; User marker, movement keys
DEF     K1,-BF-Nr          ; Operator terminal no. =1
DEF     K12,-Kett_Anz      ; last cascade = 12
DEF     K1,-K_Start        ; first cascade = 1
DEF     M52.0,-BTsperr     ; User marker, disable movements

CM      -MMIMADAP,7
;                    +---+
P0   W  -BF-Nr     ; <   ! Operator terminal no., starting with T1, no gaps
P1   W  -Kett_Anz  ; <   ! Number of last cascade
P2   W  -K_Start   ; <   ! Number of first cascade
P3   W  -BewTast   ; <   ! Movement keys: HIGH byte=Right, LOW byte=Left
                   ; <   ! (T1=Bit0, T8=Bit7)
P4   W  -DIAGMMI   ; <   ! DIAGMMI program module
P5   W  -KETTE     ; <   ! Cascade module (for version ID only)
P6   B  -BTsperr   ; <   ! Remote locking of movement keys

;                    +---+
```

Fig. 4-35 Module Call, MMIMADAP Program Module

Current data modules:

- DM1 through DM12 Cascade data modules
- DM230 Operator terminal data, operator terminal 1
- DM231 Status display data, operator terminal 1
- DM232 Movement locks, operator terminal 1
- DM233 User communications, operator terminal 1
- DM234 Diagnostic messages, operator terminal 1
- DM253 I/O assignment abd SC table
- DM255 Global operator terminal data

4.11 Processing Fault Diagnostics

4.11.1 Definitions

Introduction

The DIAGMMI program module is managed in the MMIMADAP program module, and requires entry in the symbol files by the user.

Criteria analysis

The DIAGMMI program module is used to monitor a machine that is programmed in accordance with cascade sequencing technology. In the case of a fault, the monitored steps will be checked for non-satisfied step-on conditions. In case of a fault, the first branch for which conditions are not satisfied will be transferred to the diagnostic module as a diagnostic result. The search for the non-satisfied branch begins in the BEFA command output branch.

The display of diagnostic results differentiates between Manual and Automatic mode. This diagnostic approach requires the step module structure as described in preceding sections.

Diagnosable step-on conditions

A given diagnostic step can contain a maximum of 64 step-on conditions for a BEFA command output or WSB step-on branch.

If more than 64 step-on conditions are required, BEFA and/or WSB branches can be generated as often as required.

Programming example:

```
A       B       -Eingang1
O       B       -Eingang2
.
.
=       B       -BEFA


A       B       -BEFA
A       B       -Eingang3
.
.
=       B       -BEFA
```

In this case the diagnostics always investigate the first non-satisfied BEFA and/or WSB branch.

The following instructions are permitted for the formation of step-on conditions:

Bit instructions

- A
- O
- AN
- ON
- S
- R
- =

Special instructions

- (
-)
-)N
- Program segmentation commands (control codes)

Permitted operands for link instructions:

I	Inputs	I0.0	–	E63.7
O	Outputs	O0.0	–	A63.7
M	Markers	M0.0	–	M255.7
T	Times	T0	–	T127
C	Counters	C0	–	Z127
SM	Special markers	SM0.0	–	SM31.7

Definition of terms

- **Fist-value error**

A system runs without problems when all cascades contributing to the program sequence (max. 64) are processed without error in Automatic mode. If an error occurs within a cascade, this will be recognized by the controller. As this error happens to be the one processing error to have occurred first in the system, it will be designated the *first-value error*.

- **Sequence error**

If a first-value error is present in the system, it may be assumed that additional cascades enter into a fault state. These failures are designated *sequence errors*. As a rule, sequence errors are of low significance because in most cases the system can be returned to normal operation by remedying the first-value error.

Error entry criteria

There are two ways in which an error occurrence (error entry) can be triggered in a given cascade:

- **Monitoring time**

Each step of the cascade performs a specific function. This function may a translate into a physical system movement, and it may also comprise a preparation of additional movements. Each function requires specific execution time which can be measured. The monitoring time function is used to control this time interval, and triggers an error entry in the event of a time overrun.

- **Fault marker**

In the case of time-critical faults, e.g. the opening of protective doors, it may not be possible to employ the monitoring time function in a useful manner. In this case, the occurrence of a fault will cause an immediate reset of the fault marker, triggering an instant error entry.

4.11.2 Programming

As this program module is called from within the MMIMADAP program module, it does not require specific attention on the part of the user.

The module call for the program module is structured as follows:

```
CM      - DIAGMMI,8
                  ; +---+
P0   B  -A/H-Diag ; <   !  0 : Auto-Diagnosis; 1 : Manual diagnosis
P1   B  -H/A-Zwg  ; <   !  Diagnose 0 : Manual branch, 1 : Auto branch
P2   BY -KettNr   ; <   !  Cascade no. for Manual diagnosis (DMx, PMx)
P3   BY -KettAnz  ; <   !  Last cascade for Auto-diagnosis
P4   BY -BasKette ; <   !  First cascade of station
P5   BY -DiagMldg ; <   !  Return of module messages
P6   W  -DiagDB   ; <   !  Number of DM in which diagnostic data is filed
P7   BY -Station  ; <   !  Station ID 1-8, for display on MMI
                  ; +---+
```

Fig. 4-36 Module Call, DIAGMMI Program Module

Description

Any station, the cascade group of which is defined by the parameter range P3 to P4, can generate its own first-value error and cause it to be entered in the assigned data module.

In the cascade information data, bit 11 is used to identify the Cascade Reports First-value Error message.

In manual diagnostics, each cascade can be cursor-selected and subjected to analysis.

The only permitted station numbers are numbers 1 through 8.

The foregoing enables MMI-MADAP to label its own cascade group with a station number. The MMIMADAP module writes this number (local ID) into data byte D13 of the data module which is addressed by parameter P7. In addition, a data range encompassing 8 words is required, into which the cascade range for each station is entered, stating first and last cascade.

D-Addr	last cascade	first cascade	Comment
D420 : D434			Status, 1 : Status, 8

The diagnostics module must be called for each cascade.

4.11.3 Storing Diagnostic Information

In the data modules listed below, and optionally in the data modules DM222 through DM225 (substations 5 through 8), information about the current (active) first-value error is automatically stored in a specially designated data range (D0 - D148).

- DM234 MMI-MADAP operator terminal 1 (base station 1)
- DM239 MMI-MADAP operator terminal 2 (base station 2)
- DM244 MMI-MADAP operator terminal 3 (base station 3)
- DM249 MMI-MADAP operator terminal 4 (base station 4)

The data pertaining to sequence errors occurring in the PLC controller during cascade processing can be queried via the manual diagnostic range (D278 through D438).

	DW	Contents, HIGH byte	Contents, LOW byte
	D0	Control flags	
Range of Automatic Diagnostics (First-value message)	D2	Day	Month
	D4	Year	Hour
	D6	Minute	Second
	D8	Weekday (0 = Sunday)	unused
	D10	Cascade number	Step number
	D12	Station ID	Module number
	D14	Cascade status	Number of messages
	D16	1. Opcode	
	D18	2. Opcode	
	:	:	
	D142	64. Opcode	
	D144	Reserved	
	D146	Reserved	
	D148	Reserved	
	D150	Cascade 1 information	
	:	:	
	D276	Cascade 64 information	
Range of Manual Diagnostics	D278	Cascade number	Step number
	D280	Stationskennung	Module number
	D282	Cascade status	Number of messages
	D284	Opcode 1	
	:	:	
	D410	Opcode 64	
Station list	D420	Last cascade, station 1	First cascade, station 1
	:	:	:
	D434	Last cascade, station 8	First cascade, station 8
	D436	Reserved	
	D438	Reserved	

Fig. 4-37 Data Module, Diagnostic Information

☞ **The data modules up to and including D438 (440 bytes) must be opened.**

Automatic diagnostic range

- **Control flags**

Data word D0 contains the control flags that are managed by MMI-MADAP, with individual data bits serving different functions.

D0.0

In the event that a first-value error was entered, diagnostic module DIAGMMI sets data bit D0.0 is set HIGH. As soon as the operator terminal signals "Error Stored", the MMIMADAP again sets this bit LOW. This data bit is functionally interdependent with data bit D0.1.

D0.1

Data bit D0.1 controls the response to an active first-value error.

- **D0.1 = LOW**

The first-value error is always entered. If an unacknowledged first-value error is already present in the data module, this will be overwritten by the new first-value error.

- **D0.1 = HIGH ; MMI-MADAP default setting**

A new first-value error can be entered only if the acknowledgement of a preceding first-value error was effected by means of data bit D0.0. If this is not the case, the new first-value error will be discarded.

D1.0

If a first-value error is active, data bit D1.0 is set HIGH by the DIAGMMI diagnostic module. If this is not the case, the bit will be reset automatically.

Diagnostic data

In the case of an error the DIAGMMI module automatically enters diagnostic data in the following data words. These are then read by MMI-MADAP and, subsequent to processing for the display, entered in the diagnostics module in the operator terminal.

Date format

The time and date of an active first-value error are written to data words **D2** through **D8** in hexadecimal format. Weekdays are coded as follows:

- 0_{hex} = Sunday
- 1_{hex} = Monday
- 2_{hex} = Tuesday
- 3_{hex} = Wednesday
- 4_{hex} = Thursday
- 5_{hex} = Friday
- 6_{hex} = Saturday

Step number

This data byte (D10) stores the step number of the faulty cascade.

Cascade number

This data byte (D11) stores the cascade number of the faulty cascade.

Module number

Each cascade is programmed in an associated program module. The number of the program module is stored in data byte D12. It corresponds to both the cascade and to the cascade data module number.

Station ID

Data byte D13 contains the station ID of the faulty cascade.

Number of messages

Data byte D14 contrains the number of conditions attached to the active first-value error. The representation uses hexadecimal format. Only the first 64 conditions are stored in the data module. In the event that the BEFA command output or WSB step-on condition branch consists of more than 64 conditions, the number of messages will be set to 65.

If the value FF_H is returned with this word, this indicates that an illegal instruction was detected in the instruction block that was subject to the diagnostic check.

Cascade status

Data byte D15 indicates the active operating mode of the faulty cascade at the time the first-value error occurred.

- Bit 0 (value 1_{hex}) = cascade in Manual mode
- Bit 1 (value 2_{hex}) = cascade in Inching mode
- Bit 2 (value 4_{hex}) = cascade in Automatikbetrieb

Opcode

Starting with data word D16, the opcode of the criteria of the faulty branch is stored. Each line of instructions is represented by a data word. The significance of a data word is pointed out below.

Command code / link status

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Z	Opcode			Operand + Byte addr.									Bit 0-7			Instruction
	z	0	0	0													A
	z	0	0	1													AN
	z	0	1	0													O
	z	0	1	1													ON
	z	1	0	0													S
	z	1	0	1													R
	z	1	1	0													=
		1	1	1										0	0	0	(
		1	1	1										0	0	1	O(
	z	1	1	1										0	1	0	)
	z	1	1	1										0	1	1	)N
		1	1	1										1	X	X	Reserved
	Operand or link status/condition: 1: satisfied; 0: not satisfied																

Fig. 4-38 Opcode Definiton for Operator

Operand ID and byte address ranges

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex.	
	Z	Instruct.			Operand + ***Byte addr.***									Bit 0-7			Mask	Operand
from					0	0	0	0	0	*0*	*0*	*0*	*0*				0000	C: Counter status
to										*1*	*1*	*1*	*1*				0078	128 bytes
from					0	0	0	0	1	*0*	*0*	*0*	*0*				0080	T: Timer status
to										*1*	*1*	*1*	*1*				00F8	128 bytes
from					0	0	0	1	*0*	*0*	*0*	*0*	*0*				0100	SM:Special marker
to									*1*	*1*	*1*	*1*	*1*				01F8	Bytes 0-31
from					0	0	1	0	0	0	0	0	0				0200	I: Inputs
to					0	0	1	1	1	1	1	1	1				03F8	Bytes 64-127
from					0	1	0	*0*	*0*	*0*	*0*	*0*	*0*				0400	Bytes 0-63
to								*1*	*1*	*1*	*1*	*1*	*1*				05F8	
from					0	1	1	*0*	*0*	*0*	*0*	*0*	*0*				0600	O: Outputs
to								*1*	*1*	*1*	*1*	*1*	*1*				07F8	Bytes 0-63
from					1	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*				0800	M: Markers
to						*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*				0FF8	Bytes 0-255
					Convention applies also to C and T Example: C 127 is represented as C 15.7												Subsequent to masking of bits 0-2 and 12-15	

Fig. 4-39 Opcode Definition, Operand ID and Byte Address

Cascade information structure

The first-value error range in the diagnostic module is followed, beginning at D150, by a block containing information about the available cascades. The table below shows the cascade information structure.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Z					Opmode			Step no.							

0 0 1 = Manual mode
0 1 0 = Inching mode
1 0 0 = Automatic mode

0 = No first-value errors
1 = Casc. returns first-value error

0 X 1 = Halt
0 1 X = Start
1 0 0 = Reset

Cascade status:
0 = fault-free
1 = faulty

Fig. 4-40 Cascade Information in Diagnostic Data Module

Manual diagnostic range

With regard to functional contents, the manual diagnostic range corresponds to that of the automatic diagnostics. However, the date and time of a given entry are not recorded. It starts with the step number in D278. The opcode range starts with data word D284.

In the case of the manual diagnostics, a control flag is omitted. If diagnostic messages are active, the value in D282 is higher than zero by the message count.

Diagnostic changes can be interpreted only by means of the data word D278 (cascade, step number) and word D282 (number of messages).

If the value FF_H is returned with data word D282, this indicates that an illegal instruction was detected in the instruction block that was subject to the diagnostic check.

4.11.4 Processing faults

Messages from the DIAGMMI module are entered in data word D22 of the operator terminal data modules.

The referred messages comprise error messages that cannot be entered in data word D0 of the cascade data module. The status word has the following significance:

Bit	Cause of error	Error remedy
15	Group fault indication	At least one of the following errors, except bit 0, is active.
14	not relevant	
13	The default first cascade in P4 was defined with a higher number than the last cascade.	→ Correct parameter settings in DIAGMMI module.
12	Data module for cascade not found OR too short.	The DIAGMMI is attempting to diagnose a cascade but either cannot find the associated data module, or detects that it is incomplete. → Link the data modules of correct length with your program.
11	No free storage capacity in data field	BOSCH standard modules occupy data field blocks. A total of 12 data field blocks are available to the controller. → Reduce the total number of calls for BOSCH modules.
10	Step number is too high.	The step number to be diagnosed exceeds 128. A step number in excess of 128 may have been accidentally selected in Manual mode.
9	Cascade number not permitted with Manual diagnostics.	Parameter P2 of DIAGMMI module is not within permitted range (between 1 and 64). → Change parameter defaults.
8	Number of cascades not permitted.	Parameter P3 of DIAGMMI module has a value in excess of 64. You can only process a maximum of 64 cascades, however. → Change parameter defaults.

Bit	Cause of error	Error remedy
7	Error in system range (System command)	This bit indicates that too many system commands (e.g. LAD and TAD) are being used in the overall PLC system. → Reduce the number of system commands OR arrange the commands in a coordinated sequence. Refer also to the manual **CL500 System commands**, 1070 072 068.
6	Structural fault in cascade jump distributor	The DIAGMMI module has detected a structural fault in the cascade jump distributor. This may be caused by a mismatch between the step sequence in the jump distributor and that in the step program. → In the step module, change the step sequence either in the jump distributor or in the process sequence.
5	not relevant	
4	Too many conditions in bramnch to be diagnosed.	The maximum number of 64 conditions per BEFA command output or WSB step-on assignment was exceeded. → Separate the branch into several secondary branches.
3	Illegal instruction in branch to be diagnosed.	In the BEFA or WSB branch you have used instructions that cannot be diagnosed. You can program these instructions either before or after the respective step branches. → Modify your cascade.
2	not relevant	
1	not relevant	
0	Warnng: Number of cascades = 0, without group fault indication	Parameter P3 of DIAGMMI has the value 0. → Change this value in accordance with your application.

Fig. 4-41 Status Message, DIAGMMI Program Module

4.12 MMI-MADAP Multidiagnostic Concept

The functional enhancement of the previous MMI-MADAP diagnostic concept now features the diagnostic procedure in conjunction with logging function and statistics for several stations (operating modes) on a single operator terminal.

Prerequisites:

- PROFIBUS COM-P module, part no. 107078590
- MMI-MADAP software version 1.3 and higher

4.12.1 Previous Diagnostic Concept

Each of the operator terminals BT 1 through 4 manages a cascade range (Station 1 - 4) which, upon calling the MMIMADAP program module from the panel, is written to parameters P1 and P2.

Functional description, usung BT1 as an example:

The MMIMADA program module prepares the parameters required for calling the DIAGMMI program module, activating thar module with the fixed assignment of BT1 ←→ Station 1. For the purpose of display and logging/recording, the operator terminal writes the obtained diagnostic data into the communication data module DM 234. This data is then picked up by the BT, acknowledged, displayed, and written into the protocol record. Only then can a new first-value error be entered.

The operator terminals are subject to the following assignments:

BT1 ←→ Station 1 communication DM = DM234

BT2 ←→ Station 2 communication DM = DM239

BT3 ←→ Station 3 communication DM = DM244

BT4 ←→ Station 4 communication DM = DM249

4.12.2 Multiple Diagnostics

The new multiple diagnostics provide a functional extension of the method previously used. This means that the stations labelled 1 through 4 (*base stations*) continue to point to the operator terminals (BT) 1 through 4. As a new feature however, 4 stations may be defined that may be freely assigned to the base stations. The resulting assignment of substations and of the associated cascade ranges occurs in DM255. As the assignments of cascades to the base stations remain unchanged, they continue to be expressed as parameters of the MMIMADAP program module. This ensures the expansion of the previous diagnostic concept without necessitating any other intervention in the PLC program.

Definitions and assignments:

For the purpose of diagnosing a ZS central processing unit, 4 additional *substations* can be assigned to existing *base stations* that are directly assigned to the 4 operator terminals.

The referred 4 substations can be freely distributed to the operator terminals designated 1 through 4, whereby multiple assignments are not permitted. This means that a single operator terminal can operate a maximum of 1 *base station* plus 4 *substations*.

Multiple diagnostics of stations located on different operator terminals (BT) is not possible.

The above definitions provide for the following combinations:

- 4 BT are able to diagnose a maximum of 8 stations
- 1 BT is able to handle a maximum of 5 stations

Definition of station assignments in DB255

Substation mask:

Assignment indicating which stations and terminals (in addition to the base station) are handled by the diagnostics.

Cascade range:

Definition of the cascade range that is assigned to a station.

The cascade range of the base station continues to be defined by the MMIMADAP parameters.

Definition of station assignments in DM255

In the table below, entries made by the operator terminal appear in boldface.

No.	Symbol	Type	Sign	Data field	F
:	:	:	:	:	:
		;		First-value communication counter, station 1-8 Value 500 (approx. 10 sec) acknowledges first value	
D156	EW_KZ1	Word	N		D
D158	EW_KZ2	Word	N		D
:	:	:	:	:	:
D168	EW_KZ7	Word	N		D
D170	EW_KZ8	Word	N		D
		;		First values to operator terminals (BT), station 5-8 Entry: 1:FV active, 2:BT has acknowledged	
D172	EW_St5	Word	N		H
D174	EW_St6	Word	N		H
D176	EW_St7	Word	N		H
D178	EW_St8	Word	N		H
		;		**Station mask assignment / BT 1-4** **No multiple bit assignment permitted.** **Bit0 > Stat5 (DM222) , Bit3 > Stat8 (DM225)**	
D180	**BT1StMsk**	**Word**	**N**	**0000**	**B**
D182	**BT2StMsk**	**Word**	**N**	**0000**	**B**
D184	**BT3StMsk**	**Word**	**N**	**0000**	**B**
D186	**BT4StMsk**	**Word**	**N**	**0000**	**B**
		;		Multiple station assignment HIGH byte = 1:Multiple assignment recognized LOW byte : First multiple assignment to be found	
D188	MehrBel	Word	N		H
		;		**Substation cascade ranges 5-8** **HIGH byte: Last cascade, LOW byte: First cascade** **Overlaps are possible**	
D190	**K_Ber5**	**Word**	**N**	**0000**	**H**
D192	**K_Ber6**	**Word**	**N**	**0000**	**H**
D194	**K_Ber7**	**Word**	**N**	**0000**	**H**
D196	**K_Ber8**	**Word**	**N**	**0000**	**H**
		;		BT Diagnostic participant (auxiliary marker)	
D198	BT_Teiln	Word			B
		;		Editing screen activated	
D200	StatEdit	Word			H
		;		Operator terminal stations 1-4, cascade ranges	
D202	K_Ber1	Word	N		H
D204	K_Ber2	Word	N		H
D206	K_Ber3	Word	N		H
D208	K_Ber4	Word	N		H
:	:	:	:	:	:
D 510		Word	N		H

Fig. 4-42 Extended Diagnostic Concept, Entries in DM 255

Functional principle of PLC program:

By means of the station masks the MMIMADAP program module determines if multistation mode was selected (D180/ 182/ 184/ 186 ≠ 0) and, in accordance with the information obtained, repeatedly executes the cascade diagnostics for the defined cascade ranges. If it is found that stations were assigned to several operator terminals, the diagnostics will be performed on the base station only.

Data module assignment and PROFIBUS object management

Base stations with fixed assignments to operator terminals 1 - 4

Exist. obj.	BT1 (Stn.1) DM234	Exist. obj.	BT2 (Stn.2) DM239	Exist. obj.	BT3 (Stn.3) DM244	Exist. obj.	BT4 (Stn.4) DM249
	First-value Diagnostics		First-value Diagnostics		First-value Diagnostics		First-value Diagnostics
28		37		46		55	
	Cascade information		Cascade information		Cascade information		Cascade information
29	Manual Diagnostics	38	Manual Diagnostics	47	Manual Diagnostics	56	Manual Diagnostics
	Station information		Station information		Station information		Station information

Substations for free assignment to base stations (BT) 1 - 4

New obj.	Stn.5 DM222	New obj.	Stn.6 DM223	New obj.	Stn.7 DM224	New obj.	Stn.8 DM225
	First-value Diagnostics		First-value Diagnostics		First-value Diagnostics		First-value Diagnostics
64		65		66		67	
	Cascade information		Cascade information		Cascade information		Cascade information
	Manual Diagnostics		Manual Diagnostics		Manual Diagnostics		Manual Diagnostics
	Station information		Station information		Station information		Station information

Fig. 4-43 Data Module Assignment and PROFIBUS Object Management

5 Interfaces – PLC <-> Operator Terminals

5.1 Definition

Introduction

The entirety of data interfaces for the MMI-MADAP is defined in the form of data modules (DM). Within individual data modules, the defined data ranges differentiate separate user and MMI-MADAP ranges.

The term *user ranges* denotes data interfaces on which the MMI-MADAP operator terminal makes data available to the user, and/or on which the user furnishes data destined for the MMI-MADAP operator terminal.

The referred data interfaces handle all communications between user and MMI-MADAP operator terminal.

All remaining data ranges are used by MMI-MADAP for exchanging information between the MMI-MADAP program modules and the MMI-MADAP operator terminal.

With regard to data interfaces, a differentiation is made between *screen-dependent* and *screen-independent* data ranges.

The screen-dependent data ranges are valid only in conjunction with the screen that is currently selected on the MMI-MADAP operator terminal.

The screen-independent data ranges are permanently defined. They are thus independent of the screen that is currently selected on the MMI-MADAP operator terminal.

All data interfaces are managed by the MMI-MADAP operator terminal. The updating of data interfaces with regard to the PLC program is both synchronous and asynchronous. The updating of the user data ranges must be monitored by the user.

☞ **The user data interfaces must be read and/or written in a cyclical fashion**

For the maximum of 4 MMI-MADAP operator terminals, *local* as well as *global* data modules are available.

The local data modules are designated for the respective corresponding MMI-MADAP operator terminal. The global data modules contain operator terminal data for general use.

5.2 List of Data Interfaces

DM no.	Name	Function	R/E	Length
DM 1	DB_K01	Kette 1 cascade information	R	100
:	:	:	:	:
DM 64	DB_K64	Kette 64 cascade information	R	100
DM 217	Taktzeit	Cycle time recording	R	230
DM 218	Auslast1	Machine usage data 1	R	512
DM 219	Auslast2	Machine usage data 2	R	512
DM 220	DP_Daten	DESI-DP data	R	512
DM 221	DP_Komm	DESI-DP communication channel 1	R	320
DM 222	Diag_St5	Station 5 diagnostic data (optional)	R	512
DM 223	Diag_St6	Station 6 diagnostic data (optional)	R	512
DM 224	Diag_St7	Station 7 diagnostic data (optional)	R	512
DM 225	Diag_St8	Station 8 diagnostic data (optional)	R	512
DM 226				
DM 227				
DM 228				
DM 229				
DM 230	BF1_DB	Communication / display data	R	512
DM 231	BF1_Stat	OPD status	R	512
DM 232	BF1_Sper	Movement lock functions	R	512
DM 233	BF1_Anw	Communication DM, 1 Object of 220 bytes	R	512
DM 234	BF1_Diag	Diagnostic result for BT1 = Station 1	R	512
DM 235	BF2_DB	Communication / display data	R	512
DM 236	BF2_Stat	Operand status	R	512
DM 237	BF2_Sper	Movement lock functions	R	512
DM 238	BF2_Anw	Communication DM, 1 Object of 220 bytes	R	512
DM 239	BF2_Diag	Diagnostic result for BT2 = Station 2	R	512
DM 240	BF3_DB	Communication / display data	R	512
DM 241	BF3_Stat	Operand status	R	512
DM 242	BF3_Sper	Movement lock functions	R	512
DM 241	BF3_Anw	Communication DM, 1 Object of 220 bytes	R	512
DM 244	BF3_Diag	Diagnostic result for BT3 = Station 3	R	512
DM 245	BF4_DB	Communication / display data	R	512
DM 246	BF4_Stat	Operand status	R	512
DM 247	BF4_Sper	Movement lock functions	R	512
DM 246	BF4_Anw	Communication DM, 1 Object of 220 bytes	R	512
DM 249	BF4_Diag	Diagnostic result for BT4 = Station 4	R	512
DM 250	BF1_4Anw	DM250 for all BT, 2 objects of 220 bytes ea.	R	512
DM 251	ResKObj1	Reserved for future communication objects	R	
DM 252	ResKObj2	Reserved for future communication objects	R	
DM 253	EA_SK	I/O assignment and SC table	R	512
DM 254	AL_DP_D	Display of machine usage & DP diagnostics	R	512
DM 255	BF_Globa	DM, valid for all BTs	R	512

Fig. 5-1 MMI-MADAP Data Modules

5.3 Description of User Data Interfaces

5.3.1 Data Modules DM1-64 – Cascade Data 1-64

Data modules DM1 through DM64 are required for sequential control management, and provide all data essential to system control. An accompanying data module must be generated for each step module.

Data module contents DM1 through DM64, "DB_K01" through "DB_K64": cascade information for KETTE1 through KETTE64

	Symbol	**Explanation**	**Data format**	**Entry caused by: K: KETTE M: MMI-MADAP A: User**
D00	nnFehler	Error bits	binary	K
D02	nnKettNr	Cascade no., n (1-64)	decimal	K
D04	nnSchAnz	Number of steps in cascade	decimal	K
D06	nnBaWahl	Operating mode selection	binary	K
D08	nnBaMldg	Confirmed operating mode	binary	K
D10	nnSchr-1	Step number, preceding step	decimal	K
D12	nnSchr.	Step number, current step	decimal	K
D14	nnSchr.S	Step number, Set Step	decimal	M
D16	nnBEFA	Command output for all steps	decimal	K
D18	nn-KWA	Wait time, actual value	dec. x 100 ms	K/A
D20	nn-KUE	Monitoring time, actual value	dec. x 100 ms	K/A
D22	nnINT0	internal use		
D24	nnINT1	internal use Bit 2 MADAP aktive bit	binary	M
D26	nnINT2	intern verwendet		
D28	nnINT3	intern verwendet		
D30	nnBa_Ext	Operating mode selection for external operator terminals	binary	M
D32	nnINT4	internal use		
::	::	internal use		
D48	nnSyn16	Synchronization steps 1 thru 16	binary	K
::	::	::		K
D62	nnSyn128	Synchronization steps 113 thru 128	binary	K
D64	nnSch16	Steps 1 thru 16	binary	K
::	::	::		K
D78	nnSch128	Steps 113 thru 128	binary	K
D80	nnBef16	Command output, steps 1 thru 16	binary	K
::	::	::		K
D94	nnBef128	Command output, steps 113 thru 128	binary	K

Fig. 5-2 Data Module Contents, Cascade Data

5.3.2 Data Module DM217 – Machine Cycle Times

Data module contents, DM217, "Taktzeit" (Cycle Time): machine cycle time recording

No.	Symbol	Type	Sign	Data field	F
		;		Display values for cycle times (TZ) 1-48	
D 0	TZ01	Word	N		H
D 2	TZ02	Word	N		H
:	:	:	:	:	:
D 92	TZ47	Word	N		H
D 94	TZ48	Word	N		H
D 96	TZint3	Word	N		H
D 98	TZint4	Word	N		H
		;		Number of cycle times to be processed	
D 100	TZ_ANZ	Word	N		D
		;		Start bits for cycle times	
D 102	STA16-01	Word	N		B
D 104	STA32-17	Word	N		B
D 106	STA48-33	Word	N		B
		;		Stop bits for cycle times	
D 108	STP16-01	Word	N		B
D 110	STP32-17	Word	N		B
D 112	STP48-33	Word	N		B
		;		Time base bits for cycle times	
		;		Recording accuracy: 0 = 0.1 sec; 1 = 1.0 sec	
D 114	BAS16-01	Word	N		B
D 116	BAS32-17	Word	N		B
D 118	BAS48-33	Word	N		B
		;		Current measured cycle time values	
D 120	AE-TZ01	Word	N		H
D 122	AE-TZ02	Word	N		H
:	:	:	:	:	:
D 212	AE-TZ47	Word	N		H
D 214	AE-TZ48	Word	N		H
D 216	HW_0.1s	Word	N		H
D 218	HW_1s	Word	N		H
D 220	KorrWert	Word	N		H
D 222	TZint5	Word	N		H
D 224		Word	N		H
D 226		Word	N		H
D 228		Word	N		H

Fig. 5-3 Data Module Contents, Machine Cycle Times

5.3.3 Data Modules DM 218 & DB219 – Machine Usage

Data module contents, DM218 "Auslast1": Machine usage data 1

No.	Symbol	Type	Sign	Data field	F
D 0	%S1PrT1	Word	N	Production, % value, Shift 1 record, Day 1 (current)	D
:	:	:	:	:	:
D 26	%S1PrT14	Word	N	Day 14	D
D 28	S1Prsec	Word	N	Current recording time Seconds	D
D 30	S1Prmin	Word	N	Minutes	D
D 32	%S2PrT1	Word	N	Production, % value, Shift 2 record, Day 1 (current)	D
:	:	:	:	:	:
D 58	%S2PrT14	Word	N	Day 14	D
D 60	S2Prsec	Word	N	Current recording time Seconds	D
D 62	S2Prmin	Word	N	Minutes	D
D 64	%S3PrT1	Word	N	Production, % value, Shift 3 record, Day 1 (current)	D
:	:	:	:	:	:
D 90	%S3PrT14	Word	N	Day 14	D
D 92	S3Prsec	Word	N	Current recording time Seconds	D
D 94	S3Prmin	Word	N	Minutes	D
D 96	S1StT1	Word	N	Piece count actual, Shift 1 record, Day 1 (current)	D
:	:	:	:	:	:
D 122	S1StT14	Word	N	Day 14	D
D 124	S1Mult	Word	N	Multiplier (Machine cycles x Multipl. = Pcs. Day 1)	D
D 126	S1Mzykl	Word	N	Machine cycles	D
D 128	S2StT1	Word	N	Piece count actual, Shift 2 record, Day 1 (current)	D
:	:	:	:	:	:
D 154	S2StT14	Word	N	Day 14	D
D 156	S2Mult	Word	N	Multiplier (Machine cycles x Multipl. = Pcs. Day 1)	D
D 158	S2Mzykl	Word	N	Machine cycles	D
D 160	S3StT1	Word	N	Piece count actual, Shift 3 record, Day 1 (current)	D
:	:	:	:	:	:
D 186	S3StT14	Word	N	Day 14	D
D 188	S3Mult	Word	N	Multiplier (Machine cycles x Multipl. = Pcs. Day 1)	D
D 190	S3Mzykl	Word	N	Machine cycles	D
D 192	PrT1	Word	N	Production, % value, Daily record, Day 1 (current)	D
:	:	:	:	:	:
D 218	PrT14	Word	N	Day 14	D
D 220	Prsec	Word	N	Current recording time Seconds	D
D 222	Prmin	Word	N	Minutes	D
D 224	JahrT1	Word	N	Date recording, Year Day 1 (current)	D
:	:	:	:	:	:
D 250	JahrT14	Word	N	Day 14	D
D 252		Word	N		D
D 254		Word	N		D
D 256	MET1	Word	N	Machine On, % value, Daily record, Day 1 (current)	D
:	:	:	:	:	:
D 282	MET14	Word	N	Day 14	D
D 284	MEsec	Word	N	Current recording time Seconds	D
D 286	MEmin	Word	N	Minutes	D
D 288	KTT1	Word	N	No Parts, % value, Daily record, Day 1 (current)	D
:	:	:	:	:	:
D 314	KTT14	Word	N	Day 14	D
D 316	KTsec	Word	N	Current recording time Seconds	D
D 318	KTmin	Word	N	Minutes	D
D 320	PVT1	Word	N	Buffer Full, % value, Daily record, Day 1 (current)	D
:	:	:	:	:	:
D 346	PVT14	Word	N	Day 14	D
D 348	PVsec	Word	N	Current recording time Seconds	D
D 350	PVmin	Word	N	Minutes	D

No.	Symbol	Type	Sign	Data field	F
D 352	StgT1	Word	N	Fault, % value, Daily record, Day 1 (current)	D
:	:	:	:	:	:
D 378	StgT14	Word	N	Day 14	D
D 380	Stgsec	Word	N	Current recording time Seconds	D
D 382	Stgmin	Word	N	Minutes	D
D 384	SST1	Word	N	Stillstand %-Wert Tageserfassung, Day 1 (current)	D
:	:	:	:	:	:
D 410	SST14	Word	N	Day 14	D
D 412	SSsec	Word	N	Current recording time Seconds	D
D 414	SSmin	Word	N	Minutes	D
D 416	DTT1	Word	N	Date recording - Day, Day 1 (current)	H
:	:	:	:	:	:
D 442	DTT14	Word	N	Day 14	H
D 444		Word	N		H
D 446		Word	N		H
D 448	DMT1	Word	N	Date recording - Month, Day 1 (current)	D
:	:	:	:	:	:
D 474	DMT14	Word	N	Day 14	D
D 476		Word	N		D
D 478		Word	N		H
:	:	:	:	:	:
D 510	END124	Word	N		H

Fig. 5-4 Data Module Contents, Machine Usage (1)

Data module contents, DM219 "Auslast2": Machine usage data 2

No.	Symbol	Type	Sign	Data field	F
D 0	S1VONH	Word	N	Start of work shift, Shift 1 (from) Hour	D
D 2	S1VONM	Word	N	Minute	D
D 4	S1P1VONH	Word	N	Start of break 1, Shift 1 (from) Hour	D
D 6	S1P1VONM	Word	N	Minute	D
:	:	:	:	:	:
D 24	S1P6VONH	Word	N	Start of break 6, Shift 1 (from) Hour	D
D 26	S1P6VONM	Word	N	Minute	D
D 28	S1BISH	Word	N	End of work shift, Shift 1 (to) Hour	D
D 30	S1BISM	Word	N	Minute	D
D 32	S1P1BISH	Word	N	End of break 1, Shift 1 (to) Hour	D
D 34	S1P1BISM	Word	N	Minute	D
:	:	:	:	:	:
D 52	S1P6BISH	Word	N	End of break 6, Shift 1 (to) Hour	D
D 54	S1P6BISM	Word	N	Minute	D
D 56	S2VONH	Word	N	Start of work shift, Shift 2 (from) Hour	D
D 58	S2VONM	Word	N	Minute	D
D 60	S2P1VONH	Word	N	Start of break 1, Shift 2 (from) Hour	D
D 62	S2P1VONM	Word	N	Minute	D
:	:	:	:	:	:
D 80	S2P6VONH	Word	N	Start of break 6, Shift 2 from) Hour	D
D 82	S2P6VONM	Word	N	Minute	D
D 84	S2BISH	Word	N	End of work shift, Shift 2 (to) Hour	D
D 86	S2BISM	Word	N	Minute	D
D 88	S2P1BISH	Word	N	End of break 1, Shift 2 (to) Hour	D
D 90	S2P1BISM	Word	N	Minute	D
:	:	:	:	:	:
D 108	S2P6BISH	Word	N	End of break 6, Shift 2 (to) Hour	D
D 110	S2P6BISM	Word	N	Minute	D

No.	Symbol	Type	Sign	Data field	F
D 112	S3VONH	Word	N	Start of work shift, Shift 3 (from) Hour	D
D 114	S3VONM	Word	N	Minute	D
D 116	S3P1VONH	Word	N	Start of break 1, Shift 3 (from) Hour	D
D 118	S3P1VONM	Word	N	Minute	D
:	:	:	:	:	:
D 136	S3P6VONH	Word	N	Start of break 6, Shift 3 (from) Hour	D
D 138	S3P6VONM	Word	N	Minute	D
D 140	S3BISH	Word	N	End of work shift, Shift 3 (to) Hour	D
D 142	S3BISM	Word	N	Minute	D
D 144	S3P1BISH	Word	N	End of break 1, Shift 3 (to) Hour	D
D 146	S3P1BISM	Word	N	Minute	D
:	:	:	:	:	:
D 164	S3P6BISH	Word	N	End of break 6, Shift 3 (to) Hour	D
D 166	S3P6BISM	Word	N	Minute	D
D 168	SOLL_S1	Word	N	Nominal/setpoint value, Shift 1	D
D 170	SOLL_S2	Word	N	Nominal/setpoint value, Shift 2	D
D 172	SOLL_S3	Word	N	Nominal/setpoint value, Shift 3	D
D 174	ERF_PARA	Word	N	Active logging parameters: HIGH byte = Default, LOW byte = transformed by logic	B
:	:	:	:	:	:
D 194	S1	Word	N	Shift 1: Length in minutes	D
D 196	S1INFO	Word	N	Information (bit 0 = Shift active)	B
D 198	S1P1	Word	N	Break 1, length in minutes	D
:	:	:	:	:	:
D 218	S1P6	Word	N	Break 6, length in minutes	D
D 220	100%S1	Word	N	100 % = Shift length - Σ breaks	D
D 222	S2	Word	N	Shift 2: Length in minutes	D
D 224	S2INFO	Word	N	Information (bit 0 = Shift active)	B
D 226	S2P1	Word	N	Break 1, length in minutes	D
:	:	:	:	:	:
D 246	S2P6	Word	N	Break 6, length in minutes	D
D 248	100%S2	Word	N	100 % = Shift length - Σ breaks	D
D 250	S3	Word	N	Shift 3: Length in minutes	D
D 252	S3INFO	Word	N	Information (bit 0 = Shift active)	B
D 254	S3P1	Word	N	Break 1, length in minutes	D
:	:	:	:	:	:
D 274	S3P6	Word	N	Break 6, length in minutes	D
D 276	100%S3	Word	N	100 % = Shift length - Σ breaks	D
:	:	:	:	:	:
D 384	S1St1S	Word	N	Shift 1 setpoint data, Day 1 (current)	D
:	:	:	:	:	:
D 410	S1St14S	Word	N	Day 14	D
D 412		Word	N		D
D 414		Word	N		D
D 416	S2St1S	Word	N	Shift 2 setpoint data, Day 1 (current)	D
:	:	:	:	:	:
D 442	S2St14S	Word	N	Day 14	D
D 444		Word	N		D
D 446		Word	N		D
D 448	S3St1S	Word	N	Shift 3 setpoint data, Day 1 (current)	D
:	:	:	:	:	:
D 474	S3StT14S	Word	N	Day 14	D
D 476		Word	N		H
D 478		Word	N		H
:	:	:	:	:	:
D 510	SCHTEND	Word	N		H

Fig. 5-5 Data Module Contents, Machine Usage (2)

5.3.4 Data Modules DM220 & DM221 – DESI-DP Diagnostics

Data module contents, DM220 "DP_Daten": DESI-DP data

No.	Symbol	Type	Sign	Data field	F
		;		Switching matrix address (BM-DP12 DIP switch)	
D 0	KoppAdr	Word	N		D
		;		Control of start-up behaviour & data communications	
D 2	AnlaufV	Word	N		H
		;		DM numbers for communication channels	
D 4	1DB-KK	Word	N		D
D 6	2DB-KK	Word	N		D
D 8	3DB-KK	Word	N		D
D 10	4DB-KK	Word	N		D
D 12	5DB-KK	Word	N		D
D 14	6DB-KK	Word	N		D
D 16	7DB-KK	Word	N		D
D 18	8DB-KK	Word	N		D
		;		High-priority communication channel	
		;		(DM no. from 1DM comm chan. - 8DM comm chan.)	
D 20	PrioDB	Word	N		D
		;		Communication channel for Auto-execution	
		;		of classified slave diagnostics (KSD)	
		;		(DM no. from 1DM comm chan. - 8DM comm chan.)	
D 22	KSD-DB	Word	N		D
		;		DM offset for KSD_DB (classified slave diagnostics)	
D 24	Off-KSD	Word	N		D
		;		Error and status messages	
D 26	F/Stat	Word	N		B
		;		FIFO parameters	
D 28	SendOff	Word	N		H
D 30	SendLng	Word	N		H
D 32	RecOff	Word	N		H
D 34	RecLng	Word	N		H
		;		Up to end of module: internal processing data only	
D 36	DPint	Word	N		H
:	:	:	:	:	:
D 510		Word	N		H

Fig. 5-6 Data Module Contents, DESI_DP Data

Data module contents, DM221 "DP_Komm": DESI-DP communication channel 1

No.	Symbol	Type	Sign	Data field	F
		;		Processing status	
D 0	B_St	Word	N		B
		;		Global status	
D 2	G_St	Word	N		B
		;		Minute / Second	
D 4	Z/m_s	Word	N		H
		;		Day / Hour	
D 6	Z/T_h	Word	N		H
		;		Year / Month	
D 8	Z/M_W	Word	N		H
		;		Switching matrix address / Weekday	
D 10	KF_WT	Word	N		H
		;		Receive status	
D 12	E_St	Word	N		H
		;		Transmit and Receive data length	
D 14	S_E_DatL	Word	N		H
		;		Transmit data and commands	
D 16	SDat1	Word	N		H
D 18	Sdat2	Word	N		H
D 20	SDat3	Word	N		H
		;		Receive data up to D318	
D 22	Edat	Word	N		H
:	:	:	:	:	:
D 318		Word	N		H
		;		Switching matrix address converted to DIP switch	
D 320	DIP_Adr	Word	N		H
		;		ZSx I/O address offeset	
D 322	ZS_Offs	Word	N		H

Fig. 5-7 Data Module Contents, DESI-DP Communication Channel 1

5.3.5 Data Modules – BT 1-4 Communication & Display Data

- MMI-MADAP operator terminal 1, DM230
- MMI-MADAP operator terminal 2, DM235
- MMI-MADAP operator terminal 3, DM240
- MMI-MADAP operator terminal 4, DM245

Data ranges:

- Screen number, MMI-MADAP operator terminal
- Terminal identifier of MMI-MADAP operator terminal
- Lamp test user prompt
- Cursor position, cascade selected for manual diagnostics
- Synchronization mask bits
- Command code for cascade/step
- Movement keys

Screen number, MMI-MADAP operator terminal

D0	Word	Number of screen on display

The number of the screen that is currently displayed on the MMI-MADAP operator terminal is entered in data word D0.

Using the screen number, the user is able to determine which data is currently present on the screen-dependent data interfaces, and/or when data may be written to specific screen-dependent data ranges.

MMI-MADAP operator terminal ID

D14 through D18	ASCII ASCII	User-defined terminal ID max. 6 characters

The designation for the MMI-MADAP operator terminal is entered in ASCII format in data word D14 through D18.

In the event that the operator terminal ID is entered by the user, it will be automatically added to the display upon opening the base screen (main menu) on the MMI-MADAP operator terminal. It will subsequently be displayed on each screen.

Lamp test user prompt

D116	Word	Lamp test prompt

If a lamp test user prompt is displayed on the MMI-MADAP operator terminal, a 1_{bin} will be entered in data word D116.

The user can read this data word and initiate a lamp test.

Cursor position, cascade selected for manual diagnostics

D118	Word	Cursor position for manual diagnostics indicated in diagnostic screen.

The cursor position of the cascade selected in the Manual diagnostics screen on the MMI-MADAP operator terminal is entered in data word D118 (value range if selected $1\text{-}64_{dec}$, otherwise 0).

Synchronizing mask bits

D120.0 through D127-.7	Bit Bit	Synchronization mask bits for Automatic D120.0 (cascade 1) thru 127.7 (cascade 64) Bit = LOW Participation in synchronization Bit = HIGH No participation in synchronization

The synchronization mask bits in data words D120 through D126 are read and interpreted by the MMI-MADAP operator terminal.

In the event that specific cascades are not to participate in synchronization, they can be masked by the user.

Operator terminal Live message

D154	Word	Life cycle counter

Starting at 0, this data word counts cyclically upwards. Each time the operator terminal registers in D30 (BF life marker), the counter is flushed. If the counter reaches the value of 2001, this means that the BF has not signalled to D30 for the past 10 seconds, and the counter will stop. You can interpret this value and specify suitable responses to its occurrence.

Command code, cascade/step

D256 through D286	Word Word	Cascade and step number of current movement screen HIGH Byte = cascade number LOW Byte = step number

Dependent upon the selected movement screen, the current command code for all of the 16 movement functions is stored by the MMI-MADAP operator terminal in data words D256 through D286.

- D256 thru D270: cascade & step no., left half of screen
- D272 thru D286: cascade & step no., right half of screen

Dependent upon movement screen and movement key, this command code could be manipulated by the user for the purpose of special functions. The number of the movement screen can be interpreted in data word D0.

☞ **The data range is valid for all movement screens.**

Movement keys

D292.0 through D293.7	Bit	Statuses of the 16 movement keys in the movement screen LOW Byte = Left half of screen HIGH Byte = Right half of screen

The movement keys selected in the movement screen of the MMI-MADAP operator terminal are entered in data word D292.

Dependent upon the respective movement screen, the movement keys can be interpreted for special functions. The number of the movement screen can be taken from data word D0.

☞ **The data range is valid for all movement screens.**

Data module contents, DM230 "BF1_DB": Communication/display data (example of operator terminal 1)

No.	Symbol	Type	Sign	Data field	F
		;		Number of displayed screen	
D 0	1Bild-Nr	Word	N		D
		; ; ;		K-Segm. + upper/lower half f. status DF HBy = upper (1)/ lower (0) DF segment half LBy = K-segment	
D 2	1DF-Seg	Word	N		H
		;		Number of DM being processed	
D 4	1DB-Nr	Word	N		D
		;		Length of selected DM	
D 6	1DB-Laen	Word	N		D
		;		Number of I/O and EI/EO bytes, "local" ZS	
		;		HIGH byte=EI/EO, LOW byte=I/O	
D 8	1EA_ZS	Word	N		H
		;		Number of first linked DM	
D 10	1Erst_DB	Word	N		D
		;		Reserved	
D 12	1dirBild	Word	N		D
		;		User ID	
D 14	1Anw-K	ASCII	N	Operator terminal 1	
		;		Number of cascades (number of last cascade)	
D 20	1KettAnz	Word	N		D
		;		Return message (error code) from DIAG PM	
D 22	1Diag	Word	N		B
		;		Parameters for the DIAG PM, managed internally	
D 24	1DIAG1	Word	N		H
D 26	1DIAG2	Word	N		H
		;		PLC Type code code + ZS no with CL5xx (x=1-3)	
D 28	1SPS-Typ	Word	N	CL350=0350, CL400=0400, CL500=050x, CL51x=150x, CL501=450x	H
		;		Life marker, operator terminal	
D 30	1BFlebt	Word	N		H
D 32	1Start_K	Word	N		H
				Security Level	
D 34	1Sec_Lev	Word	N		H
		;		Operator terminal number	
D 36	1BF_Nr	Word	N		H
		;		Auto-synch synchronization range	:
D 38	1ASynBer	Word	N		H
:	:	:	:	:	:
D 46		Word	N		H
		;		Statuses, power-up conditions 1 through 16	
D 48	1E-Bed1	Word	N		B
		;		Statuses, power-up conditions 17 through 32	
D 50	1E-Bed2	Word	N		B
		;		Codes for 32 Power-up conditions indirect bit address	
D 52	1E-Cod1	Word	N		H
D 114	1E-Cod32	Word	N		H
		;		Lamp test Station 1	
D 116	1LT	Word	N		H
		;		Cursor position, manual diagnostics in Diag. screen	
D 118	1K-Curs	Word	N		D
		;		Synchronization mask bits for Auto	
D 120	1A_SynM1	Word	N		B
D 122	1A_SynM2	Word	N		B
D 124	1A_SynM3	Word	N		B
D 126	1A_SynM4	Word	N		B

No.	Symbol	Type	Sign	Data field	F
		; ;		Synchronization request from operator terminal Bit0: request, Bit8: was requested	
D 128	1SynAnf	Word	N		H
		;		Operating mode bits for synchronization	
D 130	1BA_Syn1	Word	N		B
D 132	1BA_Syn2	Word	N		B
D 134	1BA_Syn3	Word	N		B
D 136	1BA_Syn4	Word	N		B
		;		Synchronization results	
D 138	1E_Syn1	Word	N		B
D 140	1E_Syn2	Word	N		B
D 142	1E_Syn3	Word	N		B
D 144	1E_Syn4	Word	N		B
		;		Synch auxiliary bits B0=Synch poss.; B1=Auto synch	
D 146	1HiB_Syn	Word	N		H
D 148	1int1	Word	N		H
D 150	1int2	Word	N		H
D 152	1int3	Word	N		H
		;		BF Life cycle counter	
D 154	1BFlebtZ	Word	N		H
D 156	1SynAnst	Word	N		H
:	:	:	:	:	:
D 190	1BewB0	Word	N		H
		;		16 End pos. codes (ind. bit addr.) in Movement scrn	
D 192	1E-End1	Word	N		H
:	:	:	:	:	:
D 222	1E-End16	Word	N		H
		;		16 Action codes (ind. bit addr.) in Movement screen	
D 224	1E-Akt1	Word	N		H
:	:	:	:	:	:
D 254	1E-Akt16	Word	N		H
		;		16 Command codes cascade/step in Movement scrn	
D 256	1E-Bew1	Word	N		H
:	:	:	:	:	:
D 286	1E-Bew16	Word	N		H
		; ;		End position statuses in movement screen HIGH byte = right, LOW byte = left; 1=bit0	
D 288	1Endl	Word	N		B
		; ;		Action statuses in movement screen HIGH byte = right, LOW byte = left; 1=bit0	
D 290	1Akt	Word	N		B
		; ;		Synchronization result statuses in movement screen HIGH byte = right, LOW byte = left; 1=bit0	
D 292	1Syn	Word	N		B
		; ;		Movement key statuses in movement screen HIGH byte = right, LOW byte = left; 1=bit0	
D 294	1BewT	Word	N		B
		;		New movement screen opened (handshake)	
D 296	1neuBild	Word	N		D
				Movement lock for active screen	H
D 298	1BewSper	Word	N		H
		;		Map of function keys	
D 300	1FT	Word	N		B
		;			
D 302	1DB-Komm	Word	N		D
D 304	1DB-Stat	Word	N		D
D 306	1DB-DIAG	Word	N		D
D 308	1KNr	Word	N		H
D 310	1KAnz	Word	N		H

No.	Symbol	Type	Sign	Data field	F
		; ;		Synchronization triggered externally (oper. terminal) w/ "1" in LOW byte; HIGH byte: internal transition bits	
D 312	SynExt	Word	N		H
D 314	1SynZaeh	Word	N		H
D 316	1H-lKett	Word	N		H
D 318	1H-lBew	Word	N		H
		; ;		Universal status display, with control line 1 (prepared) Operand identifier	
D 320	1OPDKz1	Word	N		H
		;		DM number	
D 322	1DBNr1	Word	N		H
		;		DM length	
D 324	1DBL1	Word	N		H
		;		Byte address	
D 326	1ByAdr1	Word	N		H
		;		Control enable when 1	
D 328	1StFrg1	Word	N		H
		;		Display value	
D 330	1AWert1	Word	N		H
		;		Conbtrol value	
D 332	1SWert1	Word	N		H
D 334		Word	N		H
D 336	1OPDKz2	Word	N		H
:	:	:	:	:	:
D 348	1SWert2	Word	N		H
D 350		Word	N		H
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
D 462		Word	N		H
D 464	1OPDKz10	Word	N		H
:	:	:	:	:	:
D 476	1SWert10	Word	N		H
D 478		Word	N		H
D 480	1OPDKz11	Word	N		H
:	:	:	:	:	:
D 492	1SWert11	Word	N		H
D 494		Word	N		H
D 496	1OPDKz12	Word	N		H
:	:	:	:	:	:
D 510	1StFrg	Word	N		H

Fig. 5-8 Data Module Contents, Communication / Display Data

5.3.6 Data Modules – BT 1-4 Status Display Data

- MMI-MADAP operator terminal 1, DM231
- MMI-MADAP operator terminal 2, DM236
- MMI-MADAP operator terminal 3, DM241
- MMI-MADAP operator terminal 4, DM246

Data ranges:

- Data contents of data types selected on MMI-MADAP operator terminal

The data modules are used by the MMIMADAP program module to store the statuses and/or data contents of selected data ranges of the status display on the MMI-MADAP operator terminal. The data is read and displayed by the MMI-MADAP operator terminal.

Dependent upon selection, the following data ranges are stored in their entirety:

- Inputs / Extended input fields
- Outputs / Extended output fields
- Markers / Special markers
- System data range
- Times / Counters
- Data field / Data buffer
- Data modules

Data module contents, DM231 "BF1_Stat": Operand status (example of operator terminal 1)

No.	Symbol	Type	Sign	Data field	F
		;		The statuses of operands selected in operator terminal 1 are deposited here for display purposes.	
D 0	Stat_BF1	Word	N		H
D 2		Word	N		H
:	:	:	:	:	:
D 506		Word	N		H
D 508		Word	N		H
D 510		Word	N		H

Fig. 5-9 Data Module Contents, Operand Status

5.3.7 Data Modules – BT 1-4 Movement Blocks

- MMI-MADAP operator terminal 1, DM232
- MMI-MADAP operator terminal 2, DM237
- MMI-MADAP operator terminal 3, DM242
- MMI-MADAP operator terminal 4, DM247

Data ranges:

- Movement block for a movement screen, with return of blocking code

Dependent upon specific events, specific user-selected movement screens can be blocked within the data modules.

- D0 = Unused
- D1 - D64 = Movement screen 1 through 64
- D65 = Unused

How to use the movement blocks:

Each movement screen has a **data byte** permanently assigned to it. Dependent upon specific events, you can write to this data byte a user-designated number code within the range of 1 to 96. If the movement screen is then opened, the screen will display the following default message:

Actions Are Blocked! Code: nn

This text may be freely modified by yourself.

Permanent messages:

Code 97: reserved

Code 98: **Current Movements Blocked!**
No active screen number. Upon changing a movement screen group by pressing a function key, this message will be returned until the new screen number is available in the PLC.

Code 99: **All Movements Blocked!**
This is a global movement block that is initiated by parameter P6 of the MMIMADAP program module.

Data module contents, DM232 "BF1_Sperr": Movement blocks, (example of operator terminal 1)

No.	Symbol	Type	Sign	Data field	F
		; ;		Block of movements in the screens corresponding to the byte no. (0 = free). The following applies to individual movement screen groups: Byte x0 = free; byte x1-x8 = scrn 1-8; byte x9 = free. Example: Value 3 in byte 11 blocks all movements in movement screen group 2, scrn 1. The entered value represents a code that is displayed on the terminal. Bewegungsbildgruppe 1, Bild 1-8	
D 0	1Sp01/--	Word	N		H
D 2	1Sp03/02	Word	N		H
D 4	1Sp05/04	Word	N		H
D 6	1Sp07/06	Word	N		H
D 8	1Sp--/08	Word	N		H
		;		Movement screen group 2, screens 1-8	
D 10	1Sp11/--	Word	N		H
D 12	1Sp13/12	Word	N		H
D 14	1Sp15/14	Word	N		H
D 16	1Sp17/16	Word	N		H
D 18	1Sp--/18	Word	N		H
		;		Movement screen group 3, screens 1-8	
D 20	1Sp21/--	Word	N		H
D 22	1Sp23/22	Word	N		H
D 24	1Sp25/24	Word	N		H
D 26	1Sp27/26	Word	N		H
D 28	1Sp--/28	Word	N		H
		;		Movement screen group 4, screens 1-8	
D 30	1Sp31/--	Word	N		H
D 32	1Sp33/32	Word	N		H
D 34	1Sp35/34	Word	N		H
D 36	1Sp37/36	Word	N		H
D 38	1Sp--/38	Word	N		H
		;		Movement screen group 5, screens 1-8	
D 40	1Sp41/--	Word	N		H
D 42	1Sp43/42	Word	N		H
D 44	1Sp45/44	Word	N		H
D 46	1Sp47/46	Word	N		H
D 48	1Sp--/48	Word	N		H
		;		Movement screen group 6, screens 1-8	
D 50	1Sp51/--	Word	N		H
D 52	1Sp53/52	Word	N		H
D 54	1Sp55/54	Word	N		H
D 56	1Sp57/56	Word	N		H
D 58	1Sp--/58	Word	N		H
		;		Movement screen group 7, screens 1-8	
D 60	1Sp61/--	Word	N		H
D 62	1Sp63/62	Word	N		H
D 64	1Sp65/64	Word	N		H
D 66	1Sp67/66	Word	N		H
D 68	1Sp--/68	Word	N		H
		;		Movement screen group 8, screens 1-8	
D 70	1Sp71/--	Word	N		H
D 72	1Sp73/72	Word	N		H
D 74	1Sp75/74	Word	N		H
D 76	1Sp77/76	Word	N		H
D 78	1Sp--/78	Word	N		H
D 510		Word	N		H

Fig. 5-10 Data Module Contents, Movement Blocks

5.3.8 Data Modules – BT 1-4 Diagnostic Results

- DM234 MMI-MADAP operator terminal 1 (base station 1)
- DM239 MMI-MADAP operator terminal 2 (base station 2)
- DM244 MMI-MADAP operator terminal 3 (base station 3)
- DM249 MMI-MADAP operator terminal 4 (base station 4)

Also, optional data modules DM222 through DM225 (substations 5-8)

Data ranges:

- Automatic diagnostics of cascade first-value errors
- Manual diagnostics of faulty cascades
- Cascade statuses

With the aid of the MMIMADAP program module, a cascade range-specific diagnostic routine can be parameterized.

Up to four different cascade ranges can be diagnosed per central processing unit.

This facilitates a station-specific diagnostic routine at the operator terminal.

☞ **For additional information, refer to sections discussing diagnostics and operator terminal connectivity.**

Data module contents, DM234 "BF1_Diag": Diagnostic results, (example of operator terminal 1)

No.	Symbol	Type	Sign	Data field	F
		; ; ; ;		First-value message ========== Control flags From here: Object 1 for PROFIBUS communications	
D 0	1FLAGS	Word	N		B
		;		Date / Time	
D 2	1TT/MM	Word	N		H
D 4	1JJ/SS	Word	N		H
D 6	1MIN/SEK	Word	N		H
D 8	1WoT/__	Word	N		H
		;		Cascade no. / Step no.	
D 10	1KNr/SNr	Word	N		H
		;		Station number / Step module number	
D 12	1BstNr	Word	N		H
		;		Number of messages	
D 14	1AnzMeld	Word	N		H
		;		Instruction codes 1 - 64 for Auto diagnostics	
D 16	1_01ANW	Word	N		H
:	:	:	:	:	:
D 142	1_64ANW	Word	N		H
		;			
D 144	1EWint1	Word	N		H
D 146	1EWint2	Word	N		H
D 148	1EWint3	Word	N		H
		;		Cascade information, cascade 1 - 64	
D 150	1K_Inf01	Word	N		H
:	:	:	:	:	:
D 218	1K_Inf35	Word	N		H
		;		From here: Object 2 for PROFIBUS communications	
D 220	1K_Inf36	Word	N		H
:	:	:	:	:	:
D 276	1K_Inf64	Word	N		H
		; ; ;		From here: Manual diagnostics ========== Cascade number / Step number	
D 278	1K/S-Nr	Word	N		H
		;		Module type / Module number	
D 280	1Baust	Word	N		H
		;		Cascade status / Number of messages	
D 282	1Z/Anz	Word	N		H
		;		Instruction codes 1 - 64 for Manual diagnostics	
D 284	H01.ANW	Word	N		H
:	:	:	:	:	:
D 410	H64.ANW	Word	N		H
D 412		Word	N		H
		;		Station list, Last cascade / First cascade	
D 420	1St1	Word	N		H
:	:	:	:	:	:
D 434	1St8	Word	N		H
D 436		Word	N		H
D 438		Word	N		H
		;		From here: Internal data, no communication data	
D 440	1noKomm	Word	N		H
:	:	:	:	:	:
D 510		Word	N		H

Fig. 5-11 Data Module Contents, Diagnostic Results

5.3.9 Data Module DM253 – I/O Assignment and SC Table

Data ranges:

- Display of I/EI and O/EO assignment for up to 4 central processing units.
- Display of information about modules located in the PLC module rack.
- Version management of standard PLC program modules.

The MMIMADAP program module furnishes the E/EI and O/EO assignment, plus all data from the SC table to the MMI-MADAP operator terminal.

The data range encompassing data words D380 through D472 is available for standard PLC module version management.

Versions management for standard PLC program modules

The data range D380 through D422 is reserved for BOSCH-proprietary standard PLC program modules.

D380	Word	MMI-MADAP version ID
D382	ASCII	"MMIMADAP"
D390	Word	KETTE version ID
D392	ASCII	"KETTE"
D400	Word	DIAGMMI version ID
D402	ASCII	"DIAGMMI"
D410	Word	DPSTATUS version ID
D412	ASCII	"DPSTATUS"
D420	Word	Reserved: Bosch version ID
D422	ASCII	Reserved: Bosch module designation
D430	Wort	User module 1, version ID
D382	ASCII	User module 1, module designation
..	..	..
D470	Wort	User module 5, version ID
D472	ASCII	User module 5, module designation

Version ID:

- HIGH Byte: Bit 0-3 = leading comma 0-16
- HIGH Byte: Bit 4-7 = intermediate version A-F
- LOW Byte: Bit 0-3 = trailing comma 0-16

Data module contents, DM253 "EA_SK": I/O assignment and SC table

No.	Symbol	Type	Sign	Data field	F
D 0		ASCII	N	internal	
D 32	0EBel15	Word	N	Central processing unit 0: I assignment, bytes 0-63	B
D 34	0EBel31	Word	N		B
D 36	0EBel47	Word	N		B
D 38	0EBel63	Word	N		B
D 40	1EBel15	Word	N	Central processing unit 1-3: I assignment, bytes 15-63	B
:	:	:	:	:	:
D 64	0ABel15	Word	N	Central processing unit 0: O assignment, bytes 0-63	B
D 66	0ABel31	Word	N		B
D 68	0ABel47	Word	N		B
D 70	0ABel63	Word	N		B
D 72	1ABel15	Word	N	Central processing unit 1-3: O assignmt., bytes 15-63	B
:	:	:	:	:	:
D 96	0EZBel15	Word	N	Central processing unit 0: EI assignment, bytes 0-63	B
D 98	0EZBel31	Word	N		B
D 100	0EZBel47	Word	N		B
D 102	0EZBel63	Word	N		B
D 104	1EZBel15	Word	N	Central processing unit 1-3: EI assignmt., bytes 15-63	B
:	:	:	:	:	:
D 128	0AZBel15	Word	N	Central processing unit 0: EO assignment, bytes 0-63	B
D 130	0AZBel31	Word	N		B
D 132	0AZBel47	Word	N		B
D 134	0AZBel63	Word	N		B
D 136	1AZBel15	Word	N	Central processing unit 1-3: EO assign., bytes 15-63	B
:	:	:	:	:	:
D 164	0ZS=akt	Word	N		H
D 166	1ZS=akt	Word	N		H
D 168	2ZS=akt	Word	N		H
D 170	3ZS=akt	Word	N		H
:	:	:	:	:	:
D 184	SysBerA	Word	N		H
D 186	DB-Ziel	Word	N		H
D 188	SK/ZS400	Word	N	System coordinator / ZS40x	D
D 190	SKVers	ASCII	N	Version	
D 192	1Typ	ASCII	N	Module 1 Type	
D 194	1PerAdr	Word	N	Peripheral address	D
D 196	1EZ/AZ	Word	N	EI/EO assignment	D
D 198	1E/A	Word	N	I/O assignment	D
D 200	1BlAdr	Word	N	Block address	D
D 202	1BlAnz	Word	N	Block count (number of blocks)	D
D 204	1BGvorh	Word	N	Module in rack	H
D 206	1Vers	ASCII	N	Version	
D 208	2Typ	ASCII	N	Modules 2-10 (identical to module 1)	
:	:	:	:	:	:
D 350	10Vers	ASCII	N		
D 352		Word	N		H
		;		Version ID of "DIAGMMI" program module	
D 354	V-DIAG	Word	N		H
		;		Version ID of "MMIMADAP" program module	
D 356	V-MMI	Word	N		H
		;		Version ID of "KETTE" program module	
D 358	V-KETTE	Word	N		H
D 360	BG-Nr	Word	N		D
D 362	Bild-Nr	Word	N		D
D 364	SKerst	Word	N		H
D 366	HandGlob	Word	N		H
:	:	:	:	:	:

No.	Symbol	Type	Sign	Data field	F
		;		Version ID and text for default function modules	
D 380	0PB_Vers	Word	N		H
D 382	0PB_Name	ASCII	N	MMIMADAP	
D 390	1PB_Vers	Word	N		H
D 392	1PB_Name	ASCII	N	KETTE	
D 400	2PB_Vers	Word	N		H
D 402	2PB_Name	ASCII	N	DIAGMMI	
D 410	3PB_Vers	Word	N		H
D 412	3PB_Name	ASCII	N	MMIAUSL	
D 420	4PB_Vers	Word	N		H
D 422	4PB_Name	ASCII	N	DPSTATUS	
D 430	5PB_Vers	Word	N		H
D 432	5PB_Name	ASCII	N	.....	
D 440	6PB_Vers	Word	N		H
D 442	6PB_Name	ASCII	N	.....	
D 450	7PB_Vers	Word	N		H
D 452	7PB_Name	ASCII	N	.....	
D 460	8PB_Vers	Word	N		H
D 462	8PB_Name	ASCII	N	.....	
D 470	9PB_Vers	Word	N		H
D 472	9PB_Name	ASCII	N	.....	
D 480		Word	N		H
D 506		Word	N		H
D 508		Word	N		H
D 510		Word	N		H

Fig. 5-12 Data Module Contents, I/O Assignment and SC Table

5.3.10 Machine Usage and PROFIBUS-DP Diagnostics

The data module DM254 provides the data relating to machine usage and PROFIBUS-DP diagnostics for display on the BT100 operator terminal.

Data module contents, DM254 "AL_DP_D": Machine usage display & DP-Diagnostics

No.	Symbol	Type	Sign	Data field	F
		;		Shifts – Production	
D 0	PS1heut	Word	N		D
D 2	PS1gest	Word	N		D
D 4	PS2heut	Word	N		D
D 6	PS2gest	Word	N		D
D 8	PS3heut	Word	N		D
D 10	PS3gest	Word	N		D
		;		Shifts – Actual part/piece counts	
D 12	ISS1heut	Word	N		D
D 14	ISS1gest	Word	N		D
D 16	ISS2heut	Word	N		D
D 18	ISS2gest	Word	N		D
D 20	ISS3heut	Word	N		D
D 22	ISS3gest	Word	N		D
		;		Days – Production	
D 24	P_Theut	Word	N		D
D 26	P_Tgest	Word	N		D
		;		Date entry – Year	
D 28	Jah_heut	Word	N		D
D 30	Jah_gest	Word	N		D
		;		Days – Machine On	
D 32	ME_Theut	Word	N		D
D 34	ME_Tgest	Word	N		D
		;		Days – No Parts	
D 36	kT_Theut	Word	N		D
D 38	kT_Tgest	Word	N		D
		;		Days – Buffer Full	
D 40	Pv_Theut	Word	N		D
D 42	Pv_Tgest	Word	N		D
		;		Days – Fault	
D 44	St_Theut	Word	N		D
D 46	St_Tgest	Word	N		D
		;		Days – Standstill	
D 48	Ss_Theut	Word	N		D
D 50	Ss_Tgest	Word	N		D
		;		Date entry – Day	
D 52	Tag_heut	Word	N		D
D 54	Tag_gest	Word	N		D
		;		Date entry – Month	
D 56	Mon_heut	Word	N		D
D 58	Mon_gest	Word	N		D
		;		Shifts – Nominal/setpoint part/piece counts	
D 60	SSS1heut	Word	N		D
D 62	SSS1gest	Word	N		D
D 64	SSS2heut	Word	N		D
D 66	SSS2gest	Word	N		D
D 68	SSS3heut	Word	N		D
D 70	SSS3gest	Word	N		D
D 72		Word	N		H
:	:	:	:	:	:

No.	Symbol	Type	Sign	Data field	F
		;		Cycle times, mirrored from DM217 for display	
D 100	01Taktz	Word	N		H
D 102	02Taktz	Word	N		H
:	:	:	:	:	:
D 192	47Taktz	Word	N		H
D 194	48Taktz	Word	N		H
:	:	:	:	:	:
		;		Recorded value before and after LOGIK	
D 204	ErfassW	Word	N		B
		;		Shift and current production duration	
		;		Sn_100%: Shift length ./. Sum of all breaks	
		;		Sn_akt: accrued production length	
D 206	S1_100%	Word	N		D
D 208	PS1_akt	Word	N		D
D 210	S2_100%	Word	N		D
D 212	PS2_akt	Word	N		D
D 214	S3_100%	Word	N		D
D 216	PS3_akt	Word	N		D
		;		"Shift 1-3 active" message	
D 218	Sch1akt	Word	N		B
		;		From here: PROFIBUS-DP diagnostics	
		;		Processing status	
D 220	BearbSt	Word	N		B
		;		Global status	
D 222	GlobalSt	Word	N		B
		;		Minute / Second	
D 224	Zeit_m_s	Word	N		H
		;		Day / Hour	
D 226	Zeit_T_h	Word	N		H
		;		Year / Month	
D 228	Zeit_M_W	Word	N		H
		;		Switching matrix address / Weekday	
D 230	Zeit_J_K	Word	N		H
		;		Receive status	
D 232	EmpfangS	Word	N		H
		;		Receive length	
D 234	EmpfangL	Word	N		H
		;		Receive data start	
		;		DP-Diagnostics interpretation mask	
D 236	EmpfDatA	Word	N		H
		;		Slave(s) not reachable	
D 238	1EmpfD1	Word	N		B
D 240	1EmpfD2	Word	N		B
D 242	1EmpfD3	Word	N		B
D 244	1EmpfD4	Word	N		B
D 246	1EmpfD5	Word	N		B
D 248	1EmpfD6	Word	N		B
D 250	1EmpfD7	Word	N		B
D 252	1EmpfD8	Word	N		B
		;		One or more slaves report configuration fault	
D 254	2EmpfD1	Word	N		B
D 256	2EmpfD2	Word	N		B
:	:	:	:	:	:
D 266	2EmpfD7	Word	N		B
D 268	2EmpfD8	Word	N		B
		;		One or more slaves report statistical diagnostics	
D 270	3EmpfD1	Word	N		B
D 272	3EmpfD2	Word	N		B
:	:	:	:	:	:
D 282	3EmpfD7	Word	N		B
D 284	3EmpfD8	Word	N		B

No.	Symbol	Type	Sign	Data field	F
		;		One or more slaves report expanded diagnostics	
D 286	4EmpfD1	Word	N		B
D 288	4EmpfD2	Word	N		B
:	:	:	:	:	:
D 298	4EmpfD7	Word	N		B
D 300	4EmpfD8	Word	N		B
		;		One/more slaves not ready for cyclical data exchange	
D 302	5EmpfD1	Word	N		B
D 304	5EmpfD2	Word	N		B
:	:	:	:	:	:
D 314	5EmpfD7	Word	N		B
D 316	5EmpfD8	Word	N		B
		;		One or more slaves report slave error	
D 318	6EmpfD1	Word	N		B
D 320	6EmpfD2	Word	N		B
:	:	:	:	:	:
D 330	6EmpfD7	Word	N		B
D 332	6EmpfD8	Word	N		B
		;		Active shift not to be processed IF # 0	
:	:	:	:	:	:
D 510	noS_akt	Word	N		D

Fig. 5-13 Data Module Contents, Machine Usage and DP Diagnostics

5.3.11 Data Module DM255 – Global Operating Data

Data ranges:

- 511 serial user messages coming/going
- 128 parallel user status messages, priority-controlled
- List of available data modules
- Cycle times, watchdog PLC
- Time values of organization modules OM18 thru OM25
- PLC status and error bits
- Operator terminal and PLC system times

In data module DM255, the MMIMADAP program module provides the above mentioned data ranges – with the exception of serial and parallel messages – for the MMI-MADAP operator terminal.

Serial and parallel messages are used to display and manage machine statuses and errors on the MMI-MADAP operator terminal. The messages are generated on the PLC by the user, and are then made available in data module DM255.

The messages appear on all operator terminals that are connected to the central processing unit.

Serial messages coming/going

DM255/D0	Serial messages coming/going
D0.0-D0.8	Mssage number 1 - 511_{dec}
D0.9	Acknowledge all active messages
D.10	Message received (from oper. terminal 1)
D.11	Message received (from oper. terminal 2)
D.12	Message received (from oper. terminal 3)
D.13	Message received (from oper. terminal 4)
D.14	Message comes
D.15	Message goes

Serial messages that are coming/going are managed by the user in the PLC program, whereby the transmission of a message is automatically read by the MMI-MADAP operator panel, and the reception is acknowledged.

Functional principle:

Coming messages:

- PLC — Message number (bit 0-8) entered with coming-bit (bit 14).
- Oper. term. — Acknowledgement of message reception (bit 10/BF1 - bit 13/BF4). Coming message is entered in "active message record."
- PLC — Once all operator terminals have acknowledged, acknowledgement bits must be deleted. A new message can be transmitted.

Going messages:

- PLC — Message number (bit 0-8) entered with going-bit (bit 15).
- Oper. term. — Acknowledgement of message reception (bit 10/BF1 - bit 13/BF4). Going message writes active message into "historic message record" and deletes it in "active message record."
- SPS — Once all operator terminals have acknowledged, acknowledgement bits must be deleted. A new active message can be acknowledged.

Acknowledging all active messages:

- PLC — Entering "all active messages" bit (bit 9).
- Oper. term. — The "all active messages" bit is deleted. Active messages are entered in "historic message record" and deleted from "active message record."
- PLC — A new coming message can be transmitted.

Parallel status messages

D2.0- D17.7	128 Bit parallel status messages

As status messages serve only display functions, they are not stored in the operator terminal.

The status messages entered by the user are automatically read by the MMI-MADAP operator terminal, and the assigned text messages are displayed.

Functional principle:

For each status message, one bit is available to the user.

D2.0	= HIGH	Display status message 1	**highest**
	= LOW	Do not display status message 1	**priority**
:	:	:	↓
D17.7	= HIGH	Display status message 128	**lowest**
	= LOW	Do not display status message 128	**priority**

Up to 128 parallel messages can be active at the same time. The MMI-MADAP operator terminal imposes a display priority, whereby status message 1 (D2.0) is assigned the highest, and status message 128 (D17.7) the lowest priority.

List of available data modulese

D32.0	Bit	Data module DM1 through DM255
through		available / not available
D63.7	Bit	

The MMIMADAP program module writes the available data modules into the list.

For each existing data module, the corresponding bit in the list is set HIGH. The process assigns bit D32.0 to data module DM1 and bit D63.7 to data module DM255.

In the PLC, the data module list update is initiated by the trigger pulse (Power-up or restart after STOP/RUN). One-time initiation also occurs upon selecting the base screen, and continuous update inititiation is effected by selecting the DM list on the MMI-MADAP operator terminal.

PLC status and error bits

		PLC status bits
D86.0	Bit	Battery fault
D86.1	Bit	Outputs disabled
D86.2	Bit	I/O fixed
D86.3	Bit	Buffer Full system command
D86.4	Bit	Free
D86.5	Bit	Status Message Active
D86.6	Bit	Free
D86.7	Bit	Free
		PLC error bits / PLC in Stop
D87.0	Bit	Addressing fault
D87.1	Bit	PM parameter error
D87.2	Bit	Non-existent PM called
D87.3	Bit	Module stack fault
D87.4	Bit	Application stack overrun / underrun
D87.5	Bit	Parameter instruction, system command
D87.6	Bit	No data module active
D87.7	Bit	Cycle time error

The MMIMADAP program module enters PLC status and error bits into the data word. The MMI-MADAP operator terminal reads the information and displays it on the MMI-MADAP operator terminal.

The user can read the information and utilize it in his program.

System time on operator terminal and PLC

		Time on MMI-MADAP operator terminal
D96	Word	Minute / Second
D98	Word	Day / Hour
D100	Word	Year / Month
D102	Word	Weekday
		Time on PLC
D104	Word	Minute / Second
D106	Word	Day / Hour
D108	Word	Year / Month
D110	Word	Weekday

If the system time is reset on a MMI-MADAP operator terminal, the terminal will transfer the time into the data module. As a consequence, the MMIMADAP program module synchronizes the PLC system time. Multiple operator terminals respond by cyclically synchronizing to the PLC system time.

The user can read the time and/or date and utilize it in his program.

Data module contents, DM255 "BF_Globa": Data module valid for all operator terminals

No.	Symbol	Type	Sign	Data field	F
		;		Serial messages, coming / going	
D 0	SeriMeld	Word	N		D
		;		Parallel status messages	
D 2	1ZustM	Word	N		B
D 4	2ZustM	Word	N		B
D 6	3ZustM	Word	N		B
D 8	4ZustM	Word	N		B
D 10	5ZustM	Word	N		B
D 12	6ZustM	Word	N		B
D 14	7ZustM	Word	N		B
D 16	8ZustM	Word	N		B
D 18		Word	N		H
:	:	:	:	:	:
		;		Transition bits for generating data module list upon displaying base screen	
D 30	Fl_DBL	Word	N		H
		;		Data module list, 1=available 0=not available	
D 32	1DBL	Word	N		B
D 34	2DBL	Word	N		B
D 36	3DBL	Word	N		B
D 38	4DBL	Word	N		B
D 40	5DBL	Word	N		B
D 42	6DBL	Word	N		B
D 44	7DBL	Word	N		B
D 46	8DBL	Word	N		B
D 48	9DBL	Word	N		B
D 50	10DBL	Word	N		B
D 52	11DBL	Word	N		B
D 54	12DBL	Word	N		B
D 56	13DBL	Word	N		B
D 58	14DBL	Word	N		B
D 60	15DBL	Word	N		B
D 62	16DBL	Word	N		B
		;		Maximally measured cycle time	
D 64	Zykl_max	Word	N		D
		;		Time of last cycle	
D 66	Zykl_akt	Word	N		D
		;		Watchdog setting	
D 68	WatchDog	Word	N		D
		;		Time values for time-controlled processing	
D 70	T-OB18	Word	N		D
D 72	T-OB19	Word	N		D
D 74	T-OB20	Word	N		D
D 76	T-OB21	Word	N		D
D 78	T-OB22	Word	N		D
D 80	T-OB23	Word	N		D
D 82	T-OB24	Word	N		D
D 84	T-OB25	Word	N		D
		;		PLC status and error bits	
D 86	Z-Bits	Word	N		B
		;		Handshake for diagnostic results	
D 88	EW_BF1	Word	N		H
D 90	EW_BF2	Word	N		H
D 92	EW_BF3	Word	N		H
D 94	EW_BF4	Word	N		H

No.	Symbol	Type	Sign	Data field	F
		;		System time on terminal	
D 96	BT-m/s	Word	N		H
D 98	BT-T/h	Word	N		H
D 100	BT-J/M	Word	N		H
D 102	BT-WT	Word	N		H
		;		System time on PLC	
D 104	SPS-m/s	Word	N		H
D 106	SPS-T/h	Word	N		H
D 108	SPS-J/M	Word	N		H
D 110	SPS-WT	Word	N		H
		;			
D 112	int1	ASCII	N		
:	:	:	:	:	:
		;		Machine usage active	
D 118	Ausl_akt	Word	N		H
D 120		Word	N		H
:	:	:	:	:	:
		;		First-value communication counter, station 1-8 Value 500 (approx. 10 sec) acknowledges first value	
D156	EW_KZ1	Word	N		D
D158	EW_KZ2	Word	N		D
:	:	:	:	:	:
D168	EW_KZ7	Word	N		D
D170	EW_KZ8	Word	N		D
		;		First values (FV) to operator terminals, station 5-8 Entry: 1:FV active, 2:oper. term. has acknowledged	
D172	EW_St5	Word	N		H
D174	EW_St6	Word	N		H
D176	EW_St7	Word	N		H
D178	EW_St8	Word	N		H
		;		Station mask assignment / BT1-4, Multiple bit assignments not permitted. Bit0 > Stat5 (DM222) , Bit3 > Stat8 (DM225)	
D180	BT1StMsk	Word	N	0000	B
D182	BT2StMsk	Word	N	0000	B
D184	BT3StMsk	Word	N	0000	B
D186	BT4StMsk	Word	N	0000	B
		;		Multiple station assignment HIGH byte=1:multiple / LOW byte:1st multiple assignment found	
D188	MehrBel	Word	N		H
		;		Substation cascade ranges 5-8, HIGH byte: last cascade, LOW byte: cascade 1, Overlaps are possible	
D190	K_Ber5	Word	N	0000	H
D192	K_Ber6	Word	N	0000	H
D194	K_Ber7	Word	N	0000	H
D196	K_Ber8	Word	N	0000	H
		;		BT_Diagnostics participants (auxiliary marker)	
D198	BT_Teiln	Word			B
		;		Editing screen active	
D200	StatEdit	Word			H
		;		Operator terminal cascade ranges 1-4	
D202	K_Ber1	Word	N		H
D204	K_Ber2	Word	N		H
D206	K_Ber3	Word	N		H
D208	K_Ber4	Word	N		H
:	:	:	:	:	:
D 510		Word	N		H

Fig. 5-14 Data Module Contents, Global Data

6 MMI-MADAP Operator Terminal

6.1 Introduction

Two software modules are available:

- Development module
- Runtime module

The development module facilitates the development of user-defined applications. It also provides for the extension of the existing MMI-MADAP application.

The runtime module facilitates the operation, diagnosing and visualization of the system without, however, the option to generate new screens.

MMI-MADAP is a software application for facilitating the operation, diagnostics and visualization of machine systems. Direct communication with the connected PLC controller is possible via the PROFIBUS-FMS or via a point-to-point connection.

MMI-MADAP provides 64 user screens for the purpose of implementing user-defined visualization and controller functions.

The standard software provides the following functions:

- Display of power-up criteria
- Execution of movements,
- User screens, e.g. system overview
- Status and information display of the communicating PLC controller
- Display of entire PLC data range
- Information regarding machine system via display of status and error text messages that are PLC-controlled
- Cascade diagnostics (Visualization: PLC links of first-value and sequence errors).

☞ **BOSCH documentation reference**

MMI-MADAP for System or Machine Operators — Software Manual	Part no. 1070 072 167

6.2 System Requirements for MMI-MADAP Operator Terminal

System requirements for MMI-MADAP software:

PC components	
Processor	Pentium 100 MHz or higher
Memory (RAM)	16 MB
Disk space	100 MB
Additionally, for communication via PROFIBUS:	
PROFIBUS PC card	PB-IF-03 PROFIBOARD, (Softing)
	CP 5412(A2), Mfg. Siemens

☞ **Only the use of the Bosch BT100 operator terminal assures the user of the perfect match between hardware and software!**

Software	
Operating system	DOS 6.22 and Microsoft Windows 3.11

☞ **The MMI-MADAP software is exclusively designed for use with Microsoft Windows 3.11.**

6.3 Software Installation

6.3.1 Supplied Software Files

- **MMI-MADAP diskettes:**
  Operating and visualization system for Bosch control systems.
- **FB.MMIMADAP diskette:**
  Function modules and PROFIBUS configuration for the PLC.

6.3.2 MMI-MADAP

Introduction

The CONFIG.SYS and AUTOEXEC.BAT file must be expanded and/or modified.

☞ **These two files are normally located in the root directory of drive C:\. Modifications can be made with any ASCII editor.**

- **CONFIG.SYS file**

The use of the MMI-MADAP software packet requires the following entries:

```
DEVICE=C:\WINDOWS\HIMEM.SYS

DEVICE=C:\DOS\SETVER.EXE

DOS=HIGH,UMB

FILES=99

DEVICEHIGH=C:\DOS\RAMDRIVE.SYS 160 512 /E
```

In addition, if a PROFIBUS module is to be used, the following modification will be required:

```
DEVICE=C:\DOS\EMM386.EXE NOEMS X=D000-DFFF
```

- **AUTOEXEC.BAT file**

The use of the MMI-MADAP software packet requires the following entries:

```
PROMPT $P$G

CLS

PATH C:\WINDOWS;C:\DOS;C:\;C:\MMIMADAP\BIN

SET TEMP=C:\DOS

; (NOTE: Please enter the following
; line prior to the installation!!)

C:\DOS\SMARTDRV /X

D:

MD MMIMADAP

CD MMIMADAP

COPY C:\MMIMADAP\*.DAT D:

C:

WIN
```

Subsequent to the installation of the MMI-MADAP operator terminal software the files named AUTOEXEC.MMI and CONFIG.MMI will be automatically generated in the C:\ root directory. The referred files contain the above mentioned entries, and can be renamed by the user to AUTOEXEC.BAT and CONFIG.SYS, respectively.

The Bosch BT100 operator terminal contains all data in the AUTO-EXEC.BAT and CONFIG.SYS files.

6.3.2.1 Installation diskettes

The MMI-MADAP software is supplied in the form of a set of diskettes that can be directly installed.

However, you can also download the MMI-MADAP software from the Bosch mailbox, and create your own set of installation diskettes.

The Bosch mailbox at +49 6062 7217 always provides the current version of the MMI-MADAP software in the MMIMADAP Filebox.

078580.EXE	Diskette1
078581.EXE	Diskette2
078582.EXE	Diskette3
078583.EXE	Diskette4
078584.EXE	Diskette5
078853.EXE	Diskette6

Download these files to the C:\ drive of your computer. Continue by inserting a blank diskette into drive A:\. At the DOS prompt, type the command MD MMIMADAP to create a new directory named A:\MMIMADAP. Type CD MMIMADAP to change to the new directory. Select the .EXE file on C:\078580.EXE. This will cause installation diskette 1 to be created. Use the same procedure for all remaining .EXE files.

You have now created the MMI-MADAP set of installation diskettes consisting of 6 diskettes. Please be sure to label the diskettes "MMI-MADAP/Disk 1 of 6" through diskettes consisting of 6 diskettes. Please be sure to label the diskettes "MMI-MADAP/Disk 1 of 6" through "MMI-MADAP/Disk 6 of 6".

6.3.2.2 Installing MMI-MADAP from Set of Diskettes

If you are performing a software upgrade and/or a new MMI-MADAP installation, and if applicable, precede the installation by saving your existing MMI-MADAP files under a new name in order to save your individualized definitions.

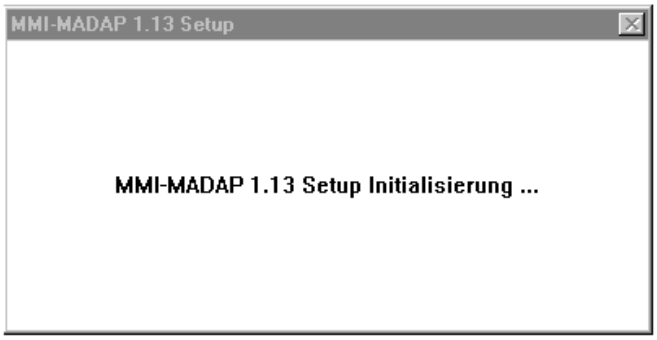
Installation

The installation uses Microsoft Windows 3.1 or 3.11 exclusively. Therefore be sure to begin by starting Windows.

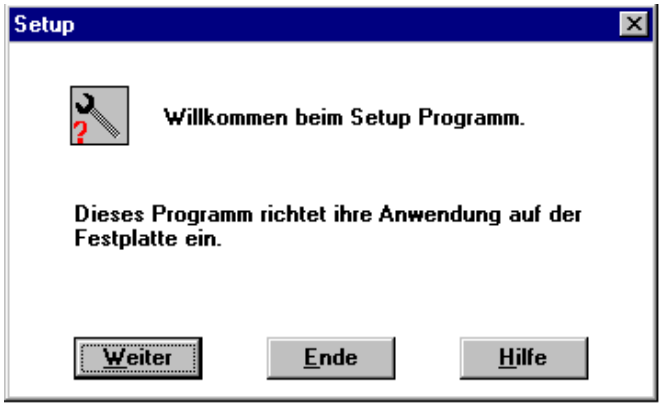
Insert Installation Diskette 1 into the disk drive.

Run the SETUP.EXE program located in A:\MMIMADAP, e.g. via the File Manager.

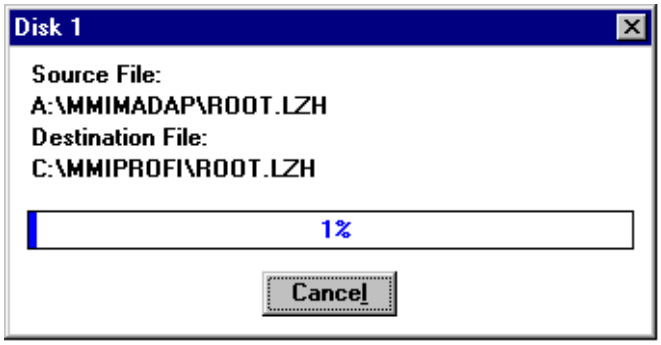
The MMI-MADAP Setup window will briefly appear. It contains the version number of the MMI-MADAP software packet that you are using.
Please be advised that the windows will appear in a German-language version only. Wherever required, specific explanations will be given in this manual.



In the Setup sign-on window that follows, press **Weiter** (Continue) to set up your program on the hard disk. (The **Weiter** button corresponds to the **Return** or **Enter** key.)



The installation will now commence.



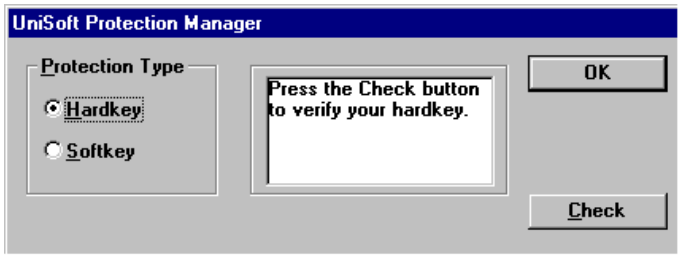
Please follow the user prompt to insert the next diskette when required.

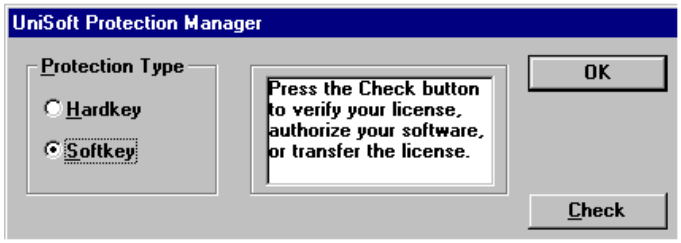
Software protection

Once the diskette installation has been concluded, you will be asked to select the desired software protection type.

In response, press the **Alt+TAB** key combination until the title UNISOFT PROTECTION MANAGER is highlighted.

As a default, the protection type is set to Hardkey protection.

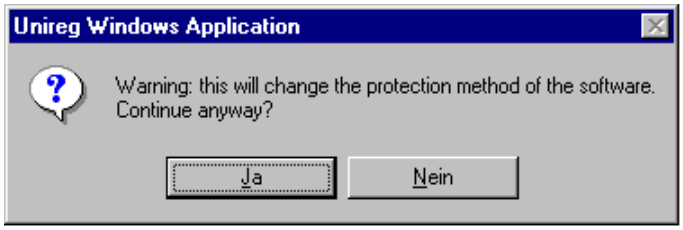




Continue by pressing **Alt+H** to select Hardkey or **Alt+S** to select Softkey protection.

Once this is done, press **Alt+C** to activate the Check function.

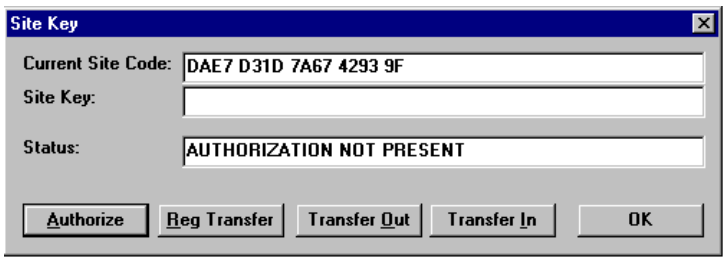
If you have changed the protection method, you will be alerted to this change by an appropriate system message.



In response to this message, press **Return**.

If a valid license is found, the licensing procedure is hereby concluded.

If a valid license is not found, a window containing an 18-digit Site Key number will open.



Write down this number, and send a fax to:

Fa. Bosch/Erbach. Attn. Mr. Kuschel, fax no. +49 6062 78 784 (where +49 denotes your access code required to get an outside line, plus the country code for Germany).

The fax form named MMIFAX.WRI is located in the MMIMADAP directory of the mailbox.

If you are a supplier or subcontractor for system projects, you are advised to ensure prior coordination with your customer regarding this matter.

In response to your fax message, you will receive the chargeable secret code, known as the *Site Key*. Once you have entered the verification number, select the **Authorize** button, followed by pressing **Return** again. This activates the text box. Press **TAB** to move to the **OK** button, and press **Return**.

To exit the licensing function, press **OK**. You can reach this button by repeatedly pressing the **TAB** key.

The next window to appear informs you that a copy of both the AUTOEXEC.MMI and CONFIG.MMI files were installed in the main directory, and that the original AUTOEXEC.BAT and CONFIG.SYS files were not modified. Acknowledge this message by pressing **Return**.

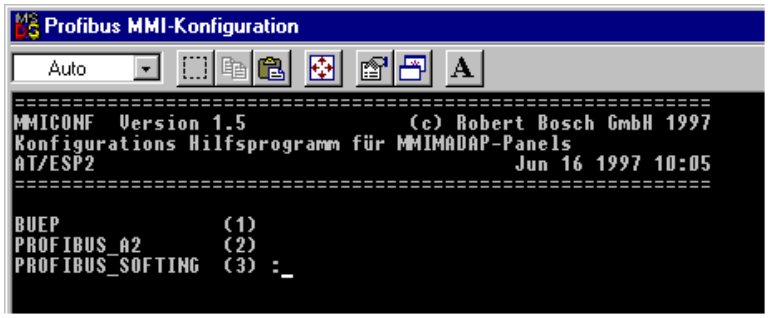
Once the licensing procedure has been concluded, the next step will be the configuration of the operator terminal.

6.3.2.3 Configuring the Operator Terminal

There exist three different options for establishing communications between the operator terminal and the control unit:

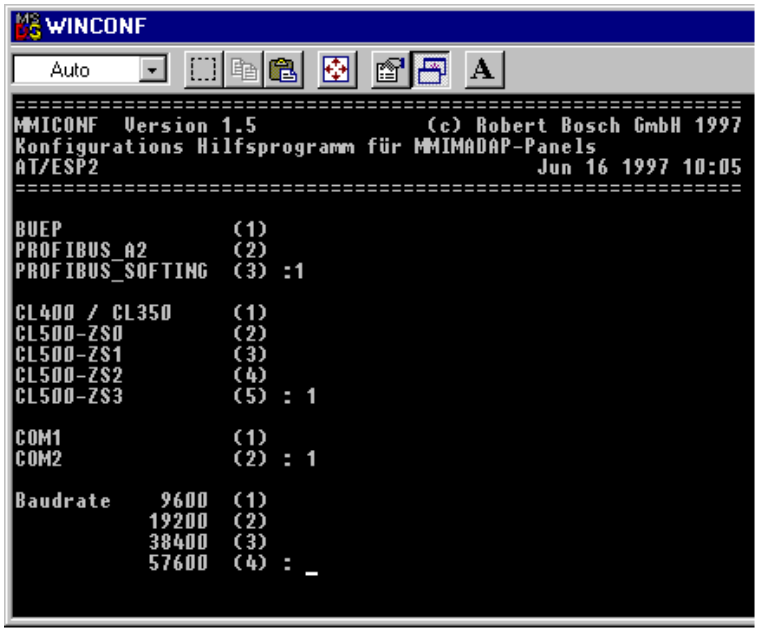
- Point-to-point connection, using the Bosch BUEP19E transmission protocol.
- PROFIBUS-FMS connection, using the CP5412-A2 hardware module manufactured by Siemens.
- PROFIBUS-FMS connection, using the PROFIBOARD hardware module manufactured by Softing Mfg.

At this point, the operator terminal configuration procedure provides you with a selection of desired PLC controller connections.



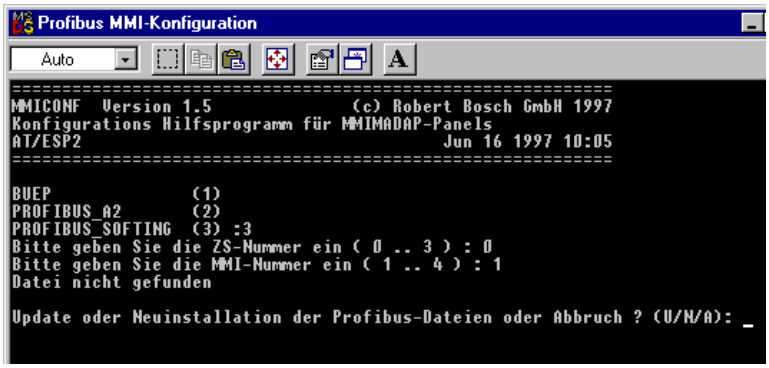
BUEP Driver

Subsequent to selecting the BUEP driver, enter the interface parameters. These must correspond to the PLC control unit parameters.



PROFIBUS CP5412-A2 and PROFIBOARD modules

This procedure requires the desired PROFIBUS station address for the operator terminal to be entered.



At the prompt:
`Bitte geben Sie die ZS-Nummer ein (0 .. 3)`, please enter the number of the central processing unit (ZSx ...). This will be a number between 0 and 3.

At the prompt:
`Bitte geben Sie die MMI-Nummer ein (1 .. 4)`,
please enter the number of the connected operator terminal. This will be a number between 1 and 4.

At the prompt:
`Update oder Neuinstallation der PROFIBUS-Dateien oder Abbruch?`
To select the right driver files, indicate whether this installation is an update of existing software (type **`U`**), or a new installation of the PROFIBUS files (type **`N`**). If you wish to cancel, type (**`A`**). Continue by selecting the PROFIBUS files that will be used to configure the system.

Once the driver type has been selected, the *.DRV driver files will be loaded.

New installation

Typing (**`N`**) copies all Bosch-proprietary forms for the selected combination of ZS central processing unit and operator terminal into the destination directory, C:\MMIMADAP\CONFIG.

BUEP001.DRV - BUEP050.DRV	BOSCH internal
SL2A2001.DRV - SL2A2050.DRV	BOSCH internal
SFMS001.DRV - SFMS050.DRV	BOSCH internal
BUEP051.DRV - BUEP130.DRV	Blank standard forms for the user
SL2A2051.DRV - SL2A2130.DRV	Blank standard forms for the user
SFMS051.DRV - SFMS130.DRV	Blank standard forms for the user

Update

Typing (ʊ) copies all Bosch-internal

forms for the selected combination of ZS central processing unit and operator terminal (e.g. SL2A2001.DRV - SL2A2050.DRV) into destination directory C:\MMIMADAP\CONFIG.
Also, the user forms that are assigned to the operator terminal (SL2A2051.DRV - SL2A2130.DRV or SFMS051.DRV - SFMS130.DRV) are copied from the respective user directory into destination directory C:\MMIMADAP\CONFIG.

User directory:

C:\PROFIUSE\1 Forms for operator terminal 1, 5, 9, 13

C:\PROFIUSE\2 Forms for operator terminal 2, 6, 10, 14

C:\PROFIUSE\3 Forms for operator terminal 3, 7, 11, 15

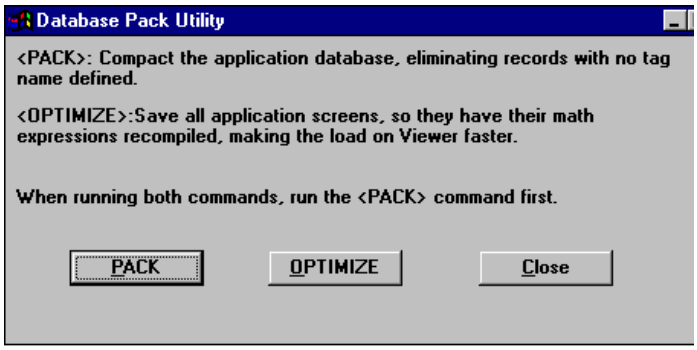
C:\PROFIUSE\4 Forms for operator terminal 4, 8, 12, 16

Packing

Once the installation has been performed correctly, it is an **absolute requirement** to optimize the database by compacting (packing) the files..

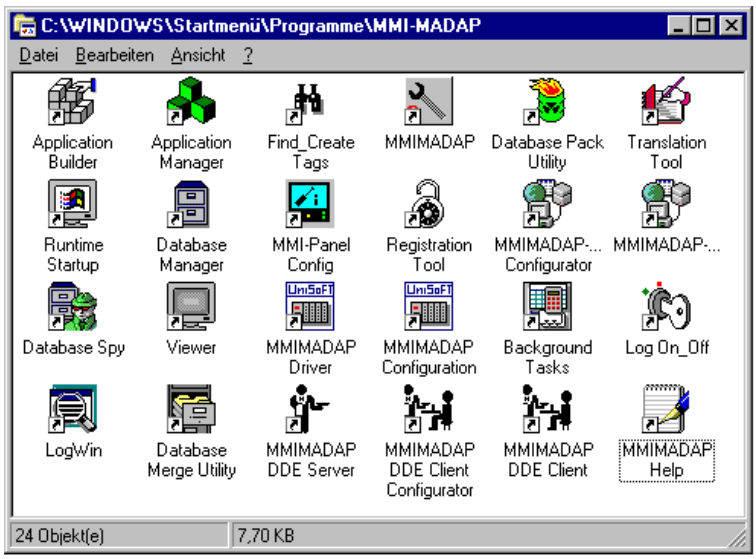
Also, the Setup Info window informs the user that, in case a Siemens CP5412-A2 card is being used, the system must be restarted subsequent to the installation. To start the packing program, press **OK**.

In the Database Pack Utility window, select the **PACK** button. Once the procedure has ended, select the **Close** button to exit the program.



MMI-MADAP Program group

All MMI-MADAP programs are located in the MMI-MADAP program group.



Starting MMI-MADAP

To start the software, double-click the MMIMADAP icon in the MMI-MADAP program group window.

NOTE:
When using the CP5412-A2 PROFIBUS module:

To ensure proper operation of the system subsequent to an installation, a restart of the operator terminal will be required.

Starting MMI-MADAP automatically

A restart of the MMI-MADAP operator terminal will cause an automatic re-start of the MMI-MADAP application. If you want to prevent this auto-restart, a modification of your WIN.INI file will be required. Use an editor to open the file, and remove the entry:

- RUN C:\MMIMADAP\BIN\DIAG.EXE

This concludes the installation and configuration of the MMI-MADAP software.

☞ **The WIN.INI file is normally located in the directory named C:\WINDOWS. Modifications to this file can be made with any ASCII editor.**

Starting MMI-MADAP manually

In the Windows Program Manager, select the MMI-MADAP directory. Start the MMI-MADAP software by double-clicking the **MMIMADAP icon** (shown below).

Changing operator terminal number after installation

In the Windows Program Manager, select the MMI-MADAP directory. Double-click the **MMI-Panel Config** icon (shown below), and make the required corrections.

Licensing operator terminal after installation

In the Windows Program Manager, select the MMI-MADAP directory. Double-click the **Registration Tool** icon (shown below), and make the required corrections.

6.3.3 Configuring Diagnostics

Introduction

In order to display diagnostic information about cascades and steps on the MMI-MADAP operator terminal, additional display information will be required.

Upon starting the MMI-MADAP application, this information is derived from the project symbol file and the respective cascade modules.

The project symbol file and step modules SCHRK1 through SCHRK64 (to the extent available) must be located in the directory C:\MMIMADAP\KETTEN.

The names for symbol file and step modules may be freely selected.

☞ **Please bear in mind that the symbol file as well as the step modules must have been created with the WinSPS application.**

Symbol file and step module installation

From your current PLC project directory, copy the current version of the project symbol file named xxx.SXS, as well as all step modules named SCHRK1.PXO through SCHRK64.PXO to C:\MMIMADAP\KETTEN (xxx = name of symbol file).

as a default, MMI-MADAP expects to see a symbol file bearing the name SYMBOL..SXS. If the name of your file is different, it is recommended that you rename your file accordingly. It is, however, possible to adapt the entry in the DIAG.INI file:

```
SymFile=C:\MMIMADAP\KETTEN\xxx.SXS
```

The DIAG.INI file is located in the C:\WINDOWS directory.

6.4 MMI-MADAP Directory Structure

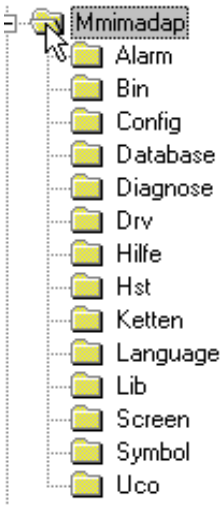


Fig. 6-1 Directory Tree, MMI-MADAP Operator Terminal

- **C:\MMIMADAP**

This directory contains all PROFIBUS configuration files, plus the generated *.DAT files and other system files.

- **C:\MMIMADAP\ALARM**

For each month, the MMI-MADAP software creates an alarm file. All alarm files are held in this directory.

- **C:\MMIMADAP\BIN**

This is the directory for the entire basic system, including, for example, all .EXE files.

- **C:\MMIMADAP\CONFIG**

This directory holds all standardization components and associated files.

- **C:\MMIMADAP\DATABASE**

This is the directory for the MMI-MADAP database.

- **C:\MMIMADAP\DIAGNOSE**

All protocol record files generated by the diagnostics function are stored in this directory.

- **C:\MMIMADAP\DRV**

This directory holds all required driver tools.

- **C:\MMIMADAP\HILFE**

The MMI-MADAP packet encompasses an online Help system. The required Help text files are located in this directory. All text files may be modified by the user with the aid of a text editor.

- **C:\MMIMADAP\HST**

This directory holds historic trend values used by the application.

- **C:\MMIMADAP\KETTEN**

For the purpose of displaying operand text, etc., the diagnostic module requires the cascade files and the symbol files used by the project. These files must be deposited here by the user.

- **C:\MMIMADAP\LANGUAGE**

This directory is used only internally for the different national language versions by the basic development and runtime software.

- **C:\MMIMADAP\LIB**

The UNISOFT development packet supplies existing symbol objects. These are located in this directory.

- **C:\MMIMADAP\SCREEN**

All MMI-MADAP screen are stored here.

- **C:\MMIMADAP\SYMBOL**

Standard functions created in the development editor can be stored here.

- **C:\MMIMADAP\UCO**

This directory is reserved for additional customer objects. (Partially written in C high-level language.)

6.5 Definition of MMI-MADAP Data Ranges

Introduction

The MMI-MADAP standard software packet defines different data ranges.

The term *data ranges* encompasses the following components:

- Online data
- Alarms (Alarm files)
- Math worksheets (Math files)
- Time-controlled processing (Scheduler files)
- Trend values (Trend files)
- Report outputs (Report files)
- Data management (Recipe files)

The entirety of data ranges is predefined and partially reserved for the user.

☞ **With the exception of the online data, all components are located in the C:\MMIMADAP\CONFIG directory.**

Online data

The MMI-MADAP software enables you to create new definitions while online. These definitions are stored in the form of *.DAT files in the C:\MMIMADAP directory.

The respective files can be randomly modified and integrated into new projects.

The copy function can be automated through the use of batch files or in the PROJEKT screen of the MMI-MADAP user interface.

The online data encompasses the following data groups:

- Power-up screens: Power-up text, conditions, softkey designations, screen titles, group designations
- Movement screens: Power-up text and functions, softkey designations, screen titles, group designations
- User screens: softkey designations, screen titles, group designations
- Parallel status messages
- Serial user messages
- Cascade synchronization definitions
- Screen assignments: Matrix for movement and user screens

Alarm files

- Definition:

The purpose of an alarm is to alert the user to unusual statuses occurring throughout the processing routines. This facilitates the initiation of required remedial procedures.

The following alarm files have been defined:

Range	Filename, *.ALR	Utilizer
1 - 10	ALARM001 - ALARM010	BOSCH
11-20	ALARM011 - ALARM020	Development
21-30	ALARM021 - ALARM030	Project / 1st user
31-40	ALARM031 - ALARM040	Project / 2nd user
41-50	ALARM041 - ALARM050	Project / 3rd user

Fig. 6-2 List of Defined Alarm Files

Alarm files 11 through 50 are reserved for the user.

Math files

- Definition:

The purpose of mathematics is the utilization of formulas and functions of the MMI-MADAP software.

The following math files have been defined:

Range	Filename, *.MAT	Utilizer
1 - 100	MATH001 - MATH0100	BOSCH
101-150	MATH101 - MATH150	Development
151-200	MATH151 - MATH200	Project / 1st user
201-250	MATH201 - MATH250	Project / 2nd user
251-300	MATH251 - MATH300	Project / 3rd user

Fig. 6-3 List of Defined Math Files

The mathematics files 101 through 300 are reserved for the user.

Scheduler files

- Definition:

The purpose of the scheduler is the initiation of functions in dependency of time intervals, date and calendar functions, and also in the case of value changes of specified variables.

The following scheduler files have been defined:

Range	Filename, *.SCH	Utilizer
1	SCHED001	BOSCH
2	SCHED002	Development
3	SCHED003	Project / 1st user
4	SCHED004	Project / 2nd user
5	SCHED005	Project / 3rd user

Fig. 6-4 List of Defined Scheduler Files

The scheduler files 3 through 5 are reserved for the user.

Trend files

- Definition:

The purpose of the trend files is the generation of online trends and historical trends. The objective of a graphical trend representation is to inform the user about the development of a given process by plotting the values of variables on a curve.

The following trend files have been defined:

Range	Filename, *.TRD	Utilizer
1 - 50	TREND001 - TREND050	BOSCH
51-70	TREND051 - TREND070	Development
71-90	TREND071 - TREND090	Project / 1st user
91-110	TREND091 - TREND110	Project / 2nd user
111-130	TREND111 - TREND130	Project / 3rd user

Fig. 6-5 List of Defined Trend Files

The trend files 50 through 130 are reserved for the user.

Report files

- Definition:

Creating of user-defined reports that can be written to the hard disk or output to a connected printer.

The following report files must be defined:

Range	Filename, *.REP	Utilizer
	ABCDEFGH.REP	BOSCH
	ABCDEFGH.REP	Development
	ABCDEFGH.REP	Project / 1st user
	ABCDEFGH.REP	Project / 2nd user
	ABCDEFGH.REP	Project / 3rd user

Fig. 6-6 List of User-Defined Report Files

Each user can use filenames of his choice. The filename extension will always the .REP. In order to prevent dual filename assignments, the users named DEVELOPMENT and PROJECT must precede the filename by an additional identifier.
For example:

PR1xxx.REP

Recipe files

- Definition:

Exchange of values between the central database and the files located on the hard disk.

The following recipe files must be defined:

Range	Filename, *.RCP	Utilizer
	ABCDEFGH.RCP	BOSCH
	ABCDEFGH.RCP	Development
	ABCDEFGH.RCP	Project / 1st user
	ABCDEFGH.RCP	Project / 2nd user
	ABCDEFGH.RCP	Project / 3rd user

Fig. 6-7 List of User-Defined Recipe Files

Each user can assign filenames of his choice. The filename extension will always the .REP. In order to prevent dual filename assignments, the users named DEVELOPMENT and PROJECT must precede the filename by an additional identifier.
For example:

PR1xxx.RCP

6.6 Online Project Design of Standard Functions

The MMI-MADAP user-specific project design function is divided into two areas:

- **Online project design**
  (refer also to chapter on online data)

and

- **User screen design**
  e.g. System overviews, production data management

☞ **The subject of online project design is discussed in detail in the documentation entitled "MMI-MADAP for System or Machine Operators."**

☞ **The steps involved in designing and configuring user screens are described in detail in the following section.**

6.7 Designing User Screens

User screens can be used to indicate system statuses and conditions, and to display them on the MMI-MADAP operator terminal.

This section will address the principles of displaying PLC information via user screens.

☞ **The design and creation of user screens will require the use of the development software. For detailed information on this subject, please refer to the documentation supplied with the development software.**

To provide an example, a simple system overview will be designed here. The objective is the graphical representation of all cascade operating modes by means of a colour change, and to display the system status and the produced quantity (part/piece count).

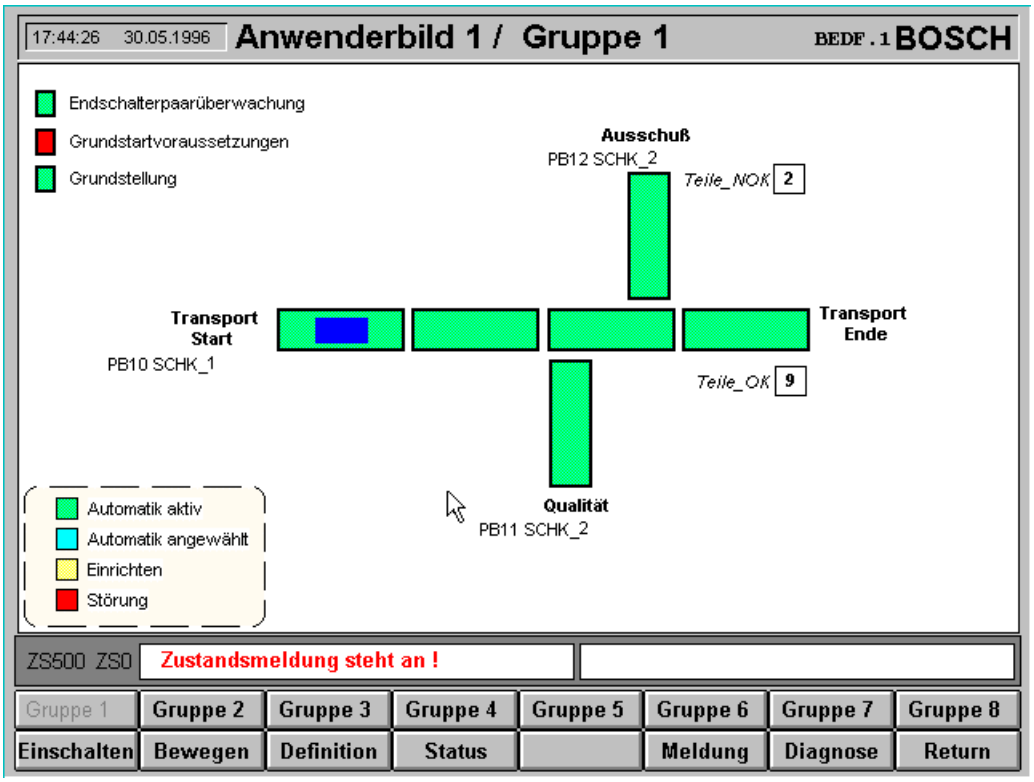


Fig. 6-8 Example of SYSTEM OVERVIEW User Screen

Basic procedure

In the design of user screens, the following steps are required:

- Activating user level
- Opening user screen
- Declaring variables
- Creating graphical objects
- Defining configuration sheet
- Testing user screen

Activating user level

A specific user level is required in order to create user screens. Please activate the required user level.

☞ **Please contact the system administrator to obtain the available user names and passwords.**

In the Windows Program Manager, select the MMI-MADAP directory. Start the Log On/Off software module by double-clicking the **Log On/Off icon** (shown below).

In the Log On dialog box, type the username and password, then select **OK** to close the application.

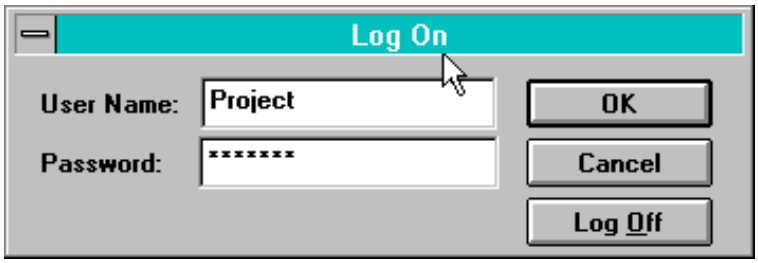


Fig. 6-9 Example of User Level Activation

Opening user screen

User screens are composed of two files, i.e., the file providing the background image (ANWXX.SCR), and the file for the user application (APPLXX.SCR). A total of 64 user screens are available.

☞ **The default background images are fully configured, and should not be modified by the user.**
In addition, the user may avail himself of the base screen (filename GRUNDANW.SCR). This screen is called from within the GRUND:SCR screen.

In the Windows Program Manager, select the MMI-MADAP directory. Start the Application Builder software module by double-clicking the **Application Builder icon** (shown below).

In the Application Builder, select the **Open** menu command to open the APPLXX.SCR for creating your applications.

($1 \leq xx \leq 64$.).

Declaring variables

As a first step, all variables required for the intended application should be declared:

In the Application Builder, select the **Tools** menu command to start the Database Manager.

For our system overview we require two variables of the Array Integer16 type.

The required variables are defined as follows:

- **Anlage** (system) for recording and reading PLC statuses, and
- **AnwBild1_aktiv** (User screen1 active) for enabling the read cycle from within the PLC.

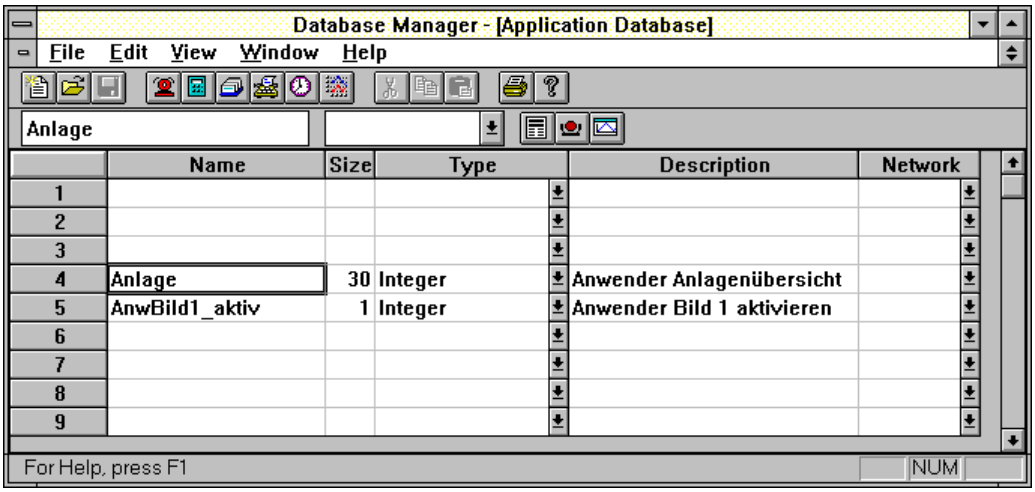


Fig. 6-10 Example of Declaration of Variables

The **AnwBild1_aktiv** variable is now assigned a function that will serve to activate the read cycle.

In this example, the read cycle on the PLC is to occur cyclically after the user screen has been selected.

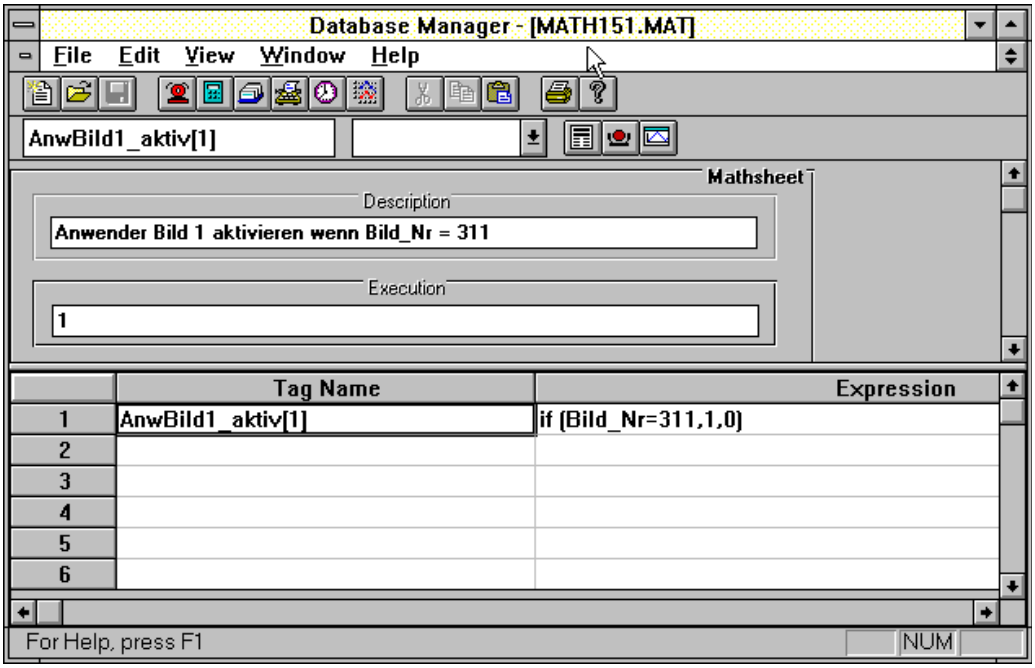


Fig. 6-11 Example of Math Sheet Assignment

The function assignment is handled in the form of a program sequence in a math sheet. By default, all math sheets are available. The first available user math sheet has the number 150.

The **Bild_Nr** variable used in this example comprises a global Bosch variable. The variable returns the value of the current MMI-MADAP screen. The first user screen has the number 311.

☞ **Only in the event that the math sheet contains a value greater than zero in the Execution line, will all math formulae in this configuration sheet be executed.**

Having applied all variables, close the Database Manager, and return to the Application Builder.

Creating graphical objects

Once the variable have been declared, the required objects will be created in the application screen, and the variables will be integrated.

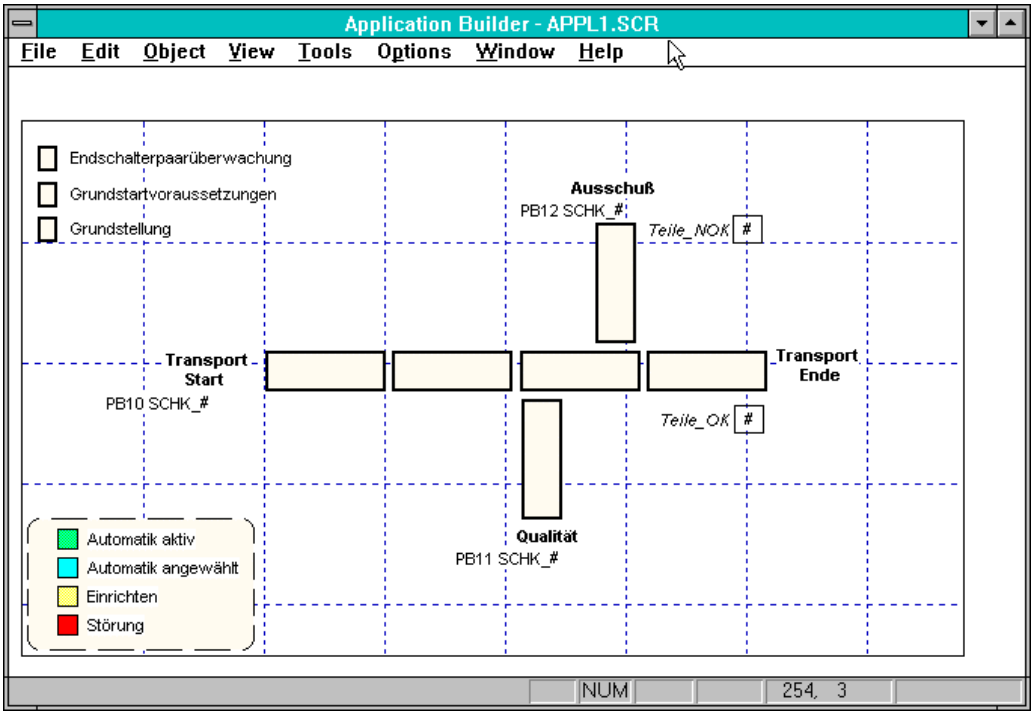


Fig. 6-12 Example of Application in Application Builder

Subsequent to defining all objects, the screen should be tested with the use of the **Execute** (running the application) and **Database Spy** (assigning variables) menu commands.

The procedure is concluded by saving the application, and by terminating the Application Builder.

Defining configuration sheet

In order to read data from the PLC controller, a Read instruction must be executed via the PROFIBUS-FMS. Read and Write instructions are defined via configuration sheets in the UniSoft **Configurator** software module.

In the Windows Program Manager, select the MMI-MADAP directory. Start the MMI-MADAP PROFIBUS Configurator software module by double-clicking the **MMIMADAP Profibus Configuration icon** (shown below).

In the Configurator, select **Open** to open a user configuration sheet, or create a new one.

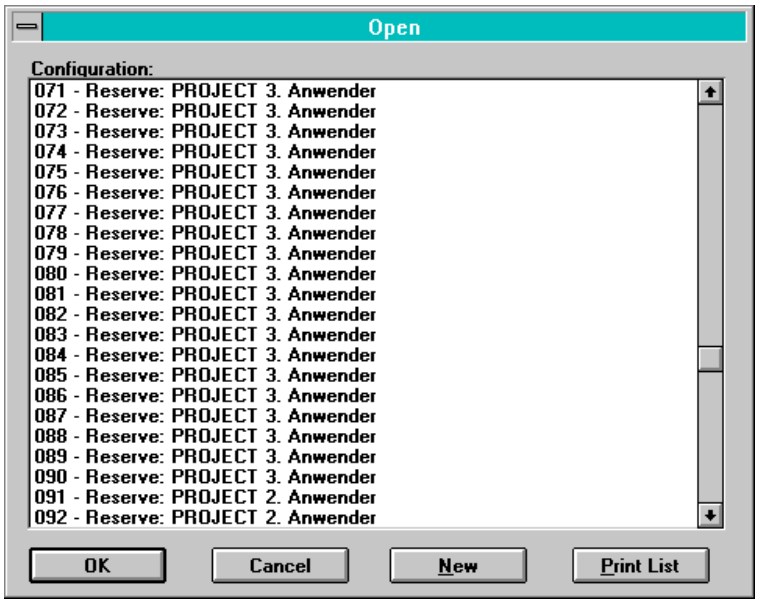


Fig. 6-13 Sample List of User Configuration Sheets

In this example, the **AnwBild1_aktiv[1]** variable is used for the purpose of transfer control.

The data is read, beginning with the first data word in the object bearing index number 30.

The data from the PLC controller is saved to the variables named **Anlage[1]** through **Anlage[14]**.

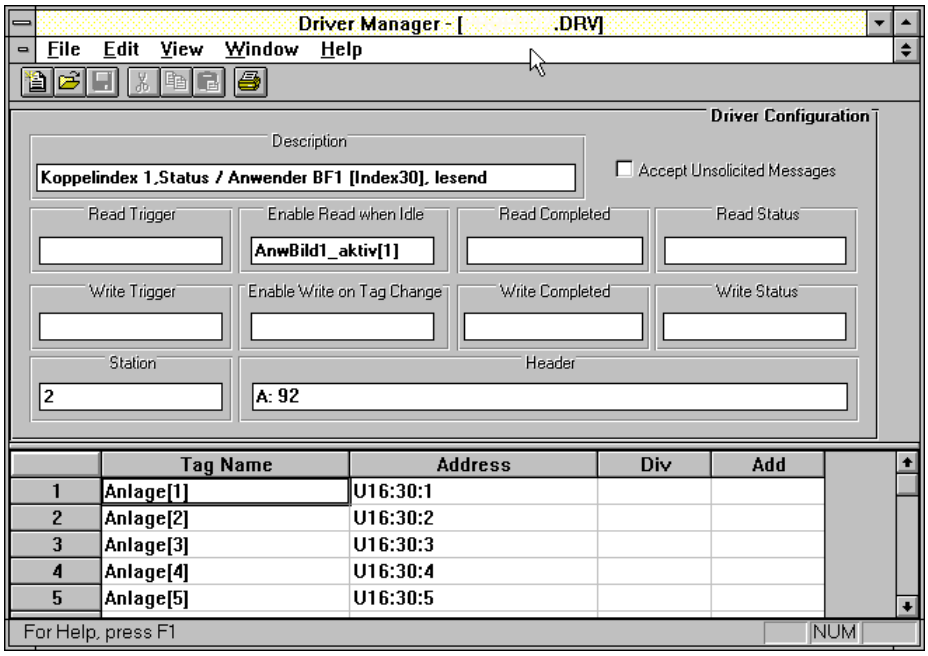


Fig. 6-14 Example of Configuration Sheet in Configurator

Once the configuration sheet has been configured, it is saved, and the application is terminated.

Testing user screen

To test your user screen, you will be required to start the MMI-MADAP software, and open your user screen via screen displays.

In the Windows Program Manager, select the MMI-MADAP directory. Start the MMI-MADAP software module by double-clicking the **MMI-MADAP icon** (shown below).

Continue by selecting the **Display** button to select the desired user screen.

6.8 MMI-MADAP Merge Function

The Merge function facilitates the merging of different applications into a single application..

In this process, four functional units must be considered:

- Application database
- Screens
- Definitions
- Data range definitions (components)

☞ **In order to work with the Merge function, all other MMI-MADAP modules, such as BGTask, Viewer, etc., must be closed.**

6.8.1 Application database

The MMI-MADAP software functions are based upon a single database. This database is named TAGL.TXT, and is located in the C:\MMIMADAP\DATABASE directory.

Each user creates a new application with the Application Manager.

Example:

Start the Application Manager.

Fig. 6-15 Application Manager Icon

Select the **Create a New Application** button.

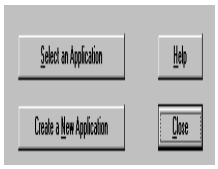


Fig. 6-16 Create New Application Button

Create a new application which you will name "Anwender" (user). This will create a new directory named C:\Anwender.

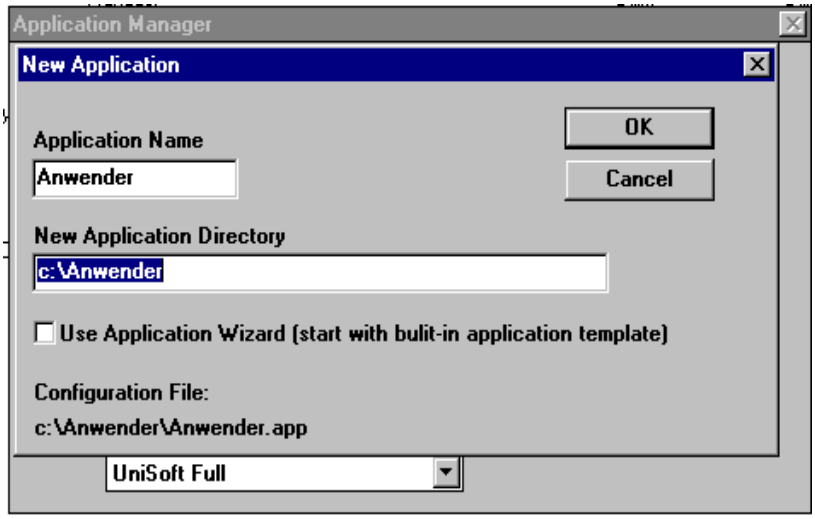


Fig. 6-17 New Application, Dialog Box

In this new application, in the subdirectory named C:\ANWENDER\DATABASE, there is an empty database named TAGL.TXT.

☞ **In an MMI-MADAP database, variables may be defined with a unique name only. In order to prevent the merging of applications containing variables with identical names, each user must assign a unique identifier to his variables at the time he creates his application.**

Example:

User in *Company A*: All names of variables begin with CA_,
e.g. CA_variable1, CA_variable2, etc.

User in *Company B*: All names of variables begin with CB_
e.g. CB_variable1, CB_variable2, etc.

With the variables thus uniquely identified, the automatic generation of a single database by defining the pathnames for the two source databases is possible.

The current project must always be MMI-MADAP (1st database). Select this project with the use of the Project Manager.

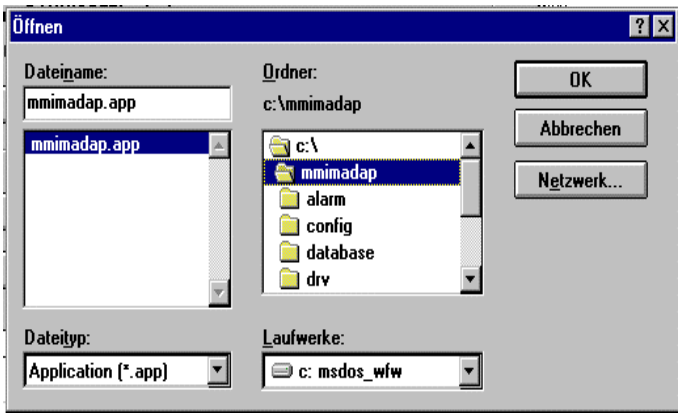


Fig. 6-18 Selecting an Application

Go to the Windows Program Manager, and start the DBMERGE.EXE program.

Fig. 6-19 Database Merge Utility Icon

This takes you to the selection of the 2nd database.

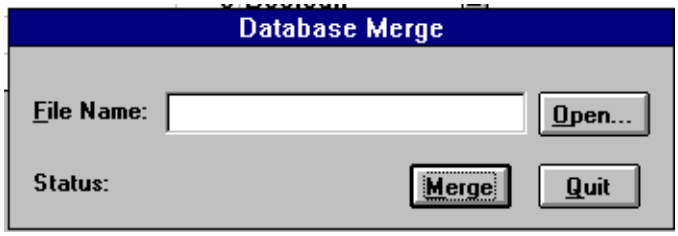


Fig. 6-20 Selecting 2nd Database

In the File Name text box, type the complete pathname and filename. At this point you can click **Open...** for quick access to the database.

Fig. 6-21 Opening TAGL.TXT Database

Selecting **Merge** will cause the contents of the 2nd database to be added to those of the 1st database.

Fig. 6-22 Database Merge, Dialog Box

Example of Merge function:

1st Database: C:\MMIMADAP\DATABASE\TAGL.TXT

2nd Database: C:\ANWENDER\DATABASE\TAGL.TXT

Result: C:\MMIMADAP\DATABASE\TAGL.TXT

Through consistent repetition of this procedure, any number of databases can be merged into a single database.

☞ **Mixing variables of the CLASS type is not possible.**

6.8.2 Screens

In the course of project design with the MMI-MADAP software, the project designer creates his own user screens. These screens must then be added to the default screen repertoire of the MMI-MADAP software.

6.8.2.1 Procedure for creating user screens

- Number of user screens

Up to 64 user screens can be designed with the use of the development editor for the MMI-MADAP software packet. The predefined user screens are located in the directory named:

C:\MMIMADAP\SCREEN

- User screen structure

Each user screen consists of two partial images.

ANW1 background image

This image contains the screen title, the softkey rows, all key functions (screen change, etc.), and is used to call the foreground screens named APPL1 through APPL64.

☞ **The user is unable to change the background image.**

APPL1 through APPL64 foreground images

These images are used to design the functions for the user screens, such as bar graphs, value entry text boxes, overview screens, etc.

Once created, all screens can be copied for further processing into other projects at any time. This means that they will also function in the runtime version of the program.

- Screen creation procedure

In the Windows Program Manager, select UNISOFT.

Open the Application Manager.

Use the **Select an Application** command to select the MMIMADAP.APP file in the C:\MMIMADAP project directory.

Start the Application Builder.

Select and edit user screen foreground images named APPL1 through APPL64.

Enter the required variables in the existing database.

Proceed with the creation of your user screens.

6.8.2.2 Definitions

The MMI-MADAP software facilitates the online creation of new definitions, such as movement text, power-up criteria, etc. The respective definitions re stored in the C:\MMIMADAP directory in the form of *.DAT files.

These files can be modified as desired, and also copied into new projects.

The copying process can be automated by means of batch files. Another option is the use of the **Copy** command in the PROJEKT window of the MMI-MADAP user interface.

6.8.3 Defining Data Ranges

6.8.3.1 Introduction

The MMI-MADAP standard software packet defines the various data ranges:

The term *data ranges* encompasses the following components:

- Alarms (Alarm files)
- Math worksheets (Math files)
- Time-controlled processing (Scheduler files)
- Trend values (Trend files)
- Report outputs (Report files)
- Data management (Recipe files)

☞ **With the exception of the online data, all components are saved as data files. They are located in the directory named C:\MMIMADAP\CONFIG. All files can be freely copied.**

6.8.3.2 Alarm files

Range	Filename, *.ALR	Utilizer
1 - 10	ALARM001 - ALARM010	BOSCH
11-20	ALARM011 - ALARM020	Development
21-30	ALARM021 - ALARM030	Project / 1st user
31-40	ALARM031 - ALARM040	Project / 2nd user
41-50	ALARM041 - ALARM050	Project / 3rd user

6.8.3.3 Math files

Range	Filename, *.MAT	Utilizer
1 - 100	MATH001 - MATH0100	BOSCH
101-150	MATH101 - MATH150	Development
151-200	MATH151 - MATH200	Project / 1st user
201-250	MATH201 - MATH250	Project / 2nd user
251-300	MATH251 - MATH300	Project / 3rd user

6.8.3.4 Scheduler files

Range	Filename, *.SCH	Utilizer
1	SCHED001	BOSCH
2	SCHED002	Development
3	SCHED003	Project / 1st user
4	SCHED004	Project / 2nd user
5	SCHED005	Project / 3rd user

6.8.3.5 Trend files

Range	Filename, *.TRD	Utilizer
1 - 50	TREND001 - TREND050	BOSCH
51-70	TREND051 - TREND070	Development
70-90	TREND071 - TREND090	Project / 1st user
91-110	TREND091 - TREND110	Project / 2nd user
111-130	TREND111 - TREND130	Project / 3rd user

6.8.3.6 Report files

Range	Filename, *.REP	Utilizer
	ABCDEFGH.REP	BOSCH
	ABCDEFGH.REP	Development
	ABCDEFGH.REP	Project / 1st user
	ABCDEFGH.REP	Project / 2nd user
	ABCDEFGH.REP	Project / 3rd user

Each user can assign filenames of his choice. The filename extension will always the .REP. In order to prevent dual filename assignments, the users named DEVELOPMENT and PROJECT must precede the filename by an additional identifier.
For example:

DEVELOPMENT: FA_001, CA_002, etc.

PROJECT: FB_001, FB_002, etc.

6.8.3.7 Recipe files

Range	Filename, *.RCP	Utilizer
	ABCDEFGH.RCP	BOSCH
	ABCDEFGH.RCP	Development
	ABCDEFGH.RCP	Project / 1st user
	ABCDEFGH.RCP	Project / 2nd user
	ABCDEFGH.RCP	Project / 3rd user

Each user can assign filenames of his choice. The filename extension will always the .REP. In order to prevent dual filename assignments, the users named DEVELOPMENT and PROJECT must precede the filename by an additional identifier.
For example:

DEVELOPMENT: FA_001, CA_002, etc.

PROJECT: FB_001, FB_002, etc.

6.8.4 Communications

With regard to standardization, the structure of PROFIBUS communications encompasses two partial areas:

- MMI-MADAP forms
- Communication files

Each operator terminal featuring executable MMI-MADAP software requires communication forms. The forms are provided in file form in the C:\MMIMADAP\CONFIG directory. This makes them accessible to any operator terminal with the same name.

Possible forms are listed below:

- BUEP19E, serial communications
- SFMS (Softing), PROFIBUS-FMS communications
- SL2A2 (Siemens), PROFIBUS-FMS communications

Range	Filename, *DRV	Utilizer
1 - 100	*001 - *050	BOSCH
101-150	*051 - *070	Development
151-200	*071 - *090	Project / 1st user
201-250	*091 - *110	Project / 2nd user
251-300	*111 - *130	Project / 3rd user

6.8.5 Modifying the Application

In normal circumstances the user will create his user screens on a designated development computer. This computer is equipped with the development license (Hardkey dongle).

When combining individual files on other operator terminals, the proper functioning of the complete application (Bosch MMI-MADAP + user screens + user database) requires adherence to specific procedural steps.

These steps are required because, due to considerations of speed, the access to variables in the individual files, such as screens, math sheets, communications drivers, is effected via internal references. these references must be modified for the entire application.

- Merging all user files

  Modifications to the screens and configuration sheets in accordance with MMI-MADAP defaults can be effected on different development computers. A so-called *MERGE* computer is used to merge the files created on different development computers.

- MMI-MADAP development version

  All development computers, including the MERGE computer, must operate the latest MMI-MADAP software version. The same applies to all operator terminals that will have to process user files. At the time of this writing, this is version 1.13.

- Merge procedure

  On the MERGE computer, the MERGE software module is used to blend the databases from other development computers with the database of the MERGE computer. All screens and configuration files must be copied into the respective directories.

☞ **Refer to preceding section.**

- DBVERSIO.BIN

  In the C:\MMIMADAP\DATABASE\DBVERSIO.BIN file, the internal references are stored in the form of a number. This number must be manipulated manually.

  Using ASCII notation, a number is generated in the YDDMM format, where

  Y: 1 = odd year, 2 = even year

  DD: 2 numbers represent today's date

  MM: 2 numbers represent the current month.

Example:

If the merge procedure is completed on 1 January, 1997, the number to be entered in the DBVERSIO.BIN file is 10108.

- Database Pack Utility

  As the next step, the Database Pack Utility is started, followed by selecting the **PACK** command, followed by **OPTIMIZE** (see also description earlier in this chapter).

- Merging the files

  All user files are now compatible for use on the destination computers.

- Installing user files on destination computers

  As a prerequisite for this procedure, the latest version of MMI-MADAP software must be installed on all destination computers. At the time of this writing, this is version 1.13. Copy all user files into the respective destination directories. In the C:\MMIMADAP\CONFIG directory, delete all files with the .MAC filename extension.
  Restart the MMI-MADAP software.

6.9 Screen List and Screen Numbers

All screens feature a screen name and a screen number.

The screens are located in file form in the directory named C:\MMIMADAP\SCREEN, and are identified by the .SCR filename extension.

The screen number is transferred to the PLC where is can be used for the purpose of selecting user-specific functions.

Modifiable screens

The term *modifiable screens* applies to the screen masks that you are using for system-specific visualization.

Screen #	Screen name	Description
311-318	Appl1-Appl8	User screen, Group 1 / Screen 1-8
321-328	Appl9-Appl16	User screen, Group 2 / Screen 1-8
331-338	Appl17-Appl24	User screen, Group 3 / Screen 1-8
341-338	Appl25-Appl32	User screen, Group 4 / Screen 1-8
351-358	Appl33-Appl40	User screen, Group 5 / Screen 1-8
361-368	App41-Appl48	User screen, Group 6 / Screen 1-8
371-378	Appl49-Appl56	User screen, Group 7 / Screen 1-8
381-388	Appl57-Appl64	User screen, Group 8 / Screen 1-8

Permanent screens

Permanent screens are those screen masks that contain the MMI-MADAP standard functions, and that are not accessible to the user.

Screen #	Description
	Base screen and secondary definition screens
1	Base screen with initialization
11	Definition, Softkey for Power-up screens
21	Definition, Softkey for Movement screens
22	Definition, Movement screen matrix
31	Definition, Softkey for User screens
32	Definition, User screen matrix
	Power-up conditions
110	Definition, Power-up conditions
111	Power-up conditions 1
112	Power-up conditions 2
113	Power-up conditions 3
114	Power-up conditions 4
115	Power-up conditions 5
116	Power-up conditions 6

	Movement screens
210	Definition, Operand addresses for Movement screens
210	Definition, Movement text and cascades/steps for movement screens
211-218	Movement screen, Group 1 / Screen 1-8
221-228	Movement screen, Group 2 / Screen 1-8
231-238	Movement screen, Group 3 / Screen 1-8
241-248	Movement screen, Group 4 / Screen 1-8
251-258	Movement screen, Group 5 / Screen 1-8
261-268	Movement screen, Group 6 / Screen 1-8
271-278	Movement screen, Group 7 / Screen 1-8
281-288	Movement screen, Group 8 / Screen 1-8
	User screens
310	Definition, User screen titles
	Status displays
411	Inputs
412	Extended inputs
421	Outputs
422	Extended outputs
431	Markers
432	Special markers
433	System range
441	Times
442	Counters
451	Data field
452	Data buffer
461	Data modules
462	Data module list
481	PLC overview, internal messages
482	I/O assignment of connected PLC
483	System configuration of connected PLC
	Messages
611	Parallel user messages
621	Serial user messages
622	Serial user messages, history
661	Diagnostic record storage
681	PLC data, stored PLC errors
741	Error statistics, First-value errors
	Diagnostics
711	Cascade diagnostics display
780	Definition, synchronization in Automatic mode
781	Cascade overview
	Machine usage
511	Display, Machine usage trend
513	Definition, Machine usage trend
521	Display, current machine data
522	Display, historic machine data
530	Definition, Shift times
531	Display, Shift times
540	Definition, Cycle times
541	Display, Cycle times

6.10 Global Standard Variables

In the user screens, predefined global standard variables can be used.

The following variables are available to the user:

Name	Description
Bild_Nr	Number of current screen
bild	Current base screen
ZSNr	Number of current ZS central processing unit
SPS_Typ	PLC type and ZS number read
LOG0	Logic "0"
LOG1	Logic "1"
K_Anz	Number of cascades
BF_Nr	Number of operator terminal
K_Start	Start address = 1st cascade of station
PG_Frei_Anw	PgUp/PgDn block at "1"
PG_Frei_Bew	PgUp/PgDn block at "1"
AnwenderDef_Bild	Definition screen for the user in user screens
Kettdar_Frei	Enable of cascade overview screen from within diagnostic screen: 1 = enable, 0 = disabled

Read-access only is permitted to the listed variables!

7 PROFIBUS-FMS Interface

7.1 Introduction

The PROFIBUS-FMS employed in this system comprises a high-performance industrial data bus. The purpose of the PROFIBUS-FMS is the facilitation of communications between the MMI-MADAP operator terminals and the PLC controller.

In order to assist the user with the use of the PROFIBUS-FMS, the following PROFIBUS-FMS configuration tasks have been predefined for the MMI-MADAP software:

- Topology definition,
- Creation of communication references,
- Creation of communication objects,
- Definition of bus parameters,

The initialization of communication references and the transfer of communication objects is effected directly from the MMI-MADAP operator terminal.

On the PLC side, no special functions are required for PROFIBUS-FMS communications.

Freely applicable communication objects are provided for the user.

☞ **Additional BOSCH documentation**

Manual: *PROFIBUS for Beginners*	No. 1070 072 140
Manual: PROFIBUS-Konfigurator	No. 1070 072 066
Manual: *R500P Computer Interface Module*	No. 1070 072 138

7.2 Communication References

The transfer of communication objects is accomplished by mans of the READ, WRITE and STATUS PROFIBUS-FMS services. To enable these services to accomplish the data transfer, they require a connection-oriented communication reference.

During the initialization phase of the MMI-MADAP operator terminal, the communication references are established and monitored by the PC-PROFIBUS card.

For each MMI-MADAP operator terminal, a communication reference is established with the R500P PLC PROFIBUS card or COM-P card. Each communication reference is assigned a unique index number on both ends. The default index number on the MMI-MADAP operator terminal end is KR2.

The standardized KBL files are prepared for 8 MMI-MADAP operator terminals.

Communication references

R500P or COM-P PROFIBUS card	MMI-MADAP BT 1	MMI-MADAP BT 2	MMI-MADAP BT 8
2	2		
3		2	
.			
.			
9			2

Fig. 7-1 MMIMADAP, R500P or COM-P Communication References

7.3 Communication Objects

A standardized PROFIBUS object configuration is provided for the MMI-MADAP software. The user will not be required to create any other objects.

From the viewpoint of the user, all PROFIBUS objects comprise data ranges within data modules.

For each MMI-MADAP operator terminal, there are eight objects that are non-terminal specific, and nine that are operator terminal-specific.

The objects are divided into three categories:

- MMI-MADAP objects
- User objects
- Common-use MMI-MADAP / User objects

Data ranges of MMI-MADAP objects are managed by the MMI-MADAP PLC software, and may not be write-accessed by the user.

Data ranges of user objects are available to the user for the purpose of communicating with user screens on the MMI-MADAP operator terminal. They must be managed by the user.

Data ranges of common-use MMI-MADAP / User objects can be used in conjunction with each other. The data ranges are managed by the MMI-MADAP software only dependent upon the screen selected on the MMI-MADAP operator terminal. The user may write-access them only while a user screen is selected on the display of the MMI-MADAP operator terminal.

PROFIBUS configuration in MMI-MADAP

All objects are classified via their index numbers. This means, for example, that the object with index number 20, defined as data range data word D0 through D218, is located in data module DM255.

On the MMI-MADAP operator terminal side, the objects are managed in so-called *configuration sheets*. Each configuration sheet is stored in file form. The identifier for these files is SL2A2xxx.DRV.

The time for object update/refresh action is defined in the configuration sheets.

On the PLC side, the object update process is coordinated, subsequent to EP (end of program) or STOP (PLC Stop). This means that the data is being refreshed after each PLC cycle.

Common-use objects for operator terminals BT 1 - 4:

Index	Local Addr. DM/D/No.By	Description	Coordination
20	255/000/220	Messages, DM list, System time, PLC statuses	EP or STOP
21	253/000/184	I/O assignment; SC table	EP or STOP
22	253/184/184		EP or STOP
23	253/368/144		EP or STOP
24	254/000/220	Machine usage and PROFIBUS-DP diagnostics	EP or STOP
25	254/220/220		EP or STOP
26	250/000/220	Reserved, users for BT 1-4	EP or STOP
27	250/220/220		EP or STOP

Fig. 7-2 PROFIBUS – Common-Use Objects for Operator Terminals 1-4

Number of objects used: 8

The objects identified with indexes 26 and 27 represent user objects. These defined data ranges can be managed in the PLC by the user.

Objects for operator terminal 1:

Index	Local Addr. DM/D/No.By	Description	Coordination
28	234/000/220	Cascade diagnostics	EP or STOP
29	234/220/220		EP or STOP
30	231/000/184	Link index 1, Status / User	EP or STOP
31	231/184/184	Link index 2, Status / User	EP or STOP
32	231/368/184	Link index 3, Status / User	EP or STOP
33	230/000/118	General communication and display data	EP or STOP
34	230/118/202		EP or STOP
35	230/320/192		EP or STOP
36	233/000/220	Reserved user object	EP or STOP

Fig. 7-3 PROFIBUS – Objects for BT1 Operator Terminal

Number of objects used: 9

The objects with indexes 30 through 32 are common-use MMI-MADAP / User objects, and can be write-accessed by the user only if a user screen was selected on the MMI-MADAP operator terminal.

The object with index 36 is a user object. This defined data range can be managed in the PLC by the user.

Objects for operator terminal 2:

Index	Local Addr. DB/D/No.By	Description	Coordination
37	239/000/220	Cascade diagnostics	EP or STOP
38	239/220/220		EP or STOP
39	236/000/184	Link index 1, Status / User	EP or STOP
40	236/184/184	Link index 2, Status / User	EP or STOP
41	236/368/184	Link index 3, Status / User	EP or STOP
42	235/000/118	General communication and display data	EP or STOP
43	235/118/202		EP or STOP
44	235/320/192		EP or STOP
45	238/000/220	Reserved user object	EP or STOP

Fig. 7-4 PROFIBUS – Objects for BT2 Operator Terminal

Number of objects used: 9

The objects with indexes 39 through 41 represent common-use MMI-MADAP / User objects, and can be write-accessed by the user only if a user screen was selected on the MMI-MADAP operator terminal.

The object with index 45 is a user object. This defined data range can be managed in the PLC by the user.

Objects for operator terminal 3:

Index	Local Addr. DM/D/No.By	Description	Coordination
46	244/000/220	Cascade diagnostics	EP or STOP
47	244/220/220		EP or STOP
48	241/000/184	Link index 1, Status / User	EP or STOP
49	241/184/184	Link index 2, Status / User	EP or STOP
50	241/368/184	Link index 3, Status / User	EP or STOP
51	240/000/118	General communication and display data	EP or STOP
52	240/118/202		EP or STOP
53	240/320/192		EP or STOP
54	243/000/220	Reserved user object	EP or STOP

Fig. 7-5 PROFIBUS – Objects for BT3 Operator Terminal

Number of objects used: 9

The objects with indexes 48 through 50 represent common-use MMI-MADAP / User objects, and can be write-accessed by the user only if a user screen was selected on the MMI-MADAP operator terminal.

The object with index 54 is a user object. This defined data range can be managed in the PLC by the user.

Objects for operator terminal 4:

Index	Local Addr. DM/D/No.By	Description	Coordination
55	249/000/220	Cascade diagnostics	EP or STOP
56	249/220/220		EP or STOP
57	246/000/184	Link index 1, Status / User	EP or STOP
58	246/184/184	Link index 2, Status / User	EP or STOP
59	246/368/184	Link index 3, Status / User	EP or STOP
60	245/000/118	General communication and display data	EP or STOP
61	245/118/202		EP or STOP
62	245/320/192		EP or STOP
63	248/000/220	Reserved user object	EP or STOP

Fig. 7-6 PROFIBUS – Objects for BT4 Operator Terminal

Number of objects used: 9

The objects with indexes 57 through 59 represent common-use MMI-MADAP / User objects, and can be write-accessed by the user only if a user screen was selected on the MMI-MADAP operator terminal.

The object with index 63 is a user object. This defined data range can be managed in the PLC by the user.

Objects for extended diagnostics (effective only in conjunction with Com-P PROFIBUS card):

Index	Local Addr. DM/D/AnzBy	Description	Coordination
64	222/000/143	Cascade diagnostics for station 5 - 8	EP or STOP
65	223/000/143		EP or STOP
66	224/000/143		EP or STOP
67	225/000/143		EP or STOP

Fig. 7-7 PROFIBUS – Objects for Extended Diagnostics

Table of objects required on R500P or COM-P (with extended diagnostics):

	No. of ZS	Number of objects Basic unit	per terminal	1 BT	2 BT	3 BT	4 BT
R500P	1	8	9	17	26	35	44
	2	8	9	34	52	70	88
	3	8	9	51	78	105	132
	4	8	9	68	104	140	176
Com-P	4	12	9				192

Fig. 7-8 PROFIBUS – Number of Objects for R500P or COM-P

☞ **The total number of objects to be managed is limited to 100 with the use of the PLC R500P PROFIBUS card, and to 200 with the use of the Com-P.**

Useful and practical combinations per each R500P PLC PROFIBUS card

	ZS0	ZS1	ZS2	ZS3	Object total
Number of MMI	3	3	2	0	96
Number of MMI	4	3	1	0	96
Number of MMI	4	4	0	0	88

Fig. 7-9 Practical MMI Combinations per ZS Central Processing Unit

7.4 Manipulating User Objects

User objects represent defined data objects within the PLC. The data ranges can be read and write-accessed be the MMI-MADAP operator terminal.

For the purpose of transferring user data between PLC and MMI-MADAP operator terminal, a transfer control mechanism must be programmed on the MMI-MADAP terminal side. This is accomplished with the use of defined variables.

On the MMI-MADAP operator terminal side, a differentiation is made between the following:

- Control variables (transmission control) and
- FMS variable (transmission data).

The user can integrate the variables in a user screen. This provides to option to effect both data control and data management in a single user screen.

All user objects are preconfigured in existing configuration sheets.

The generation of variables is accomplished by means of the UNISOFT **Application Builder** module.

The definition of control and extension of configuration sheets is handled with the assistance of the UNISOFT **Configurator** module.

☞ **Both the Application Builder and the Configurator are standard components of the development software.**

☞ **Documentation reference:**

Manual: *Development Module*, Chapter "Application Builder"
Manual: *Development Module*, Chapter "Driver Configuration"

Assigning variables

In the Windows Program Manager, select the MMI-MADAP directory. Start the Configurator software by double-clicking the **MMIMADAP Configuration icon** (shown below).

In the Configurator, select the **OPEN** menu command.

Select a communication object from the list display of configuration sheets, and select **OK**.

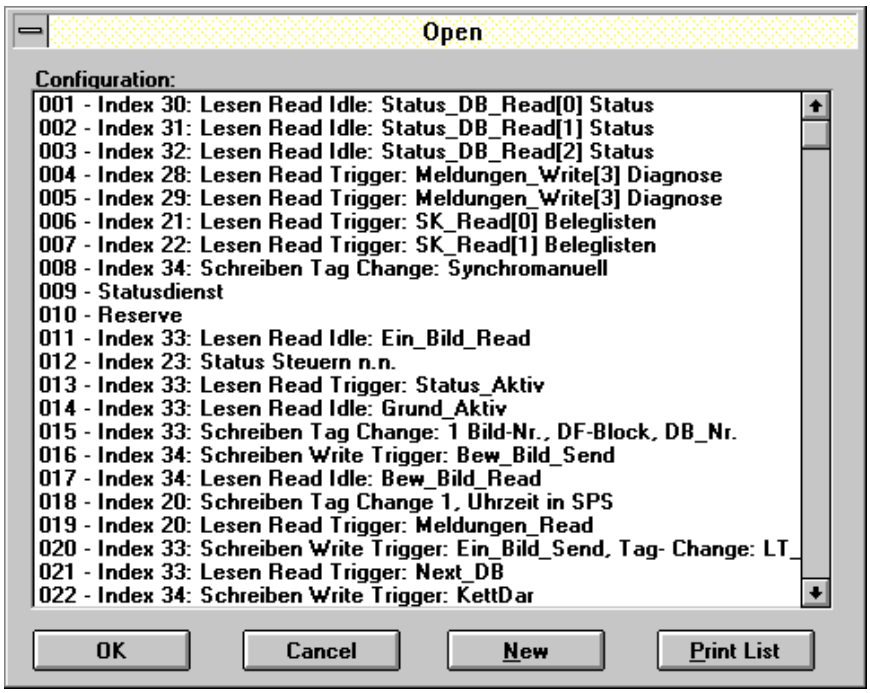


Fig. 7-10 Sample List of Current Communication Sheets

This opens the selected configuration sheet in which you can define the desired variables.

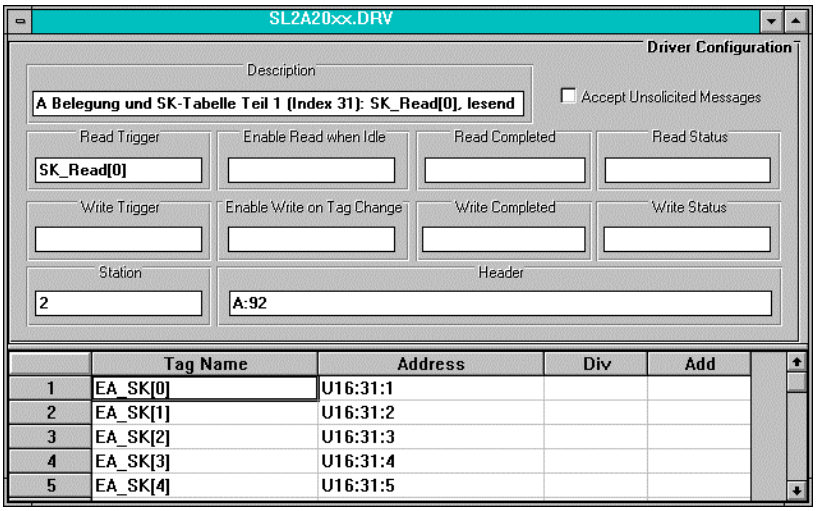


Fig. 7-11 PROFIBUS-FMS – Example of Configuration Sheet

Variables in display:

- SK_READ[0] Control variable
- FMS variable, type Unsigned 16, object index 31; EA_SK[0], data word 1 through EA_SK[91], data word 92

Proceed by assigning the defined control and FMS variables to the configuration sheet.

Once you have concluded your entries, save and close the configuration sheet, and terminate the Configurator utility program by selecting the **Close** menu command.

Configuration sheet description

- **Read Trigger**

This variable used to trigger a Read cycle. Each time this variable changes its value, a cycle is executed, and the listed variables are updated.

- **Enable Read When Idle**

If the variable entered here is larger than 0, a continuous Read cycle is executed while the driver is idle.

- **Read Complete**

The variable in this field is incremented when the Read cycle is concluded.

- **Read Status**

The variable contained in this field receives an error code from the cycle.

- **Write Trigger**

This variable triggers a Write cycle. Each time this variable changes its value, a cycle is executed, and the listed variables are transferred to the PLC.

- **Enable Write on Variable Change**

If the value of the variable entered here exceeds 0, the driver module keeps checking whether a variable in the specified list has changed its value. If this is the case, a Write cycle is triggered, and all changed variables are transferred to the PLC.

- **Write Complete**

The variable in this field is incremented when the Write cycle is concluded.

- **Write Status**

The variable contained in this field receives an error code from the cycle.

8 BUEP19E Interface

8.1 Introduction

The Bosch BUEP19E transmission protocol employed in this system establishes a point-to-point connection between the PLC controller and the MMI-MADAP operator terminal.

☞ **Additional BOSCH documentation**

Manual: *R500 Computer Interface Module*	No. 1070 072 131

8.2 Communication Objects

For use of the BUEP19E protocol in conjunction with the MMI-MADAP software, a standardized object configuration is provided.

From the viewpoint of the user, all BUEP19E objects comprise data ranges within data modules.

For each MMI-MADAP operator terminal, there are eight objects that are non-terminal specific, and nine that are operator terminal-specific.

The objects are divided into three categories:

- MMI-MADAP objects
- User objects (user-defined)
- Common-use MMI-MADAP / User objects

Data ranges of MMI-MADAP objects are managed by the MMI-MADAP PLC software, and may not be write-accessed by the user.

Data ranges of user objects are available to the user for the purpose of communicating with user screens on the MMI-MADAP operator terminal. They must be managed by the user.

Data ranges of common-use MMI-MADAP / User objects can be used in conjunction with each other. The data ranges are managed by the MMI-MADAP software only dependent upon the screen selected on the MMI-MADAP operator terminal. The user may write-access them only while a user screen is selected on the display of the MMI-MADAP operator terminal.

BUEP19E configuration in MMI-MADAP

All objects are defined as data ranges contained within data modules.

On the MMI-MADAP operator terminal side, the objects are managed in so-called *configuration sheets*. Each configuration sheet is stored in file form. The identifier for these files is BUEPxxx.DRV.

The time for object update/refresh action is defined in the configuration sheets.

On the PLC side, the object update process is coordinated, subsequent to EP (end of program) or STOP (PLC Stop). This means that the data is being refreshed after each PLC cycle.

Common-use objects for operator terminals 1 - 4:

Local Addr. DM/D/No.By	Description	Coordination
255/000/220	Messages, DM list, System time, PLC statuses	EP or STOP
253/000/184	I/O assignment; SC table	EP or STOP
253/184/184		EP or STOP
253/368/144		EP or STOP
254/000/220	Machine usage and diagnostics	EP or STOP
254/220/220		EP or STOP
250/000/220	Reserved, users for BT 1-4	EP or STOP
250/220/220		EP or STOP

Fig. 8-1 BUEP19E – Common-Use Objects for Operator Terminals 1-4

Number of objects used: 8

The objects in data module 250 represent user objects. These defined data ranges can be managed in the PLC by the user.

Objects for operator terminal 1:

Local Addr. DM/D/No.By	Description	Coordination
234/000/220	Cascade diagnostics	EP or STOP
234/220/220		EP or STOP
231/000/184	Link index 1, Status / User	EP or STOP
231/184/184	Link index 2, Status / User	EP or STOP
231/368/184	Link index 3, Status / User	EP or STOP
230/000/118	General communication and display data	EP or STOP
230/118/202		EP or STOP
230/320/192		EP or STOP
233/000/220	Reserved user object	EP or STOP

Fig. 8-2 BUEP19E – Objects for BT1 Operator Terminal

Number of objects used: 9

The objects in data module 230 represent common-use MMI-MADAP / User objects, and can be write-accessed by the user only if a user screen was selected on the MMI-MADAP operator terminal.

The object in data module 230 is a user object. This defined data range can be managed in the PLC by the user.

Objects for operator terminal 2:

Local Addr. DM/D/No.By	Description	Coordination
239/000/220	Cascade diagnostics	EP or STOP
239/220/220		EP or STOP
236/000/184	Link index 1, Status / User	EP or STOP
236/184/184	Link index 2, Status / User	EP or STOP
236/368/184	Link index 3, Status / User	EP or STOP
235/000/118	General communication and display data	EP or STOP
235/118/202		EP or STOP
235/320/192		EP or STOP
238/000/220	Reserved user object	EP or STOP

Fig. 8-3 BUEP19E – Objects for BT2 Operator Terminal

Number of objects used: 9

The objects in data module 236 represent common-use MMI-MADAP / User objects, and can be write-accessed by the user only if a user screen was selected on the MMI-MADAP operator terminal.

The object in data module 238 is a user object. This defined data range can be managed in the PLC by the user.

Objects for operator terminal 3:

Local Addr. DM/D/No.By	Description	Coordination
244/000/220	Cascade diagnostics	EP or STOP
244/220/220		EP or STOP
241/000/184	Link index 1, Status / User	EP or STOP
241/184/184	Link index 2, Status / User	EP or STOP
241/368/184	Link index 3, Status / User	EP or STOP
240/000/118	General communication and display data	EP or STOP
240/118/202		EP or STOP
240/320/192		EP or STOP
243/000/220	Reserved user object	EP or STOP

Fig. 8-4 BUEP19E – Objects for BT3 Operator Terminal

Number of objects used: 9

The objects in data module 241 represent common-use MMI-MADAP / User objects, and can be write-accessed by the user only if a user screen was selected on the MMI-MADAP operator terminal.

The object in data module 234 is a user object. This defined data range can be managed in the PLC by the user.

Objects for operator terminal 4:

Local Addr. DM/D/No.By	Description	Coordination
249/000/220	Cascade diagnostics	EP or STOP
249/220/220		EP or STOP
246/000/184	Link index 1, Status / User	EP or STOP
246/184/184	Link index 2, Status / User	EP or STOP
246/368/184	Link index 3, Status / User	EP or STOP
245/000/118	General communication and display data	EP or STOP
245/118/202		EP or STOP
245/320/192		EP or STOP
248/000/220	Reserved user object	EP or STOP

Fig. 8-5 BUEP19E – Objects for BT4 Operator Terminal

Number of objects used: 9

The objects in data module 246 represent common-use MMI-MADAP / User objects, and can be write-accessed by the user only if a user screen was selected on the MMI-MADAP operator terminal.

The object in data module 248 is a user object. This defined data range can be managed in the PLC by the user.

Objects for extended diagnostics:

Local Addr. DM/D/No.By	Description	Coordination
222/000/143	Cascade diagnostics for station 5 - 8	EP or STOP
223/000/143		EP or STOP
224/000/143		EP or STOP
225/000/143		EP or STOP

Fig. 8-6 BUEP19E – Objects for Extended Diagnostics

8.3 Manipulating User Objects

User objects represent defined data objects within the PLC. The data ranges can be read and write-accessed be the MMI-MADAP operator terminal.

For the purpose of transferring user data between PLC and MMI-MADAP operator terminal, a transfer control mechanism must be programmed on the MMI-MADAP terminal side. This is accomplished with the use of defined variables.

On the MMI-MADAP operator terminal side, a differentiation is made between the following:

- Control variables (transmission control) and
- BUEP19E variable (transmission data).

The user can integrate the variables in a user screen. This provides to option to effect both data control and data management in a single user screen.

All user objects are preconfigured in existing configuration sheets.

The generation of variables is accomplished by means of the UNISOFT **Application Builder** module.

The definition of control and extension of configuration sheets is handled with the assistance of the UNISOFT **Configurator** module.

☞ **Both the Application Builder and the Configurator are standard components of the development software.**

☞ **Documentation reference:**

Manual: Development Module, Chapter "Application Builder"
Manual: Development Module, Chapter "Driver Configuration"

Assigning variables

In the Windows Program Manager, select the MMI-MADAP directory. Start the Configurator software by double-clicking the **MMIMADAP Configuration icon** (shown below).

In the Configurator, select the **OPEN** menu command.

Select a communication object from the list display of configuration sheets, and select **OK**.

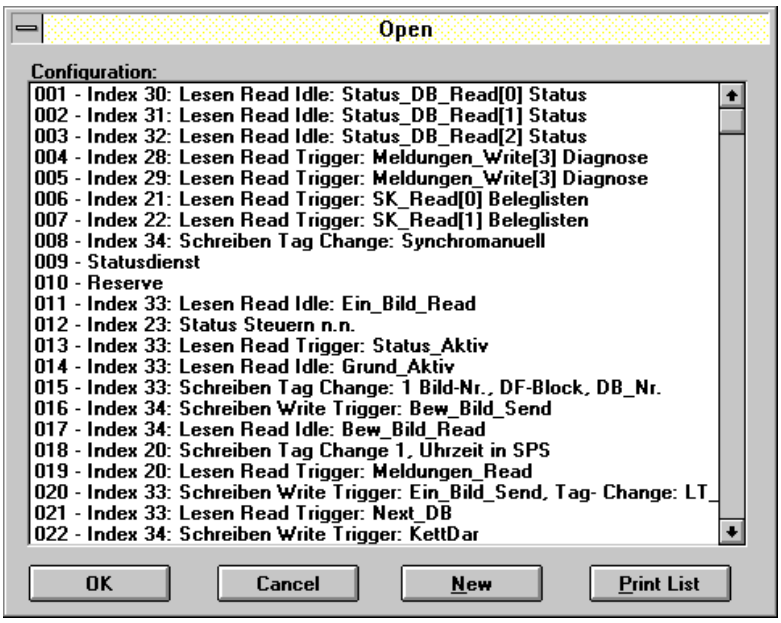


Fig. 8-7 Example of List of Current Communication Sheets

This opens the selected configuration sheet in which you can define the desired variables.

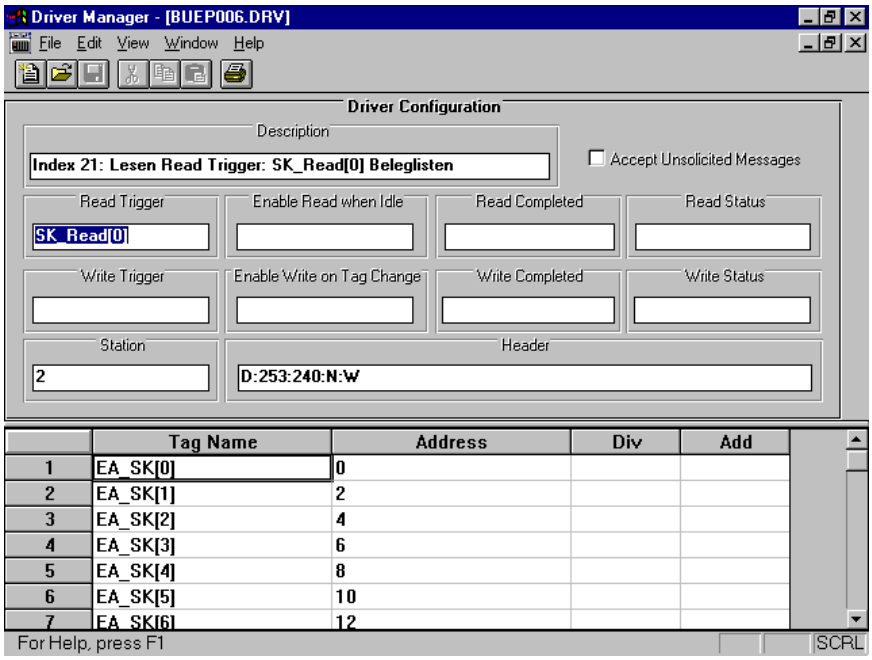


Fig. 8-8 BUEP19E – Example of Configuration Sheet

Variables in display:

- SK_READ[0] Control variable
- BUEP19E variable, typ EA_SK[0], data word 1 through EA_SK[91], data word 92 of data module 253

Proceed by assigning the defined control and BUEP19E variables to the configuration sheet.

Once you have concluded your entries, save and close the configuration sheet, and terminate the Configurator utility program by selecting the **Close** menu command.

Configuration sheet description

- **Read Trigger**

This variable used to trigger a Read cycle. Each time this variable changes its value, a cycle is executed, and the listed variables are updated.

- **Enable Read When Idle**

If the variable entered here is larger than 0, a continuous Read cycle is executed while the driver is idle.

- **Read Complete**

The variable in this field is incremented when the Read cycle is concluded.

- **Read Status**

The variable contained in this field receives an error code from the cycle.

- **Write Trigger**

This variable triggers a Write cycle. Each time this variable changes its value, a cycle is executed, and the listed variables are transferred to the PLC.

- **Enable Write on Variable Change**

If the value of the variable entered here exceeds 0, the driver module keeps checking whether a variable in the specified list has changed its value. If this is the case, a Write cycle is triggered, and all changed variables are transferred to the PLC.

- **Write Complete**

The variable in this field is incremented when the Write cycle is concluded.

- **Write Status**

The variable contained in this field receives an error code from the cycle.

9 Appendix

9.1 Index

1070 072 168-101 (98.04) GB • HB SP • AT/PLS • Printed in Germany