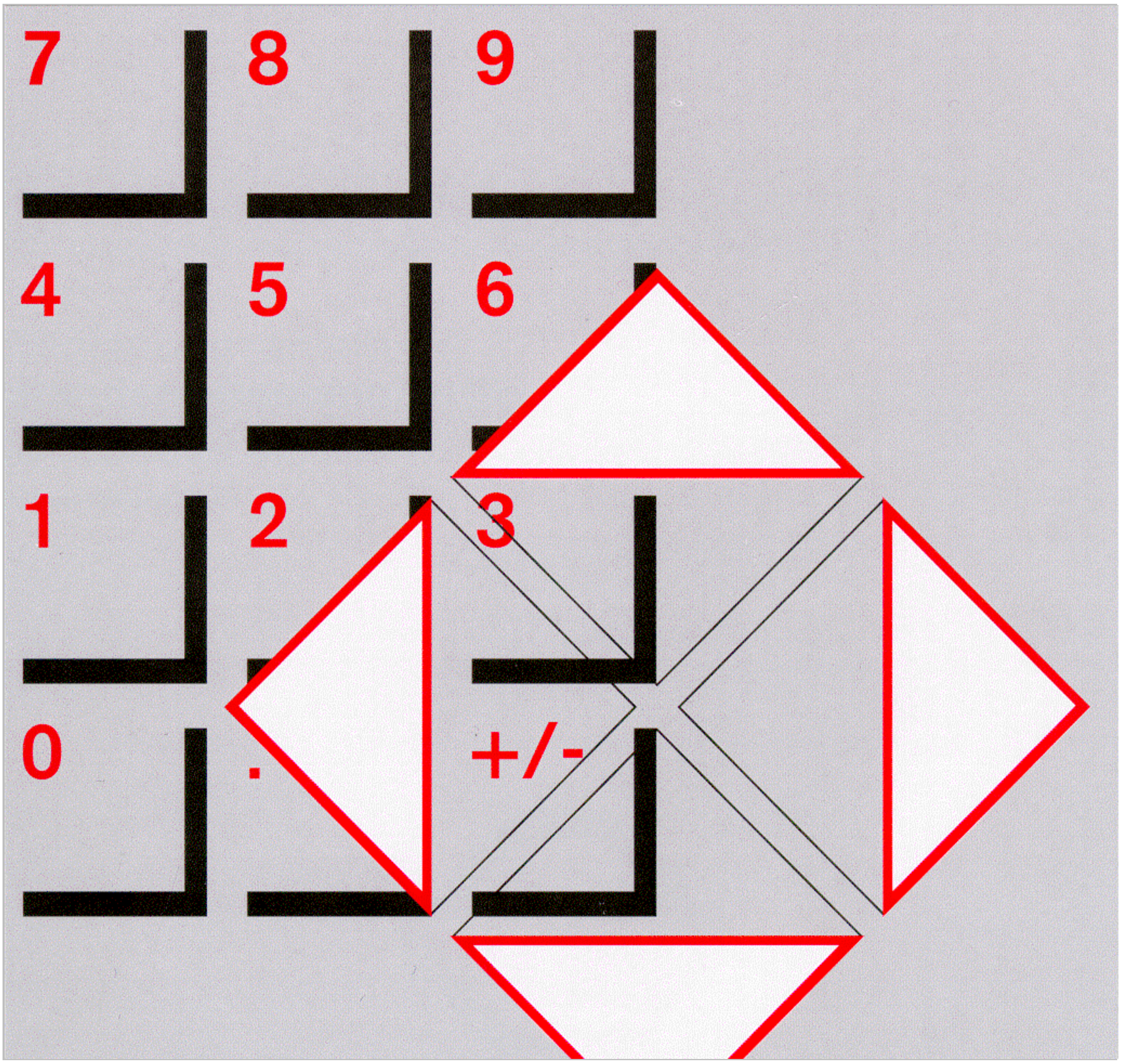
Antriebs- und Steuerungstechnik

PNC

PLC Project Planning



Edition 112

Rexroth
Bosch Group

PNC

PLC Project Planning

1070 073 741-112 (02.10) GB

Software release: V7.3

Discretionary charge EUR 18.–

Contents

Page

1 Safety Instructions 1–1
1.1 Intended use 1–1
1.2 Qualified personnel 1–2
1.3 Safety markings on products 1–3
1.4 Safety instructions in this manual 1–4
1.5 Safety instructions for the described product 1–5
1.6 Documentation, software release and trademarks 1–7

2 Communication structure and interfaces 2–1
2.1 Overview 2–1
2.2 The NCS (NC interface) 2–1
2.3 The APS (APplication software) 2–4

3 PLC<–>NC communication via APS 3–1

4 General interface signals 4–1
4.1 Overview of general interface signals 4–1
4.1.1 Overview of input signals NC-I (PLC –> NC) 4–1
4.1.2 Overview of output signals NC-O (NC –> PLC) 4–1
4.2 Signal description 4–2
4.2.1 Input signals NC-I 4–2
4.2.2 Output signals NC-O 4–6

5 Channel-related interface signals 5–1
5.1 Overview of channel-related interface signals 5–1
5.1.1 Overview of input signals NC-I (PLC –> NC) 5–1
5.1.2 Overview of output signals NC-O (NC –> PLC) 5–3
5.2 Signal description 5–4
5.2.1 Input signals NC-I 5–4
5.2.2 Output signals NC-O 5–15

6 Axis-related interface signals 6–1
6.1 Overview of axis-related interface signals 6–1
6.1.1 Overview of input signals NC-I (PLC –> NC) 6–1
6.1.2 Overview of output signals NC-O (NC –> PLC) 6–3
6.2 Signal description 6–5
6.2.1 Input signals NC-I 6–5
6.2.2 Output signals NC-O 6–17

7 Spindle-related interface signals 7–1
7.1 Overview of spindle-related interface signals 7–1
7.1.1 Overview of input signals NC-I (PLC –> NC) 7–1
7.1.2 Overview of output signals NC-O (NC –> PLC) 7–3
7.2 Signal description 7–5
7.2.1 Input signals NC-I 7–5
7.2.2 Output signals NC-O 7–15

8 External spindle speed 8–1

9 Auxiliary functions 9–1
9.1 Bit-coded auxiliary functions 9–1
9.2 BCD-coded channel-independent auxiliary functions 9–3

10 PROFIBUS-DP (DP master) interface 10–1
10.1 Commissioning of PROFIBUS-DP interface 10–1
10.2 Determining the DP master status 10–3
10.3 Digital and analog osa dc interface I/Os (PNC-R with SNCI4) . 10–4

11 Machine error and status display (MSD) 11–1
11.1 MSD files and displays 11–1
11.2 Activating messages 11–3

12 Program modules for iPCL and ICL700 12–1
12.1 General structure of program modules 12–1
12.2 Read actual axis values (-B02AXPOS) program module 12–6
12.2.1 Module call 12–6
12.2.2 Configuration example 12–8
12.2.3 Status messages 12–9
12.3 External program selection/deselection (-B03PANAB) program module 12–10
12.3.1 Module call 12–10
12.3.2 Configuration example 12–13
12.3.3 Status messages 12–14
12.4 NC block input (-B04SATZV) program module 12–15
12.4.1 Module call 12–15
12.4.2 Configuration example 12–17
12.4.3 Status messages 12–18
12.5 Temperature compensation (-B05TKOMP) program module . 12–19
12.5.1 Module call 12–19
12.5.2 Configuration example 12–21
12.5.3 Status messages 12–22
12.6 Logic analyzer (-B06LGANA) program module 12–23
12.6.1 Module call 12–23
12.7 Read MACODA parameters (-B07MPARA) program module . . 12–24
12.7.1 Module call 12–24
12.7.2 Configuration example 12–26
12.7.3 Status messages 12–27
12.8 Open/close serial interface (-B10SEROP) program module . . 12–28
12.8.1 Module call 12–28
12.8.2 Configuration example 12–32
12.8.3 Status messages 12–34
12.9 Data read-in/read-out (-B11SERIO) program module 12–35
12.9.1 Module call 12–35
12.9.2 Configuration example 12–38
12.9.3 Status messages 12–39

12.10 Program module handwheel data (-B14HANDW) 12–40
12.10.1 Module call 12–40
12.10.2 Configuration example 12–42
12.10.3 Status messages 12–42
12.11 Read/Input SERCOS ident. number (-B15IDTNR) program module 12–43
12.11.1 Module call 12–43
12.11.2 Configuration example 12–46
12.11.3 Status messages 12–48
12.12 Edit tool list (-B20WZLST) program module 12–49
12.12.1 Module call 12–49
12.12.2 Configuration example 12–52
12.12.3 Status messages 12–54
12.13 Edit tool data (-B21WZDAT) program module 12–55
12.13.1 Module call 12–55
12.13.2 Configuration example 12–58
12.13.3 Status messages 12–59
12.14 External tool compensation (-B22WZKOR) program module . 12–60
12.14.1 Module call 12–61
12.14.2 Configuration example 12–64
12.14.3 Status messages 12–65
12.15 External zero shift (-B23XTNPV) program module 12–66
12.15.1 Module call 12–66
12.15.2 Configuration example 12–69
12.15.3 Status messages 12–71
12.16 Communication with FIFO modules (-B55FIFOC) program module 12–72
12.16.1 Module call 12–72
12.16.2 Status messages 12–75
12.17 Conversion Integer –> ASCII (-B97ITOA) program module . . 12–77
12.17.1 Module call 12–77
12.18 Read system date/time (-B98DATTM) program module 12–78
12.18.1 Module call 12–78

13 Machine operating panel 13–1
13.1 Machine operating panel with PROFIBUS-DP 13–1
13.1.1 Signal transmission 13–2
13.2 Machine operating panel with CAN bus 13–5
13.2.1 Activating the machine operating panel 13–6
13.2.2 Signal transmission 13–9

A Appendix A–1
A.1 Overview of interface signals A–1
A.1.A General interface signals, PLC –> NC A–1
A.1.B General interface signals, NC –> PLC A–1
A.1.C Channel-related interface signals, PLC –> NC A–2
A.1.D Channel-related interface signals, NC –> PLC A–4
A.1.E Axis-related interface signals, PLC –> NC A–5
A.1.F Axis-related interface signals, NC –> PLC A–6
A.1.G Spindle-related interface signals, PLC –> NC A–8
A.1.H Spindle-related interface signals, NC –> PLC A–9
A.2 Overview of version dependency of the APS modules for ICL 700 A–11
A.3 Transferring an existing ICL project into an iPCL project A–12
A.4 Index A–14

1 Safety Instructions

Please read this manual before commissioning the integrated PLC. Store this documentation in a place to which all users have access at all times.

1.1 Intended use

This manual contains information on the proper use of the control unit. However, for reasons of clarity, it cannot contain every detail of all function combinations possible. Likewise, it is impossible to consider every conceivable case of integration or operation.

The numerical controls PNC-R and PNC-P are used to

- activate feed drives, spindles and auxiliary axes of a machine tool via SERCOS interface for the purpose of guiding a processing tool along a programmed path to machine a workpiece (CNC). Furthermore, a PLC is required with appropriate I/O components which – in communication with the actual CNC – controls the machine processing cycles holistically and acts as a technical safety monitor.
- program contours and the machining technology (path feedrate, spindle speed, tool change) of a workpiece.

Any other application is deemed improper use!

The products described

- have been developed, manufactured, tested and documented in compliance with the safety standards. These products pose no danger to persons or property if they are used in accordance with the handling regulations and safety notes prescribed for their configuration, installation and proper operation.
- comply with the requirements of
  - the EMC Directives (89/336/EEC, 93/68/EEC and 93/44/EEC)
  - the Low-Voltage Directive (73/23/EEC)
  - the harmonized standards EN 50081-2 and EN 50082-2
- are designed for operation in industrial environments, i.e.
  - no direct connection to public low-voltage power supply,
  - connection to the medium- or high-voltage system via a transformer.

  In residential environments, in trade and commerce as well as small enterprises class A equipment may only be used if the following warning is attached:

☞ **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.**

The faultless, safe functioning of the product presupposes proper transport, storage, erection and installation as well as careful operation.

1.2 Qualified personnel

The requirements as to qualified personnel depend on the qualification profiles described by ZVEI (central association of the electrical industry) and VDMA (association of German machine and plant builders) in:
Weiterbildung in der Automatisierungstechnik
edited by: ZVEI and VDMA
MaschinenbauVerlag
Postfach 71 08 64
D-60498 Frankfurt.

The present manual is designed for **project engineers and NC specialists**. These persons need special knowledge of the configuration and commissioning of electrical equipment.

Programming, start and operation as well as the modification of program parameters is reserved to properly trained personnel! This personnel must be able to judge potential hazards arising from programming, program changes and in general from the mechanical, electrical, or electronic equipment.

Interventions in the hardware and software of our products, unless described otherwise in this manual, are reserved to specialized Rexroth personnel.

Tampering with the hardware or software, ignoring warning signs attached to the components, or non-compliance with the warning notes given in this manual may result in serious bodily injury or material damage.

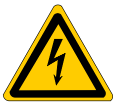
Only electrotechnicians as recognized under IEV 826-09-01 (modified) who are familiar with the contents of this manual may install and service the products described.

Such personnel are

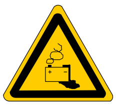
- those who, being well trained and experienced in their field and familiar with the relevant norms, are able to analyze the jobs being carried out and recognize any hazards which may have arisen.
- those who have acquired the same amount of expert knowledge through years of experience that would normally be acquired through formal technical training.

With regard to the foregoing, please note our comprehensive range of training courses. Please visit our website at http://www.boschrexroth.de for the latest information concerning training courses, teachware and training systems. Personal information is available from our Didactic Center Erbach, Telephone: (+49) (0) 60 62 78-600.

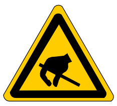
1.3 Safety markings on products



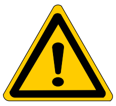
Warning of dangerous electrical voltage!



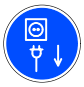
Warning of danger caused by batteries!



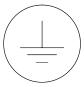
Components sensitive to electrostatic discharge!



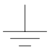
Warning of hazardous light emissions
(optical fibre cable emitters)!



Disconnect mains power before opening!

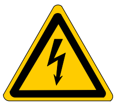


Lug for connecting PE conductor only!

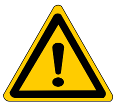


Connection of shield conductor only

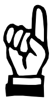
1.4 Safety instructions in this manual



DANGEROUS ELECTRICAL VOLTAGE
This symbol is used to warn of a **dangerous electrical voltage.** The failure to observe the instructions in this manual in whole or in part may result in **personal injury**.



DANGER
This symbol is used wherever insufficient or lacking compliance with instructions may result in **personal injury**.



CAUTION
This symbol is used wherever insufficient or lacking compliance with instructions may result in **damage to equipment or data files**.

☞ This symbol is used to draw the user’s attention to special circumstances.

★ This symbol is used if user activities are required.

1.5 Safety instructions for the described product

DANGER
Danger of life through inadequate EMERGENCY-STOP devices!
EMERGENCY-STOP devices must be active and within reach in all system modes. Releasing an EMERGENCY-STOP device must not result in an uncontrolled restart of the system!
First check the EMERGENCY-STOP circuit, then switch the system on!

DANGER
Risk of personal injury and equipment damage!
Always subject new programmes to initial tests while inhibiting axis movements. For this purpose, as a function of the AUTOMATIC mode, the controller provides the option to block axis movements or auxiliary functions by means of special softkey commands.

DANGER
Incorrect or undesired control unit response!
Rexroth accepts no liability for damage resulting from the execution of an NC program, an individual NC block or the manual movement of axes!

Furthermore, Rexroth accepts no liability for consequential damage which could have been avoided by programming the PLC appropriately!

DANGER
Retrofits or modifications may adversely affect the safety of the products described!
The consequences may include severe injury, damage to equipment, or environmental hazards. Possible retrofits or modifications to the system using third-party equipment therefore have to be approved by Rexroth.

DANGEROUS ELECTRICAL VOLTAGE
Unless described otherwise, maintenance works must be performed on inactive systems! The system must be protected against unauthorized or accidental reclosing.

Measuring or test activities on the live system are reserved to qualified electrical personnel!

DANGER
Tool or axis movements!
Feed and spindle motors generate very powerful mechanical forces and can accelerate very quickly due to their high dynamics.

- **Always stay outside the danger area of an active machine tool!**
- **Never deactivate safety-relevant functions!**
- **Report any malfunction of the unit to your servicing and repairs department immediately!**

CAUTION
Use only spare parts approved by Rexroth!

CAUTION
Observe all precautions for ESD protection when handling modules and components! Avoid electrostatic discharge!

The following protective measures must be observed for modules and components sensitive to electrostatic discharge (ESD)!

- Personnel responsible for storage, transport, and handling must have training in ESD protection.
- ESD-sensitive components must be stored and transported in the prescribed protective packaging.
- ESD-sensitive components may only be handled at special ESD-workplaces.
- Personnel, working surfaces, as well as all equipment and tools which may come into contact with ESD-sensitive components must have the same potential (e.g. by grounding).
- Wear an approved grounding bracelet. The grounding bracelet must be connected with the working surface through a cable with an integrated 1 MΩ resistor.
- ESD-sensitive components may by no means come into contact with chargeable objects, including most plastic materials.
- When ESD-sensitive components are installed in or removed from equipment, the equipment must be de-energized.

1.6 Documentation, software release and trademarks

Documentation

The present manual provides information on interface signals and program modules of the integrated PLC.

Overview of available documentation	**Part no.**		
	German	**English**	**French**
PNC-R – Connectivity Manual for project engineering and maintenance	1070 073 704	1070 073 736	–
PNC-R – Software installation	1070 073 796	1070 073 797	–
PNC-P – Connectivity Manual	1070 073 880	1070 073 881	–
PNC-P – BF2xxT/BF3xxT Control Panel Connectivity Manual	1070 073 814	1070 073 824	–
PNC-P – Software installation	1070 073 882	1070 073 883	–
Description of functions	1070 073 870	1070 073 871	–
MACODA Operation and configuration of the machine parameters	1070 073 705	1070 073 742	–
Operating instructions – Standard operator interface	1070 073 726	1070 073 739	1070 073 876
Operating instructions – Diagnostics Tools	1070 073 779	1070 073 780	–
Error Messages	1070 073 798	1070 073 799	–
PLC project planning manual, Software interfaces of the integrated PLC	1070 073 728	1070 073 741	–
iPCL system description and programming manual	1070 073 874	1070 073 875	–
ICL700 system description (Typ3 osa only), Program structure of the integrated PLC ICL700	1070 073 706	1070 073 737	–
DIN programming manual for programming to DIN 66025	1070 073 725	1070 073 738	–
CPL programming manual	1070 073 727	1070 073 740	–
CPL Debugger Operating Instructions	1070 073 872	–	–
Tool Management – Parameterization	1070 073 782	1070 073 793	–
Software PLC Development environment for Windows NT	1070 073 783	1070 073 792	–
Measuring cycles for touch-trigger switching probes	1070 073 788	1070 073 789	–
Universal Milling Cycles	–	1070 073 795	–

Release

☞ **The descriptive information contained in this manual applies to:**
Software version: V7.3

The current release number of the individual software modules can be viewed by selecting the ’Control-Diagnostics’ softkey in the ’Diagnostics’ group operating mode.

The software version of Windows95 or WindowsNT may be displayed as follows:

1. Click the right mouse button on the ’My Computer’ icon on your desktop.
2. Select menu item ’Properties’.

Trademarks

All trademarks of software installed on Rexroth products upon delivery are the property of the respective manufacturer.

Upon delivery, all installed software is copyright-protected. The software may only be reproduced with the approval of Rexroth or in accordance with the license agreement of the respective manufacturer.

MS-DOS® and Windows™ are registered trademarks of Microsoft Corp.

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

SERCOS interface™ is a registered trademark of the of the Interessengemeinschaft SERCOS interface e.V. (Joint VDW/ZVEI Working Committee).

2 Communication structure and interfaces

2.1 Overview

The PNC control units enable users to interact and integrate their own machine-specific developments in various function areas of the control unit.

Example:

- **MMI** (**M**an-**M**achine **I**nterface)
- **iPCL** (**i**ntegrated **P**ersonel **C**omputer and **C**ontrol **L**ogic)
- **ACB** (axis movement)
- **IPO** (**I**nter**PO**lation)
- **DB** (**D**ata**B**ase)

This manual describes software interfaces and their functioning in terms of the **PCL integrated** into the NC.

These can be:

● iPCL	in PNC-P or PNC-R
● ICL700	in the PNC-R
● KNS	in the development environment of PNC-P or PNC-R

☞ **A detailed description of the respective integrated PLC is given in further manuals, see page 1–7.**

The PLC sequential program is principally capable of communicating with all function areas and subsystems of the overall system.
The subsystems of the NC control, e.g. the man-machine interface (MMI) provide services for other subsystems (actual value display, status information, etc.).

These services can be called up by the PLC sequential program:

- via the NCS
- via APS modules (iPCL, ICL700)
- via APS modules in high-level language (ICL700)

2.2 The NCS (NC interface)

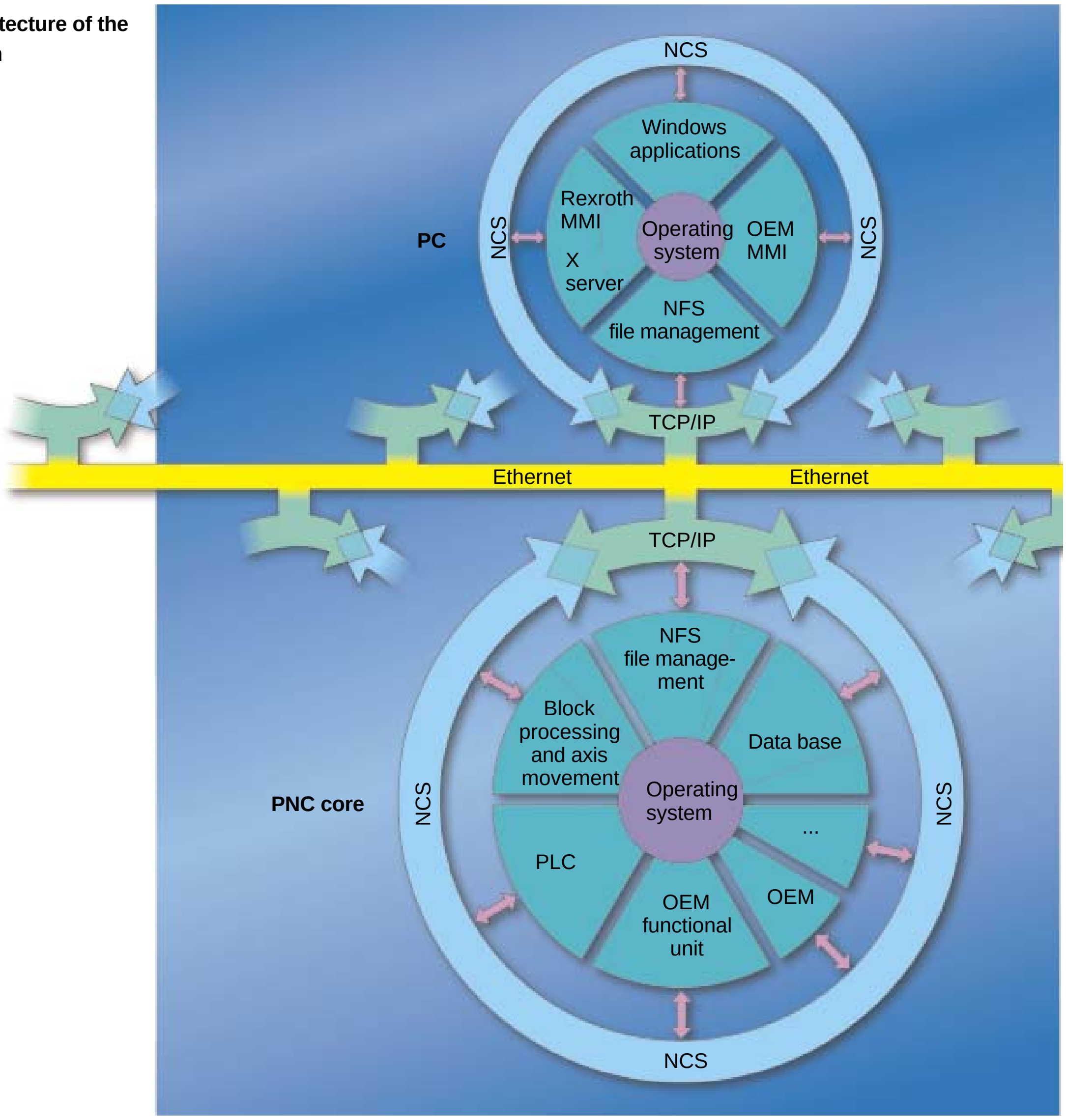
The **NC interface** (NCS) is the communication basis of the PNC and links the internal and external computer systems. It contains functions for information and data exchange among the control subsystems according to the client/server principle.

The standard **Ethernet coupling** (via TCP/IP) allows further control units and PCs to be directly connected to the control unit core via NFS file management as well as the integration of the system into networks.

The following areas communicate via the NC interface (NCS) and via the TCP/IP:

- **PNC**: iPCL, block processing and axis movement, data base, OEM functional units possible.
- **PC**: Rexroth MMI, OEM MMI, Windows applications, NFS file management)

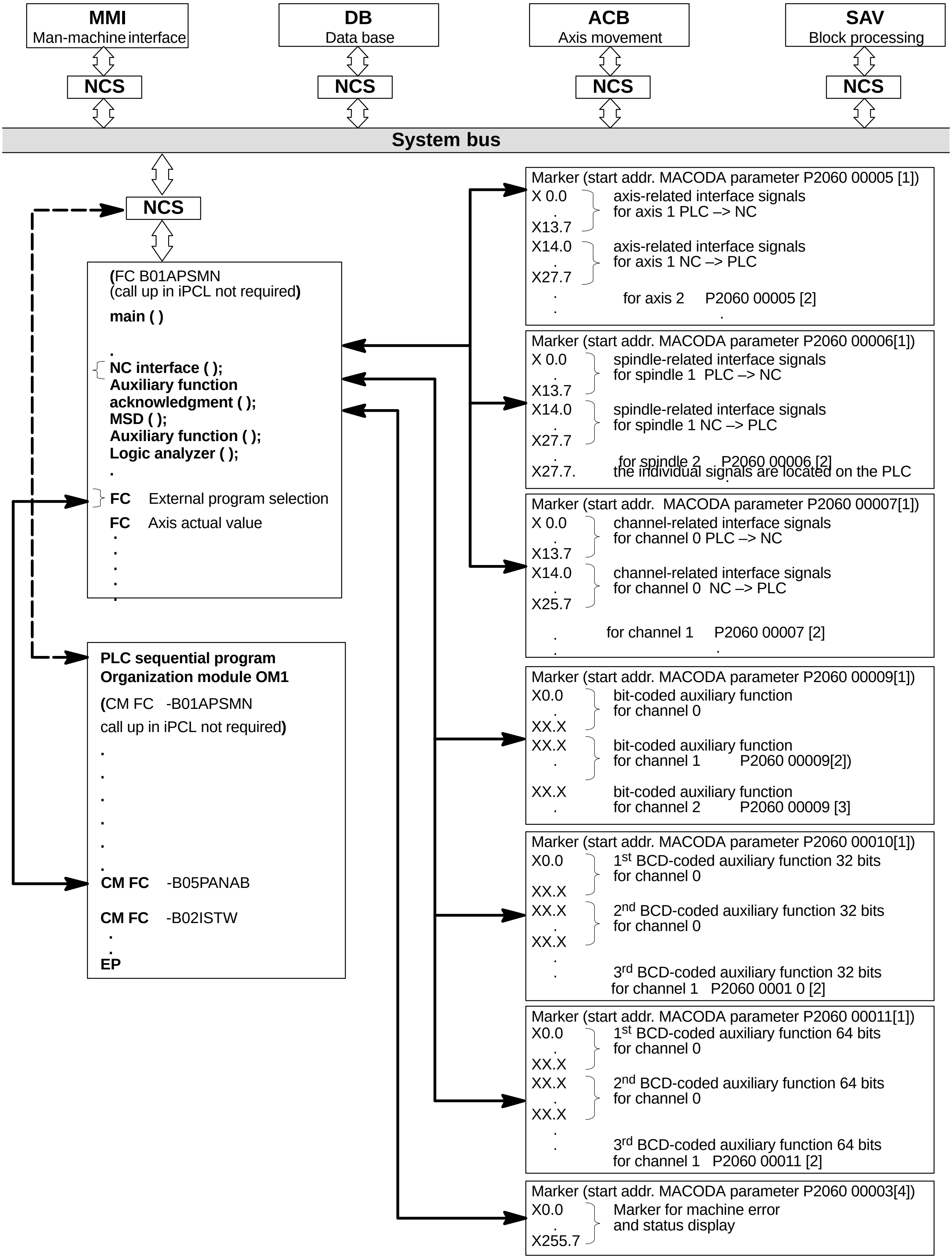
System architecture of the PNC platform



☞ **Data exchange to the PNC-P can only be performed via APS modules (not via NCS) at present.**

Example of an NCS communication structure in iPCL and ICL700

(for KNS communication structure, see ”Software PLC development environment” manual)


MMI
Man-machine interface
DB
Data base
ACB
Axis movement
SAV
Block processing
NCS
NCS
NCS
NCS
System bus
NCS
(FC B01APSMN
(call up in iPCL not required)
main ( )
NC interface ( );
Auxiliary function
acknowledgment ( );
MSD ( );
Auxiliary function ( );
Logic analyzer ( );
FC External program selection
FC Axis actual value
PLC sequential program
Organization module OM1
(CM FC -B01APSMN
call up in iPCL not required)
CM FC -B05PANAB
CM FC -B02ISTW
EP
Marker (start addr. MACODA parameter P2060 00005 [1])
X 0.0
X13.7
axis-related interface signals
for axis 1 PLC –> NC
X14.0
X27.7
axis-related interface signals
for axis 1 NC –> PLC
for axis 2 P2060 00005 [2]
Marker (start addr. MACODA parameter P2060 00006[1])
X 0.0
X13.7
spindle-related interface signals
for spindle 1 PLC –> NC
X14.0
X27.7
spindle-related interface signals
for spindle 1 NC –> PLC
for spindle 2 P2060 00006 [2]
X27.7.
the individual signals are located on the PLC
Marker (start addr. MACODA parameter P2060 00007[1])
X 0.0
X13.7
channel-related interface signals
for channel 0 PLC –> NC
X14.0
X25.7
channel-related interface signals
for channel 0 NC –> PLC
for channel 1 P2060 00007 [2]
Marker (start addr. MACODA parameter P2060 00009[1])
X0.0
XX.X
bit-coded auxiliary function
for channel 0
XX.X
bit-coded auxiliary function
for channel 1 P2060 00009[2])
XX.X
bit-coded auxiliary function
for channel 2 P2060 00009 [3]
Marker (start addr. MACODA parameter P2060 00010[1])
X0.0
XX.X
1st BCD-coded auxiliary function 32 bits
for channel 0
XX.X
XX.X
2nd BCD-coded auxiliary function 32 bits
for channel 0
3rd BCD-coded auxiliary function 32 bits
for channel 1 P2060 0001 0 [2]
Marker (start addr. MACODA parameter P2060 00011[1])
X0.0
XX.X
1st BCD-coded auxiliary function 64 bits
for channel 0
XX.X
XX.X
2nd BCD-coded auxiliary function 64 bits
for channel 0
3rd BCD-coded auxiliary function 64 bits
for channel 1 P2060 00011 [2]
Marker (start addr. MACODA parameter P2060 00003[4])
X0.0
X255.7
Marker for machine error
and status display

2.3 The APS (**AP**plication **s**oftware)

The **APS** is a software package developed by Rexroth.

The APS has the task of simplifying the data exchange for the PLC project engineer with the **subsystems of the control unit** and providing relevant data for the PLC sequential program in the form of markers, data modules or other operands to the PLC (iPCL), or transferring data contained in these operands to the NC.
The APS is continuously upgraded by Rexroth and expanded with other useful functions for the user.

The APS utilizes the functions of the NCS in order to perform a data exchange with the subsystems of the control unit.

The APS provides the NC output signals in markers and transfers the NC input signals, MSD signals and signals of the logic analyzer to the corresponding function areas in the control unit.
The APS also contains a number of program modules to be parameterized by the user which – after being called up by the PLC sequential program – select an NC program, read axis actual values or program statuses and transfer them to the iPCL. The program modules are fully accessible and facilitate the creation of the PLC sequential program by simple parameterization and calling (CM -FC1,..).

☞ **The program examples shown in the context of APS program module descriptions show the PROFI software syntax. For PLC user programs written with WinSPS, this syntax must be adapted correspondingly.**

Compatibility of iPCL with ICL 700

The functions of the APS modules are integrated in the iPCL firmware. The modules themselves only contain the frame for the actual function calls. Thus it is ensured that the APS functions always match the NC software.

The mandatory call up of module –B01APSMN required in the iPCL is not required at the start of the OM1, because this function is performed within the iPCL firmware at the beginning of the PLC processing. Calling up this module does not lead to an error, however, so that the existing programs remain operable.
If program module –B01APSMN or other program modules supplied by Rexroth in relation with –B01APSMN are modified, a re-initialization of the control unit is necessary. Initialization can be initiated by switching off and on or by using the RESET button.

If an existing ICL project is taken over, the APS modules of the ICL project have to be replaced by the iPCL APS modules, either by rewriting the APS modules or by adapting the library path, depending on the programming.

Self-created high-level language modules must be generated again in the iPCL project using the WinSPS.

The configuration of the PROFIBUS-DP has to be performed with WinDP and loaded. A remote configuration at runtime is possible in the iPCL.

Restrictions for the ICL 700:

Timer 255 of the program module –B01APSMN has been assigned.

In addition to the program module –B01APSMN, the following library modules have to be linked to the PLC program depending on the programming device software used.

- PROFI: – SPS_LIB.PAH as OM48 and
  – NCS_LIB.PAH as OM52
- WinSPS: – ICLLIB.PXH als OM48 and
  – NCSLIBW.PXH as OM52

If APS modules are called in timed modules (OM18#), make sure that these APS modules are called in the 1st PLC cycle only after module –B01APSMN has been called in OM1.
This can be ensured either by disabling the timer interrupt (DAI TI) in start-up module OM5, or by scanning for the ”Initializing pulse after Power on” (SM20.7) and skipping the remaining logic in the timed module by using:

```
A   SM20.7
EMC
```

If you choose the first procedure above, the timer interrupts may be enabled again after module –B01APSMN has been called (EAI TI).

If the function of the logic analyzer is used, the **B06LGANA** module must be linked to the PLC program. This module is only used internally and may therefore **not** be **called up** by the user program.

The possibility to implement PLC modules in the high-level language **C** as well means that the user program can also use this interface to communicate with the NC. The following points must also be taken into account in addition to the NCS documentation:

- The high-level language module may not contain a call for the function **Ncs_Receive** or **Ncs_Receive_NoWait**.
- Only **asynchronous NCS functions** may be used, and a receive function must be entered when the NCS function is called. This receive function is called up by the module **B01APSMN** as soon as the response to the request is received.
- As each high-level language module is allocated a standard 400 bytes stack and the NCS functions have different stack requirements, a test must be carried out via the **sps_stack_test** function to ascertain whether the stack is sufficient. This function is contained in the SPS_LIB.PAH library module and returns the remaining available stack. Where required, the **#define STACKSIZE** instruction can be used to define a larger or smaller stack size in the source code of the high-level language module. For runtime reasons, the call-up for measurement of the free stack should be cancelled **after commissioning of the module**.

☞ **NCS accesses created as high-level language modules can only be performed via APS modules.**

In respect of the **PROFI** programming software (only with ICL700), please note the following rules:

- a **link file** must exist for each high-level language module which uses the NCS interface.
  The name of any such **link file** must read **<module_name>.lnk** and must contain the following entries:
  - <module_name>.o
  - ncs5_fun.a (nsc5_fun.a is a c-library supplied with the APS modules).

3 PLC<–>NC communication via APS

The following is a description of the main features of the functions as implemented by the APS and of how they can be addressed directly from the PLC sequential program via the PLC operands.

Functions which are required frequently by the PLC sequential program or the NC are refreshed independently by the APS every PLC cycle, or are transferred in the appropriate direction when an event occurs.
These functions are usually concerned with **single signals** which signal a particular status to the PLC sequential program (e.g. axis in position, auxiliary function output) or activate a function in the NC (e.g. cycle start, stop, feed enable).

They are subdivided into:

- channel-related signals
- axis-related signals
- spindle-related signals
- decoded auxiliary functions
- machine error and status display

For iPCL and ICL700 please note:

Apart from the transfer of single signals, other functions are implemented in the APS software package which are required by the user only rarely or for special applications. These functions can be called and activated by the user via parameterizable program modules (CM –FCx,..).
After the call in the PLC sequential program, the APS converts the job to the NCS (NC interface) and performs the data transfer with the relevant subsystem of the control unit.
Each call of a program module is acknowledged by the APS and, according to the called function, the requested date is stored in the operands parameterized by the PLC sequential program.

Numerous functions are implemented as **program modules** in the APS and can be called with parameter settings in the PLC sequential program.

☞ **For available program modules, see Chapter 12.**

Notes:

4 General interface signals

4.1 Overview of general interface signals

General interface signals are higher-level signals used with functions relating to the NC as a whole.
The start address is defined in MACODA parameter 2060 00008.

☞ **With –1 as start address, the general interface is not evaluated (= presetting of MACODA parameter 2060 00008).**

4.1.1 Overview of input signals NC-I (PLC –> NC)

NC-I	Signal designation	NC-I	Signal designation
0.0	System control reset	1.0	Stroke inhibit
0.1	Edit inhibit	1.1	Stroke reservation
0.2	res.	1.2	Stroke ON
0.3	res.	1.3	res.
0.4	res.	1.4	res.
0.5	res.	1.5	res.
0.6	res.	1.6	res.
0.7	res.	1.7	res.
2.0	res.	3.0	res.
2.1	res.	3.1	res.
2.2	res.	3.2	res.
2.3	res.	3.3	res.
2.4	res.	3.4	res.
2.5	res.	3.5	res.
2.6	res.	3.6	res.
2.7	res.	3.7	res.

4.1.2 Overview of output signals NC-O (NC –> PLC)

NC-O	Signal designation	NC-O	Signal designation
4.0	res.	5.0	Stroke intended
4.1	res.	5.1	Stroke is not running
4.2	res.	5.2	res.
4.3	res.	5.3	res.
4.4	res.	5.4	res.
4.5	res.	5.5	res.
4.6	res.	5.6	res.
4.7	res.	5.7	res.
6.0	res.	6.0	res.
6.1	res.	6.1	res.
6.2	res.	6.2	res.
6.3	res.	6.3	res.
6.4	res.	6.4	res.
6.5	res.	6.5	res.
6.6	res.	6.6	res.
6.7	res.	6.7	res.

4.2 Signal description

4.2.1 Input signals NC-I

System reset NC-I 0.0

This signal acts on all channels of the NC. Its effect on the individual channels is identical with a control reset of each single channel.

System reset has effects on:

- the axis-related output signals
- the channel-related output signals:
  - Control reset acknowledge
  - NC ready
  - Program running

Signal level

1: A system reset is effected on **all** channels.

0: No effect.

Edit inhibit NC-I 0.1

Edit inhibit prohibits the user to perform certain operations at the NC. Only user actions directly related with the production process are allowed.

The effect of the edit inhibit signal is channel-independent.

Signal level

1: The following functions are either locked or restricted:

- file management
- tables
- editor
- MACODA

0: No effect.

CAUTION
Edit inhibit does not act on tool management.
If required, tool management can be locked via the PLC (see manual ”Tool Management – Parametrization”).

Overview of user levels to which edit inhibit applies:

User level	Edit inhibit applicable
system	no
developer	no
MTB	no
setter	no
user	yes

☞ **When activated, edit inhibit has an effect on the lowest user level only.**

For further information on edit inhibit, see the manual ”Operating instructions, Standard operator interface”.

Stroke inhibit NC-I 1.0

The **stroke inhibit** signal has the effect that the PLC locks the release of punching strokes (by setting the high-speed output to 0 (HSO-0)) on the osa dc interface module (PNC-R with SNCI4). See illustration punching HS-logic.

Signal level

1: The high-speed output HSO-0 is disabled.

0: No effect.

Stroke reservation NC-I 1.1

The **stroke reservation** signal has the effect that the PLC will reserve the high-speed output HSO-0 on the osa dc interface module (PNC-R with SNCI4) for a stroke to be released via the PLC (see illustration under Punching HS-logic).

Signal level

1: The high-speed output HSO-0 is reserved for the PLC.

0: No effect.

Stroke ON NC-I 1.2

With the **stroke ON** signal, the PLC instructs the NC to release a punching stroke via the high-speed output HSO-0 on the osa dc interface module (PNC-R with SNCl4).

Signal edge / level

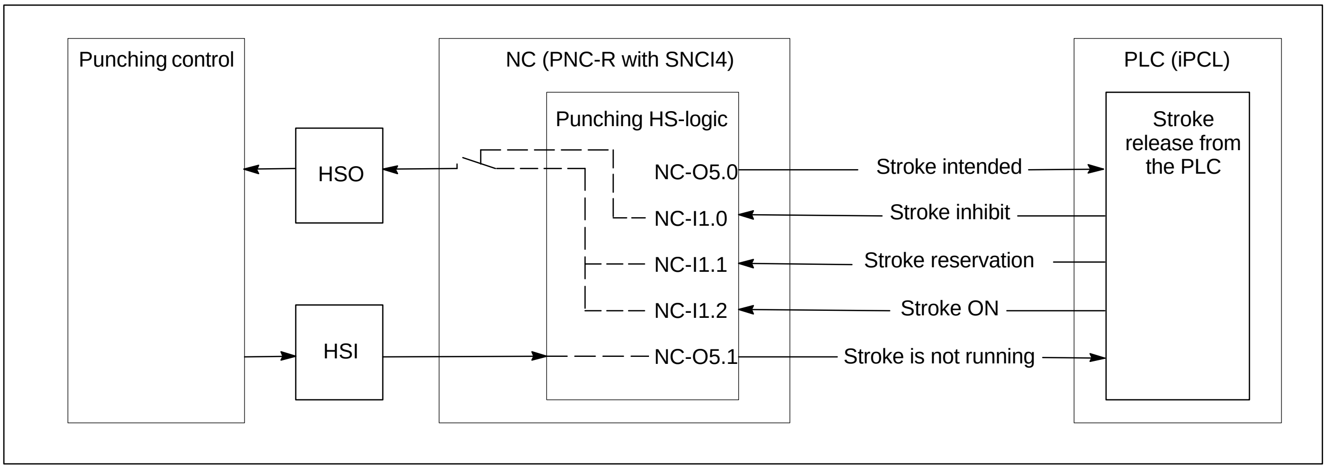
0 → 1: Punching stroke to be released from the PLC via the high-speed output HSO-0.

0: No effect.

Punching HS-logic

Individual strokes can be released from the PLC.
The PLC can release a stroke by instructing the NC via the NC-PLC interface to do so.

The bit signals used in the communication among the NC, the punching control and the PLC are shown in the following graph:

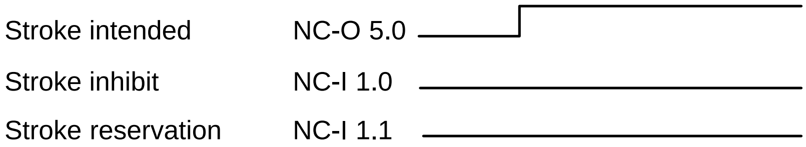


The punching HS-logic can process five NC-PLC signals:

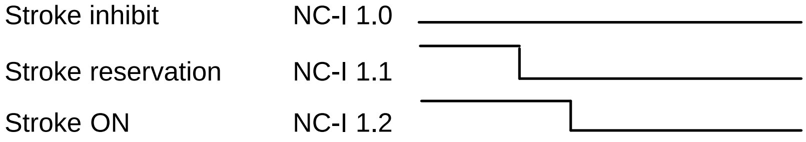
NC-O5.0	”Stroke intended”	With this signal, the NC communicates to the PLC that the NC wants to release a stroke.
NC-O5.1	”Stroke is not running”	The high-speed input HSI-0 is cleared by the punching HS-logic for the PLC.
NC-I1.0	”Stroke inhibit”	This allows the PLC to block the setting of HSO-0.
NC-I1.1	”Stroke reservation”	This allows the PLC to reserve the high-speed output HSO-0 for a stroke to be released from the PLC.
NC-I1.2	”Stroke ON”	With this signal, the PLC instructs the NC to release a stroke.

Punching strokes can be released both from the NC and from the PLC:

Stroke release from the NC:



Stroke release from the PLC:



4.2.2 Output signals NC-O

Stroke intended NC-O 5.0

With the **stroke intended** signal, the NC communicates to the PLC that the NC wants to release a punching stroke. If the PLC permits the stroke to be released (NC-I 1.0, ”stroke inhibit” = 0 and NC-I 1.1, ”stroke reservation” = 0), the high-speed output on the osa dc interface module (PNC-R with SNCI4) is set to HSO-0 by the punching HS-logic (see also Chapter 4.2.1).

Signal level

1: The NC communicates to the PLC that the NC wants to release a stroke.

0: No effect.

Stroke is not running NC-O 5.1

The response signal **stroke is not running** from the punching control (high-speed input HSI-0 on the osa dc interface module (PNC-R with SNCI4)) is relayed to the PLC.

Signal level

1: Stroke is not running = response signal from the punching control to the PLC.

0: Stroke running.

5 Channel-related interface signals

5.1 Overview of channel-related interface signals

The **channel structure** of the control unit is designed to allow for several tasks to be processed simultaneously.
Channel-related interface signals thus mainly affect NC functions which are related to the execution of the NC part program.

Every channel has its own interface, i.e. with n channels, the interface depicted in sections 5.1.1 and 5.1.2 is mapped n times on the corresponding markers.
In addition to the channel-related interface, the axis-related interface is of significance to the machining axes combined to form a channel as well as to the auxiliary axes.

The start addresses for the channel-related interface signals are defined in MACODA parameter 2060 00007.
There must always be at least 2 channels entered:

- channel 0 for switching functions
- channel 1 for one machining channel.

The following description shows the interface signals for one channel.

5.1.1 Overview of input signals NC-I (PLC –> NC)

NC-I	Signal designation	NC-I	Signal designation
0.0	Conditional jump	1.0	Cycle start
0.1	res.	1.1	Control reset
0.2	res.	1.2	Override 100%
0.3	res.	1.3	res.
0.4	res.	1.4	Automatic restart
0.5	res.	1.5	res.
0.6	Workpiece coordinates manual +	1.6	Optional stop
0.7	Workpiece coordinates manual –	1.7	Limit rapid travel
2.0	Mode selection bit 0	3.0	Mode selection by PLC
2.1	Mode selection bit 1	3.1	Return to contour
2.2	Mode selection bit 2	3.2	G131 Tool rotation release
2.3	Mode selection bit 3	3.3	Cancel distance to go
2.4	res.	3.4	Block skip 1
2.5	res.	3.5	res.
2.6	Online correction enable	3.6	res.
2.7	Online correction direction of rotation	3.7	res.

NC-I	Signal designation	NC-I	Signal designation
4.0	Override bit 0 (Override value)	5.0	Override bit 8 (with direct input in 0.01%)
4.1	Override bit 1 (Override value)	5.1	Override bit 9 (with direct input in 0.01%)
4.2	Override bit 2 (Override value)	5.2	Override bit 10 (with direct input in 0.01%)
4.3	Override bit 3 (Override value)	5.3	Override bit 11 (with direct input in 0.01%)
4.4	Override bit 4 (Override value)	5.4	Override bit 12 (with direct input in 0.01%)
4.5	Override bit 5 (with direct input in 0.01%)	5.5	Override bit 13 (with direct input in 0.01%)
4.6	Override bit 6 (with direct input in 0.01%)	5.6	Override bit 14 (with direct input in 0.01%)
4.7	Override bit 7 (with direct input in 0.01%)	5.7	Override bit 15 (with direct input in 0.01%)
6.0	Customer input 1	7.0	Automatic reselection OFF
6.1	Customer input 2	7.1	Switch to next block
6.2	Customer input 3	7.2	Coordinate coupling completed
6.3	Customer input 4	7.3	res.
6.4	Customer input 5	7.4	Rapid retraction
6.5	Customer input 6	7.5	Feed inhibit
6.6	Customer input 7	7.6	Feed hold
6.7	Customer input 8	7.7	Block transfer inhibit
8.0	Feed inhibit general 1	9.0	Feed inhibit general 9
8.1	Feed inhibit general 2	9.1	Feed inhibit general 10
8.2	Feed inhibit general 3	9.2	Feed inhibit general 11
8.3	Feed inhibit general 4	9.3	Feed inhibit general 12
8.4	Feed inhibit general 5	9.4	Feed inhibit general 13
8.5	Feed inhibit general 6	9.5	Feed inhibit general 14
8.6	Feed inhibit general 7	9.6	Feed inhibit general 15
8.7	Feed inhibit general 8	9.7	Feed inhibit general 16
10.0	Feed hold 1	11.0	Feed hold 9
10.1	Feed hold 2	11.1	Feed hold 10
10.2	Feed hold 3	11.2	Feed hold 11
10.3	Feed hold 4	11.3	Feed hold 12
10.4	Feed hold 5	11.4	Feed hold 13
10.5	Feed hold 6	11.5	Feed hold 14
10.6	Feed hold 7	11.6	Feed hold 15
10.7	Feed hold 8	11.7	Feed hold 16
12.0	Block transfer inhibit 1	13.0	Block transfer inhibit 9
12.1	Block transfer inhibit 2	13.1	Block transfer inhibit 10
12.2	Block transfer inhibit 3	13.2	Block transfer inhibit 11
12.3	Block transfer inhibit 4	13.3	Block transfer inhibit 12
12.4	Block transfer inhibit 5	13.4	Block transfer inhibit 13
12.5	Block transfer inhibit 6	13.5	Block transfer inhibit 14
12.6	Block transfer inhibit 7	13.6	Block transfer inhibit 15
12.7	Block transfer inhibit 8	13.7	Block transfer inhibit 16

5.1.2 Overview of output signals NC-O (NC –> PLC)

NC-O	Signal designation	NC-O	Signal designation
14.0	Program running	15.0	Active mode bit 0
14.1	Feed hold active	15.1	Active mode bit 1
14.2	Control reset executed	15.2	Active mode bit 2
14.3	Block transfer inhibit active	15.3	Active mode bit 3
14.4	res.	15.4	res.
14.5	res.	15.5	res.
14.6	res.	15.6	res.
14.7	res.	15.7	res.
16.0	NC ready	17.0	Activate block skip 1
16.1	Program stop with M0, M1	17.1	res.
16.2	Program end with M2, M30	17.2	res.
16.3	Optional stop activated	17.3	Channel status bit 0
16.4	Ready for re-entry	17.4	Channel status bit 1
16.5	Re-entry active	17.5	Channel status bit 2
16.6	Remove finished	17.6	Channel status bit 3
16.7	res.	17.7	Channel status bit 4
18.0	G131 Tool rotation	19.0	External tool correction active bit 0
18.1	res.	19.1	External tool correction active bit 1
18.2	res.	19.2	External tool correction active bit 2
18.3	Override 0%	19.3	External tool correction active bit 3
18.4	Override 100%	19.4	External zero offset active bit 0
18.5	res.	19.5	External zero offset active bit 1
18.6	res.	19.6	res.
18.7	Inpos range 2 active	19.7	res.
20.0	Rapid traverse active	21.0	G189 active
20.1	G70 active	21.1	G90 active
20.2	G32 active	21.2	G91 active
20.3	G33 active	21.3	G92 active
20.4	Thread cycle active	21.4	G95 active
20.5	G41 active	21.5	Const. cutting velocity active
20.6	G42 active	21.6	Coordinate coupling completed
20.7	G63 active	21.7	res.
22.0	Customer output 1	23.0	General tool compensation active bit 0
22.1	Customer output 2	23.1	General tool compensation active bit 1
22.2	Customer output 3	23.2	General tool compensation active bit 2
22.3	Customer output 4	23.3	General tool compensation active bit 3
22.4	Customer output 5	23.4	res.
22.5	Customer output 6	23.5	res.
22.6	Customer output 7	23.6	res.
22.7	Customer output 8	23.7	res.
24.0	CPL customer output 1	25.0	CPL customer output 9
24.1	CPL customer output 2	25.1	CPL customer output 10
24.2	CPL customer output 3	25.2	CPL customer output 11
24.3	CPL customer output 4	25.3	CPL customer output 12
24.4	CPL customer output 5	25.4	CPL customer output 13
24.5	CPL customer output 6	25.5	CPL customer output 14
24.6	CPL customer output 7	25.6	CPL customer output 15
24.7	CPL customer output 8	25.7	CPL customer output 16

5.2 Signal description

Some signals which are required repeatedly in the context of various machine functions, e.g. **feed inhibit general** or **feed hold**, are available up to 16 times per channel.
This will enable project designers to expand the PLC sequential program step by step without having to change any logic operations of existing function modules.

☞ **In the iPCL and ICL 700 these individual signals are combined to one signal in the APS via an OR-link.**
These sum signals can only be created internally; the PLC program cannot access them.

For purposes of diagnostics, each signal can be shown individually in monitor operation.

5.2.1 Input signals NC-I

Conditional jump NC-I 0.0

Effective in conjunction with function G23, which is programmed in the NC part program.

Signal level

1: In the NC part program, a jump is executed to the NC block which is defined as jump destination.
For this purpose, the signal level logic 1 must be detected at the **time of block preparation** already.

Block preparation may be ahead of the active block by several NC blocks.
To make sure that the jump is actually carried out, it is necessary to interrupt the block preparation at an appropriate point (prior to the NC block with the G23 function), depending on the application. The interruption is effected in the part program using the **wait** command.

0: The jump in the part program is not performed.

Workpiece coordinates manual + NC-I 0.6
Workpiece coordinates manual – NC-I 0.7

These signals are effective in the **manual set-up workpiece coordinates** operating mode. For a detailed description of this operating mode, please refer to the ”Function Description” manual (part no. see page 1–7).

Signal edge

0 → 1: When a coordinate is selected in the ”manual workpiece coordinates” operating mode, the movement of the coordinate is started using manual + or manual –.

1 → 0: The continuous movement manual + or manual – is stopped.

Cycle start NC-I 1.0

Has effect in the operating modes **automatic, single block, single step, program block** and **MDI** NC block.
The signals NC ready, feed hold and program running are significant in conjunction with cycle start.

Signal edge

0 → 1: Starts the execution of a part program (automatic) or an NC block. **Feed hold** is reset and the axis movement is initiated again.

1 → 0: No effect.

Control reset NC-I 1.1

This signal has an effect on the axes interrelated by interpolation on **one** channel as well as on the part program or machining program running on this channel.

The input signal control reset has the same effect as the control reset function, which can be performed by pressing the softkey on the NC control panel.

In connection with the control reset input signal, the channel-related output signals **control reset executed**, **NC ready** and **program running** are significant. Furthermore, the input signal control reset has an effect on the axis-related output signals **axis in position**, **axis running** and **travel command**.

Signal level

1: One function of control reset is to cause the deselection of the NC part program and the resetting of G functions to the power-up state defined in MACODA parameter 7060 00020. Also, the machining axes are decelerated until they come to a halt.

When the control reset function has been completed internally, the **control reset executed** output signal is set and the control reset input signal is reset via the PLC sequential program.

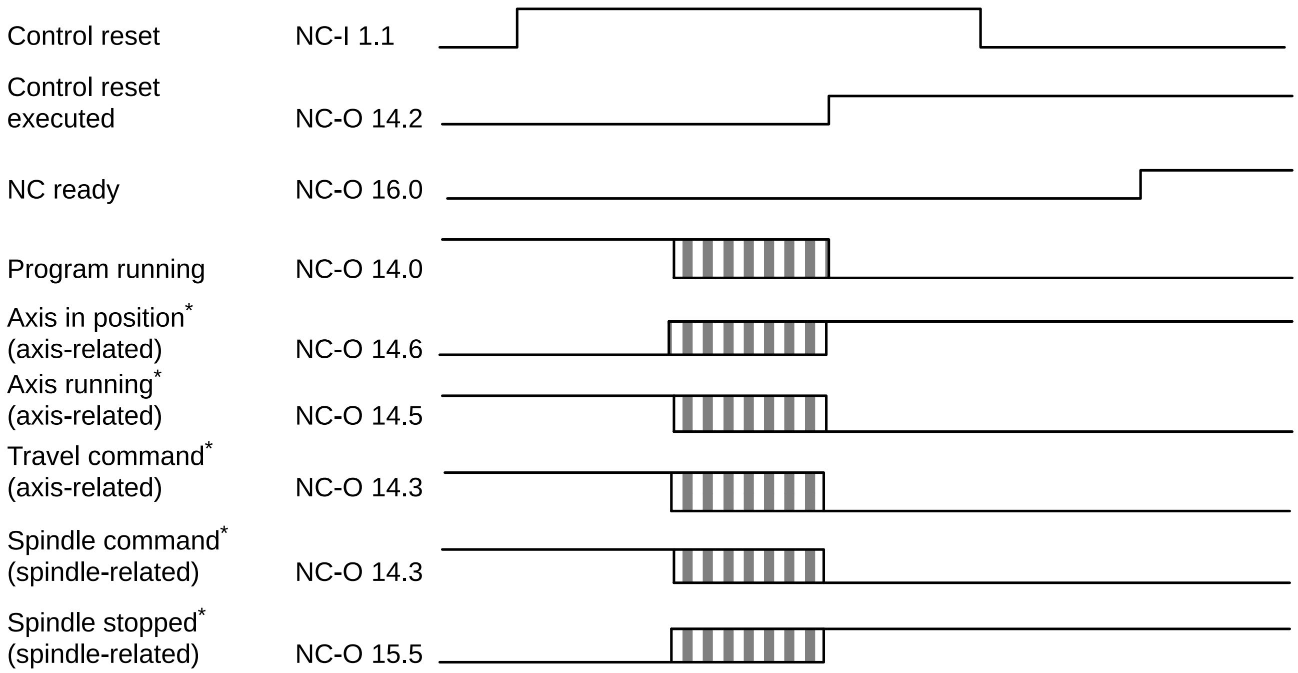
If the signal edges of all channels are simultaneously set from 0 to 1, a general control reset will be performed. This means that in addition to the individual resetting operations, auxiliary axes and spindles are decelerated until they come to a halt and cross-channel MACODA parameters are applied to them.

If the NC output signal **NC-O 20.3 G33 active** is set, **NC-I 1.1 control reset** will initiate a rapid retraction (see also NC-I 7.4. rapid retraction).

0: No effect.

Example:

The input signal **control reset** is set via the channel-related interface in order to abort the processing of a part program.



*) for auxiliary axes and spindles only with general control reset.

☞ **The signal NC ready is not set until an NC part program has been selected.**

Override 100% NC-I 1.2

Acts on all axes defined on the channel.

Signal level

1: Cancels the respective **override function**, the axes travel with the programmed or preselected speed.

0: The **override function** is effective.

Automatic Restart NC-I 1.4

Signal level

1: When a program has come to an end, it is restarted automatically.

0: No automatic program reselection.

Optional stop NC-I 1.6

Effective in conjunction with function **M1,** which is programmed in the NC part program.

Signal level

1: The execution of the NC part program is stopped after the NC block in which function **M1** has been programmed. It may be restarted with **cycle start**.

0: Function **M1** is not effective, the NC part program is not stopped.

Limit rapid travel NC-I 1.7

In operation modes manual data input of NC block, automatic, program block, single step, the G0 rapid travel is limited to a maximum value specified in MACODA Parameter 7030 00110 by setting the ”limit rapid travel” channel-related interface signal.

If the limited rapid traverse velocity is set after the program is started, it becomes effective only after all the blocks already prepared at this time have been executed.
If this is not the required action, stop machining by using the feed hold function, cancel the prepared blocks by ”Cancel distance to go”, before you continue machining by cycle start.

Channel override is valid also with the limit rapid travel function active, i.e. 100% override are exactly equivalent to the limit rapid travel velocity.

The limit rapid travel function is valid for all NC functions traversing at rapid travel velocity (G0, G10, G200, ...).

The limit rapid travel function does not affect spindle speed or the programmed feedrate.

This function has an effect only on the synchronous axes assigned to the respective channel.

Signal level

1: Limit rapid travel is active and all rapid travel blocks are prepared for execution at limit rapid travel velocity.

0: Limit rapid travel is inactive and all rapid travel blocks are prepared for execution at the maximum travel velocity possible.

Mode selection

NC-I 2.0 ... 2.3

The operating mode can entered using either the PLC or the NC operating panel (see **NC-I 3.0**).

The following operating modes are possible:

	Operating mode selection 2^0 NC-I 2.0	Operating mode selection 2^1 NC-I 2.1	Operating mode selection 2^2 NC-I 2.2	Operating mode selection 2^3 NC-I 2.3
No operating mode	0	0	0	0
Manual set-up	1	0	0	0
Manual Traverse to reference point	0	1	0	0
reserved	1	1	0	0
MDI NC block	0	0	1	0
Automatic	1	0	1	0
Program block	0	1	1	0
Single step	1	1	1	0
reserved	0	0	0	1
reserved	1	0	0	1
Single block	0	1	0	1
Return to path	1	1	0	1
CPL debugger program block	0	0	1	1
CPL debugger automatic	1	0	1	1
Manual set-up workpiece coordinates	0	1	1	1
reserved	1	1	1	1

Switching from one operating mode to another is possible on the following conditions:

Operating mode		**Mode switching, channel active**		**Mode switching, channel not active** (NC-O 14.2 = control reset was executed)
Automatic	Automatic mode group 1	yes, within group 1	yes, within group 1 and 2: if channel status is ready for program start, i.e. at the beginning or end of a program	yes
Single step				yes
Single block				yes
Program block	Automatic mode group 2	yes, within group 2		yes
CPL debugger program block				yes
CPL debugger automatic				yes
MDI NC block		yes, with a control reset executed for the channel prior to switching		yes
Return to path				yes
Manual set-up				yes
Manual traverse to reference point				yes
Manual set-up workpiece coordinates				yes

Online correction enable NC-I 2.6

This signal enables the ”online correction in workpiece coordinates” NC function for a selected coordinate.
In case of the online correction in workpiece coordinates, a handwheel is used to correct position and orientation in the workpiece coordinate system (WCS) of a channel. This correction is possible with active and inactive part program. For a detailed description, please refer to the ”Function Description” manual (part no. see page 1–7).

☞ **In the ”manual”, ”manual set-up workpiece coordinates” and ”manual approaching the reference point” operating modes, the online correction cannot be enabled.**

Signal level

1: The selected online correction processes the handwheel data..

0: The selected online correction does not respond to the handwheel data.

Online correction direction of rotation NC-I 2.7

Using this signal, it is possible to change the direction of correction.

Signal level

1: Positive direction of rotation of the handwheel results in **negative** online correction.

0: Positive direction of rotation of the handwheel results in **positive** online correction.

Mode selection by PLC NC-I 3.0

The selection of operating modes can be carried out manually via the NC panel or via the PLC sequential program.

Signal level

1: The operating mode is specified by the PLC via the **NC-I 2.0...2.3** input signals.
As long as signal level = 1 is set, it is not possible to preselect an operating mode via the PC operating panel.

0: The operating mode is entered via the **NC operating panel**.
As long as signal level = 0 is set, it is not possible to specify the operating mode via the PLC.

Return to contour NC-I 3.1

Signal level / edge

1: Following cycle start, the tool returns to contour on the stored departing movement.

0: Upon cycle start, machining is continued immediately.

0 → 1: Recording of the motion in departing from the contour is stopped.

G131 Tool rotation release NC-I 3.2

Signal **G131 tool rotation release**, allows the NC to execute an intermediate block, which is required if the contour knee angle between two blocks is wider than the adaptation angle specified and the tool axis must therefore be rotated (see also NC-O 18.0).

Signal level

1: The PLC sends the release to the NC to execute the intermediate block.

0: The release of the execution of an intermediate block is terminated. Now, the NC can go on to execute the next block.

Cancel distance to go NC-I 3.3

Has effect in the operating modes **automatic**, **single block**, **single step**, **program block** when the program is active and in **manual data input**.

First the program is stopped using feed hold (NC-I 7.6) (this is not compulsory). The display shows the distance the axes still have to traverse until the end of the block.
Having initiated ”cancel distance to go”, all prepared NC blocks, including the remainder of the current block are cancelled and prepared again.

Please note:

- CPL blocks or CPL parts are not taken into consideration:

  Example:
  CPL variable POS was given value 10 in preparation, the X[POS] NC word is interpreted as X10 following ”cancel distance to go” even though POS may have a totally different value at that time.
- Possibly changed compensation values are taken into account.

In the display, the indicated end point is set to the current position, at the same deleting the indicated distance to go. Then the channel goes into NC ready status (**NC-O 16.0**).
Following cycle start (**NC-I 1.0**), the program is continued at the **point of interruption**, taking into account the new compensation values.

Signal edge

0 → 1: Triggers cancel distance to go when the program is active.

1 → 0: No effect.

☞ **This signal must not be set when NC-O 20.3 (G33 active) is active .**

Block skip NC-I 3.4

Has effect in operating modes **automatic, program block, single block** and **single step.**

Signal level

1: If the signal level is logic 1 at the time of **block preparation**, the NC blocks marked with a slash "/" in the NC part program are skipped during execution.
Block preparation may be ahead of the active block by several NC blocks. To make sure that the skip is actually carried out, it is necessary to interrupt the block preparation at an appropriate point (prior to the NC block with the slash /), depending on the application. Interruption of the block preparation is effected in the part program using the **wait** command.

0: The NC block with the slash **"/"** is not skipped.

Override NC-I 4.0 ... 4.4

The traversing speed of all machining axes assigned to a channel can be controlled with the **override function.**

The override function has effect on the speeds programmed in the part program and on the preselectable speeds in manual mode.
The **override switch** can have a maximum of **32 positions,** which are queried by the NC in **binary coded** form.

Each binary code is assigned a value via MACODA parameter 7030 00010, which represents the respective current speed in per cent. If all the settings in MACODA parameter 7030 00010 are 0 (except 1: max. override value), the NC will interpret the value of the two input bytes 4 and 5 directly as being the override value stated in 0.01%.

☞ **Channel 0 defines the values for the auxiliary axes, for which the values cannot be set directly.**

Automatic reselection OFF NC-I 7.0

The **automatic reselection OFF** signal determines whether or not a program is to be automatically reselected.

Signal level

1: Automatic program reselection is suppressed.

0: Automatic program reselection is permitted.

Switch to next block NC-I 7.1

The switch to next block signal is effective in the active program in the **automatic, single block, single step** and **program block** operating modes.

Signal edge

0 → 1: The synchronous axis movement of the channel is stopped with the admissible path deceleration. Subsequently, all prepared NC blocks are cancelled. The channel changes to the **NC ready** state (NC-O 16.0)
After cycle start (NC-I 1.0), the subsequent blocks are prepared again and executed. The not yet executed part of the interrupted NC block is ignored in the process.
To be able to activate the function selectively, it is necessary to previously stop the program at a defined position using feed hold.

1 → 0: No effect.

☞ **This function generally only works if the aborted block is followed by a linear block (due to the changed starting position, a consistent circle is no longer achieved using e.g. the programmed parameters).**

In case of active cutter-path compensation, the function mostly leads to undesired traversing movements and thus to the destruction of the part to be machined.

Coordinate coupling completed NC-I 7.2

Using the **coordinate coupling completed** signal, the coordinate coupling for a slave channel is terminated.

Since the master channel has to stand still when the slave is coupled, it is recommended to stop the master during the coupling process using block **transfer inhibit** and **feed inhibit general**. Both signals can be reset by the slave channel outputting the coordinate coupling active signal.
When coordinate coupling is completed, the axes of the master channel should be standing still.

Signal edge

0 → 1: The coordinate coupling for a slave channel is completed. The slave channel then switches to the next block, the NC-O 21.6 signal is set to logic 0.

1 → 0: No effect.

Rapid retraction NC-I 7.4

Triggers the retraction movement in case of active thread cutting, if the **NC-O 20.3 G33 active** signal has been set and the cutting movement has started.

Prerequisite:
The retraction movement is configured via MACODA 7050 00645 – 7050 00650 or using G533 RD(,) function and released via MACODA 7050 00640 or using G533 RON1 function.

Signal level

1: Retraction from the contour is initiated.
A linear retraction movement is superimposed on the ”second cutting axis”, which moves the cutting tool away from the machined part.

Second cutting axis:

- secondary axis of the plane in case of longitudinal thread, or
- main axis of the plane in case of face thread

0: No effect.

Feed inhibit general NC-I 8.0 ... 9.7

Acts simultaneously on all machining axes assigned to a channel in operating modes **automatic**, **single block/single step** and NC block **manual data input** (MDI).

The **axis-related feed inhibit signal** has priority over the **channel-related feed inhibit general signal.**

This signal has no effect on auxiliary axes since auxiliary axes are never assigned to any particular channel.

Signal level

1: The axes do not traverse while the signal is set.
If the signal is set while the axes are in motion, all axes assigned to the channel are slowed down to a halt at the respective path deceleration.
If the NC-O 20.3 (G33 active) NC output signal has been set, this has no effect. The feedrate is dependent on the spindle speed and the thread pitch.

0: Axes start traversing immediately upon cycle start.
If the axes were stopped previously by feed inhibit general, they accelerate to the preset speed after feed inhibit general is reset.

Feed hold NC-I 10.0 ... 11.7

Has effect on all **machining axes** which are allocated to **one channel.**

Feed hold has no effect on NC blocks in which no movement is programmed, i.e. the execution of the NC program is continued until a machining axis is about to perform a traversing movement.

Feed hold is a latching signal. As long as **feed hold** is active, the interface signal **feed hold active** remains set.
Feed hold is unlatched as soon as the channel-related interface signal **cycle start** is set.

Signal level

1: Axis motions are slowed down to halt at the respective path deceleration and traversing motions cannot be started again until feed hold is unlatched.
If the **NC-O 20.3** (G33 active) NC output signal has been set, the NC-O 10.0 ... 11.7 feed hold signal has **no effect**.
In this case, the feedrate is dependent on the spindle speed and the thread pitch.

0: As soon as **feed hold** is reset, it can be unlatched by setting **cycle start** and the next traversing motion can be started.

Block transfer inhibit NC-I 12.0 ... 13.7

Acts in operating modes **automatic**, single block, and NC block manual data input (MDI).

Signal level

1: Execution of the next NC block is inhibited. However, execution of the currently active block will be completed. Upon completion of the active block, the signal **block transfer inhibit active** is set.
The part program and the programmed NC block cannot be executed with cycle start while block transfer inhibit is set.
The block transfer inhibit signal will not stop **block preparation**, which continues to run until there is no more memory space allocated for block preparation.

0: If the execution of an NC part program was previously interrupted by setting the block transfer inhibit signal, execution of the NC part program will be resumed when the signal is reset.

5.2.2 Output signals NC-O

Program running NC-O 14.0

Has an effect in operating modes **automatic**, **program block**, **single block** and **single step** only if a program or, in **MDI** mode, an NC block has been selected previously.

Signal level

1: The signal is set if an NC part program has been started previously with **cycle start** in operating modes automatic, program block, single block or single step, or if the execution of a single NC block has been initiated with cycle start while in **MDI** operating mode.
The input signals **feed inhibit general, block transfer inhibit** and **drive off** have no effect on this signal.
If **override = 0** is entered, the signal level remains at logic 1.

0: The program running signal is not set if

- no program has been selected in operating modes automatic, program block, single block or single step,
- a program has been selected or, while in MDI mode, a block activated but execution of this program or block has not started yet,
- execution of a block in operating modes program block, single block or single step is finished but execution of the next block has not started yet,
- NC functions **M0, M1, M2** or **M30** were executed previously, or
- the feed hold signal (NC-I 7.6) has been set.

Feed hold active NC-O 14.1

This signal remains set as long as the signal **feed hold** is latched.

Signal level

1: The feed hold signal is latched; no traversing motions of machining axes can be executed until the feed hold signal is reset by **cycle start**.

0: Feed hold is not active and traversing motion of machining axes can be executed.

Control reset executed NC-O 14.2

This signal has effect in conjunction with the input signal control reset and the control reset function which is initiated via the NC control panel using a softkey (see also signal diagram Control reset NC-I1.1).

Signal level

1: The control reset function has been carried out internally by the control unit and the channel is in power-up state.
The input signal control reset must be reset by this signal.
Signal level 1 is maintained for as long as the channel is in power-up state and no program has been selected.

0: The channel is no longer in power-up state, an NC part program has been selected or is already active.

Block transfer inhibit active NC-O 14.3

The signal **block transfer inhibit active** indicates that the channel will, due to an active block transfer inhibit, delay the execution of blocks until the block transfer inhibit is reset.

Signal level

1: After a block transfer inhibit has been activated and execution of the NC block active at that time has been completed, the channel waits for the block transfer inhibit to be reset and sets the ”block transfer inhibit active” signal to high.
This signal can be used if, for example, **cancel distance to go** is to be executed for the transfer of the external compensation at the block end. When the **block transfer inhibit** signal has been set and the **block transfer inhibit active** signal has been output, the interface signal **cancel distance to go** can be used to delete the prepared blocks, so that the transferred compensation values are active as of the next block.

0: The channel does not wait for a block transfer inhibit to be reset.

☞ **This signal is not an acknowledgement signal indicating that the block transfer inhibit has been recognized by the NC; it only indicates that the block transfer inhibit has become active.**

Active mode NC-O 15.0 ... 15.3

The active operating mode is output in coded form:

	Operating mode selection 2^0 NC-I 2.0	Operating mode selection 2^1 NC-I 2.1	Operating mode selection 2^2 NC-I 2.2	Operating mode selection 2^3 NC-I 2.3
No operating mode	0	0	0	0
Manual set-up	1	0	0	0
Manual traverse to reference point	0	1	0	0
reserved	1	1	0	0
MDI NC block	0	0	1	0
Automatic	1	0	1	0
Program block	0	1	1	0
Single step	1	1	1	0
reserved	0	0	0	1
reserved	1	0	0	1
Single block	0	1	0	1
Return to path	1	1	0	1
CPL debugger program block	0	0	1	1
CPL debugger automatic	1	0	1	1
Manual set-up workpiece coordinates	0	1	1	1
reserved	1	1	1	1

NC ready NC-O 16.0

The **NC ready** signal is a prerequisite for setting the **cycle start** signal.

Signal level

1: Is set when:

- a program has been selected or, while in MDI mode, a block activated but execution of this program or block has not started yet,
- execution of a block in operating modes program block, single block or single step is finished but execution of the next block has not started yet,
- execution of a program in operating modes automatic, program block, single block or single step is finished,
- NC functions **M0**, or **M1**, ’Program stop’, were executed, or
- cancel distance to go (NC-I 3.3) has been executed.

0: No NC program has been selected and no NC block entered. Also, the NC ready signal is reset if execution has been initiated with cycle start and the program running signal has been set.
Cycle start must not be set while NC ready is reset.
There is only one exception, i.e. in combination with **feed hold**. If the feed hold signal has been set previously, the NC ready signal will not be output. However, the machining program can be restarted with cycle start.

Program stop with M0, M1 NC-O 16.1

The **program stop** signal is output when the NC functions conditional and unconditional **program stop** are carried out. **NC ready (NC-O 16.0)** is set simultaneously. If the program is not started again before the end of the block (**cycle start NC-I 1.0)** then the program is stopped and must be restarted with **(cycle start NC-I 1.0)**.

Signal level

1: Is set when:

- an NC block becomes active with **program stop (M0)**,
- an NC block becomes active with **conditional program stop (M1)** and the signal **optional stop (NC-I 1.6)** is set simultaneously.

0:

- a program is restarted with cycle start (NC-I 1.0) after program stop (M0, M1),
- a program is deselected.

Program end with M2, M30 NC-O 16.2

The **program end with M2, M30** signal is output at the end of the main program.

Signal level

1: Is set when execution of the NC block is finished with M2 or M30 in the main program or when the main program has been fully executed, i.e. it has come to an end without M2 or M30.

0: Upon cycle start (NC-I 1.0) at the beginning of a program or when a program is deselected the signal is reset.

Optional stop activated NC-O 16.3

Exists in conjunction with the **M1** function programmable in the NC part program and the input signal **optional stop.**

Signal level

1: Is set when the **optional stop** softkey is actuated on the NC control panel. The signal remains set until the optional stop softkey is pressed again.
The NC-internal function **optional stop** is not yet activated by actuating the softkey; to activate the function, the input signal **optional stop** must be set additionally. The simplest way to achieve this is to feed back the output signal **activate optional stop** to the input signal **optional stop.**
During execution of the NC part program, the program execution is then interrupted in the block in which the function **M1** has been programmed. Program execution can be continued by setting cycle start.

0: The **optional stop** softkey was not activated.

Ready for re-entry NC-O 16.4

Signal level

1: In the ”automatic return to path” mode with NC-I 3.1 set to ”return to contour”, you can return to the saved contour using cycle start. In all other cases, machining is continued with the next cycle start.

0: No automatic return to path or continuation of machining possible.

Re-entry active NC-O 16.5

Signal level

1: Automatic return to the saved contour has been started with a cycle start and the point of return to path has not been reached yet.

0: Automatic return to path has not been started yet or is already finished.

Remove finished NC-O 16.6

Signal level

1: With the leading edge of NC-I 3.1, ”return to contour”, or by pressing the ”return to path” softkey, the motion in departing from the contour is finished. Any subsequent traversing motions will not be recorded.

0: Departing motions from the contour are recorded up to a maximum number of blocks (MACODA parameter 3080 00102, ”Maximum records return to path in blocks”).

Activate block skip NC-O 17.0

Exists in conjunction with the character for **block skip (/)** programmed in the NC part program and the interface signal **block skip.**

Signal level

1: Is set when the softkey **block skip** was actuated on the NC control panel. The signal remains set until the block skip softkey is pressed again.
The NC-internal function **block skip** is not yet activated by actuating the softkey; to activate the function, the input signal block skip must be set. This can be easily achieved by feeding back the output signal **activate block skip** to the input signal **block skip.**
All NC blocks in the part program identified with a slash **/** are then skipped.

0: The **block skip** softkey was not activated.

Active channel status NC-O 17.3 .. 17.7

Depending on the operating mode, the following channel statuses are output in binary code:

Mode	Channel status	Output signal				
		Bit 4 NC-O 17.7	Bit 3 NC-O 17.6	Bit 2 NC-O 17.5	Bit 1 NC-O 17.4	Bit 0 NC-O 17.3
Inactive	Mode is not active and a process may be selected.	0	0	0	0	1
Ready	Mode is ready and the process may be started.	0	0	0	1	0
Running	Mode is active and executing a program or NC block.	0	0	0	1	1
Internal/reserved	reserved	0	0	1	0	0
Internal/reserved	reserved	0	0	1	0	1
Error	An error has occurred in the mode which can be cleared only by ”control reset” or ”program deselection”.	0	0	1	1	0
Internal/reserved	reserved	0	0	1	1	1
Control reset active	Control reset is being executed on the channel. No jobs may be accepted until the status switches to ”inactive”.	0	1	0	0	0
Program selection running	A program has been selected and is being prepared (e.g. linked).	0	1	0	0	1
Cancel distance to go, in preparation	”Cancel distance to go” has been initiated. Clearing is going on.	0	1	0	1	0
Cancel distance to go, running	Mode is active and rearranges existing buffers.	0	1	0	1	1
Ready for program start	Mode is ready, the process is at the beginning of the program and may be started.	0	1	1	0	0
Ready for next block	All the blocks of the retentive NC block input have been executed. Waiting for the next input.	0	1	1	0	1

G131 Tool rotation NC-O 18.0

The **NC-O 18.0 G131 tool rotation** signal indicates that the current angle between 2 traversing blocks has exceeded the contour knee angle specified in G131 G function (tangential tool guidance; see also NC-I 3.2 tool rotation release).

Before executing the internally generated intermediate block, the NC waits until the PLC has set the acknowledgment **NC-I 3.2 G131 tool rotation release**.

The **NC-O 18.0 G131 tool rotation** signal remains set until the execution of the intermediate block is completed.

Signal level

1: The angle between two block exceeds the contour knee angle. The NC waits for the PLC to set the release or it is executing an intermediate block.

0: The tool rotation does not exceed the contour knee angle. No intermediate block waiting for the release to be set is currently present in the NC.

Override 0% NC-O 18.3

The **override 0%** signal acts in conjunction with the **Override** function. The various percentage rates of **impacting the feedrate** can be set in MACODA parameter 7030 00010.

Signal level

1: The override for the channel is set to 0%, and NC-I 1.2 (override 100%) has **not** been set.
A programmed G63, if any, has no effect on this signal. For G63, the NC-O 20.7 is provided.

0: Override has not been set to 0%, or Override 100% has been selected via the interface.

Override 100% NC-O 18.4

The **override 100%** signal acts in conjunction with the **Override** function.

The various percentage rates of **impacting the feedrate** can be set in MACODA parameter 7030 00010.

Signal level

1: The override for the channel is set to 100%, or **override 100%** has been set at the interface.
A programmed G63, if any, has no effect on this signal.

0: Override has not been set to 100%, nor is the **override 100%** interface signal active.

Inpos range 2 active NC-O 18.7

The **Inpos range 2 active** signal indicates that the **G165** function is active as exact positioning mode, and that the positioning window rough is evaluated for all axes assigned to the channel. The **positioning window rough** is specified individually for each axis with SERCOS ident. no. S-0-0261.

If **G164** or **G166** is active as exact positioning mode, the **Inpos range 2 active** interface signal is cancelled, and the **positioning window** with the SERCOS ident. no. S-0-0057 is used for generating the **axis in position** axis interface signals.

Signal level

1: The **positioning window rough** is active for all axes assigned to the channel.

0: The ”normal” **positioning window** is active for all axes assigned to the channel.

External tool correction 2^0 ... 2^3 NC-O 19.0 ... 19.3

These signals indicate the external tool compensation (G145 ... G845) currently **active** on the channel.

External tool correction	**Output signal**			
	Bit 0 **NC-O 19.0**	**Bit 1** **NC-O 19.1**	**Bit 2** **NC-O 19.2**	**Bit 3** **NC-O 19.3**
no compensation active (G146)	0	0	0	0
G145 active	1	0	0	0
G245 active	0	1	0	0
G345 active	1	1	0	0
G445 active	0	0	1	0
G545 active	1	0	1	0
G645 active	0	1	1	0
G745 active	1	1	1	0
G845 active	0	0	0	1

External zero offset 2^0 ... 2^1 NC-O 19.4 ... 19.5

These signals indicate the external zero shift (G160 ... G360) currently **active** on the channel.

External zero offset	Output signal	
	Bit 0 **NC-O 19.4**	**Bit 1** **NC-O 19.5**
no external zero offset active (G167)	0	0
G160 active	1	0
G260 active	0	1
G360 active	1	1

Rapid traverse active NC-O 20.0

The **rapid traverse active** signal is set if rapid traversing is active modally (G0, G10, G200).

Signal level

1: The signal is output for as long as G0 (rapid traverse), G10 (rapid traverse with polar coordinate programming) or G200 is active.

0: G0, G10 or G200 are not active modally.

G70 active NC-O 20.1

This is set if G70 has been selected on the respective channel. The values programmed for the linear synchronous axes are then interpreted as being set in inches.

Signal level

1: G71 is active (metric programming).

0: G70 is active (inch programming).

G32 active NC-O 20.2

The signal G32 indicates that the tapping function is active.

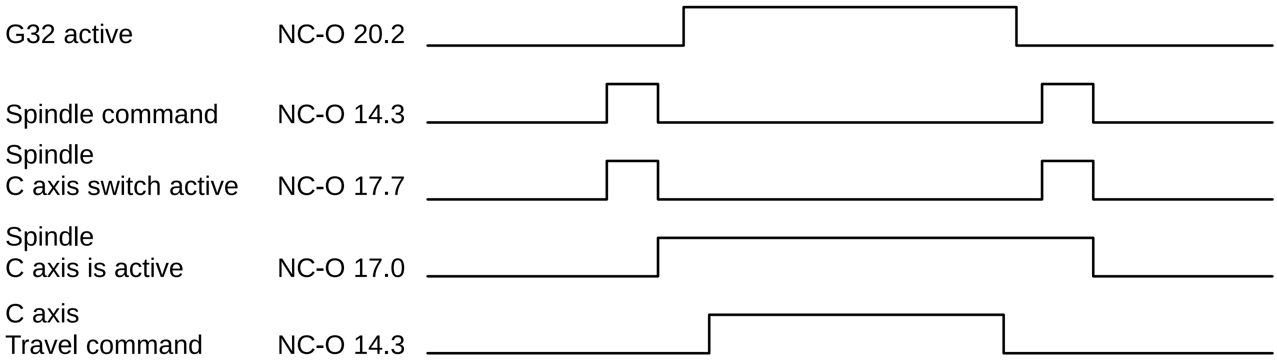
Signal level

1: Tapping (G32) is active.

0: Tapping is not active.

Example:

The sequence of the most important signals for tapping can be illustrated in an example.



G33 active NC-O 20.3

Is set in case of active thread cutting (G33).

Signal level

1: G33 active. The feedrate is dependent on the spindle speed and the thread pitch.

The following input signal reacts **differently**:

- NC-I 1.1 control reset:
  If configured and released, triggers a rapid retraction (see NC-I 7.4 rapid retraction).

The following input signals have **no effect**:

- NC-I 1.2 1st override 100%
- NC-I 4.0 .. 4.4 (4.7) 1st override
- NC-I 8.0 .. 9.7 feed inhibit general
- NC-I 10.0 .. 11.7 feed hold

The following interface signal must **not** be set:

- NC-I 3.3 cancel distance to go

0: G33 is not active.

Thread cycle active NC-O 20.4

This signal can be set from a part program using the G533 TClx signal.

☞ **End of main program (M30) or control reset cancel the signal.**

Signal level

1: G533 TCl1 is programmed.

0: G533 TCl0 is programmed or no program is active.

G41 active NC-O 20.5
G42 active NC-O 20.6

The signals **G41** and **G42** indicate that the respective function is active modally on the channel. G41 and G42 mutually exclude each other in programming, i.e. only one of these signals can be active at any time. If none of the two signals has been set, no cutter radius compensation has been selected, or G40 is active.

Signal level

1: G41 or G42 is active.

0: G41 and G42 are not active.

G63 active NC-O 20.7

The **G63 active** signal indicates that the set **override 100%** function was programmed for the channel and is applied modally. The override 100% interface signal has no effect on the **G63 active signal**. If **G63 active** is not set, the G66 function is active on the channel, which in turn can be superimposed by the override 100% interface signal.

Signal level

1: The G63 function is active.

0: The G63 function is not active.

G92 active NC-O 21.3

The **G92 active** signal indicates that a **G92 shift** is active on the channel. The G92 setting is selected by control reset or by programming G92 without any axis addresses.

Signal level

1: A G92 shift is active.

0: A G92 shift is not active.

G95 active NC-O 21.4

The **G95 active** signal indicates that the **feed programming in mm/revolution** was programmed for the channel and is active modally. If the signal is not set, either G94 or G93 is active.

Signal level

1: The G95 function is active.

0: The G95 function is not active.

Constant cutting velocity active NC-O 21.5

The **constant cutting velocity active** signal indicates whether the spindle on the channel traverses at constant cutting speed (G96 or G196).

Signal level

1: Constant cutting velocity is active.

0: Direct speed programming for spindle is active.

G189 active NC-O 21.0
G90 active NC-O 21.1
G91 active NC-O 21.2

These signals indicate whether the respective function is active modally on the channel. G90, G91, G189 mutually exclude each other in programming, i.e. only one of these signals can be active at any time.

Signal level

1: The respective function is active.

0: The respective function is not active.

Coordinate coupling active NC-O 21.6

The **coordinate coupling active** signal indicates that the respective channel is coupled to a master channel.

Signal level

1: The coordinate coupling is active.

0: The coordinate coupling is not active.

General tool correction 2^0 ... 2^3 NC-O 23.0 ... 23.3

These signals indicate the general tool compensation function (G147 ... G847) currently **active** on the channel.

General tool correction	**Output signal**			
	Bit 0 **NC-O 23.0**	**Bit 1** **NC-O 23.1**	**Bit 2** **NC-O 23.2**	**Bit 3** **NC-O 23.3**
no compensation active (G148)	0	0	0	0
G147 active	1	0	0	0
G247 active	0	1	0	0
G347 active	1	1	0	0
G447 active	0	0	1	0
G547 active	1	0	1	0
G647 active	0	1	1	0
G747 active	1	1	1	0
G847 active	0	0	0	1

CPL customer output (1-16) NC-O 24.0 ... 25.7

The **CPL customer output** signals (1-16) can be written and read directly from the CPL part program with the **IC...** CPL command (also refer to CPL programming manual).

6 Axis-related interface signals

6.1 Overview of axis-related interface signals

The axis-related interface signals act on the NC functions which are associated with the control of a machining or auxiliary axis.
The number of axis interfaces available corresponds to the index of the individual parameter (identical with the physical axis or drive index) of the last axis or C axis/spindle entered in MACODA parameter 1001 00001. The axis interface of a C axis is evaluated or written to by the NC during C axis operation only.

When axes had been suppressed or ”simple” spindles had been configured before the last axis or C axis/spindle defined, an axis-related interface is mapped to the corresponding markers for each of these axes; however, these interfaces will not be served by the NC!

The start addresses for the axis-related interface signals are defined axis-specifically in MACODA parameter 2060 00005.

The following description refers to the interface signals for one axis.

6.1.1 Overview of input signals NC-I (PLC –> NC)

NC-I	Signal designation	NC-I	Signal designation
0.0	Manual +	1.0	Manual feed/incremental bit 0
0.1	Manual –	1.1	Manual feed/incremental bit 1
0.2	Axis override 100%	1.2	Manual feed/incremental bit 2
0.3	Inch incr. step	1.3	Manual feed/incremental bit 3
0.4	Diameter incr. Step	1.4	Travel limit range bit 0
0.5	Handwheel direction of rotation	1.5	Travel limit range bit 1
0.6	res.	1.6	Mode selection bit 0
0.7	res.	1.7	Mode selection bit 1
2.0	Handwheel assignment bit 0	3.0	Axis is discharged
2.1	Handwheel assignment bit 1	3.1	Control reset (auxiliary axis)
2.2	res.	3.2	Suppress software limit switches
2.3	res.	3.3	Cancel fixed stop
2.4	res.	3.4	res.
2.5	Traverse to next grid position (Hirth axis)	3.5	res.
2.6	res.	3.6	res.
2.7	res.	3.7	Torque reduction

NC-I	Signal designation	NC-I	Signal designation
4.0	Override bit 0	5.0	res.
4.1	Override bit 1	5.1	res.
4.2	Override bit 2	5.2	res.
4.3	Override bit 3	5.3	res.
4.4	Override bit 4	5.4	res.
4.5	Coupling lag: NC error OFF	5.5	res.
4.6	Standstill torque: NC error OFF	5.6	res.
4.7	Gantry slave to master position	5.7	res.
6.0	Customer input 1	7.0	Hold command position
6.1	Customer input 2	7.1	res.
6.2	Customer input 3	7.2	res.
6.3	Customer input 4	7.3	res.
6.4	Customer input 5	7.4	res.
6.5	Customer input 6	7.5	Drive inhibit, general
6.6	Customer input 7	7.6	Drive OFF, general
6.7	Customer input 8	7.7	Feed inhibit, general
8.0	Drive inhibit 1	9.0	Drive inhibit 9
8.1	Drive inhibit 2	9.1	Drive inhibit 10
8.2	Drive inhibit 3	9.2	Drive inhibit 11
8.3	Drive inhibit 4	9.3	Drive inhibit 12
8.4	Drive inhibit 5	9.4	Drive inhibit 13
8.5	Drive inhibit 6	9.5	Drive inhibit 14
8.6	Drive inhibit 7	9.6	Drive inhibit 15
8.7	Drive inhibit 8	9.7	Drive inhibit 16
10.0	Drive off 1	11.0	Drive off 9
10.1	Drive off 2	11.1	Drive off 10
10.2	Drive off 3	11.2	Drive off 11
10.3	Drive off 4	11.3	Drive off 12
10.4	Drive off 5	11.4	Drive off 13
10.5	Drive off 6	11.5	Drive off 14
10.6	Drive off 7	11.6	Drive off 15
10.7	Drive off 8	11.7	Drive off 16
12.0	Feed inhibit 1	13.0	Feed inhibit 9
12.1	Feed inhibit 2	13.1	Feed inhibit 10
12.2	Feed inhibit 3	13.2	Feed inhibit 11
12.3	Feed inhibit 4	13.3	Feed inhibit 12
12.4	Feed inhibit 5	13.4	Feed inhibit 13
12.5	Feed inhibit 6	13.5	Feed inhibit 14
12.6	Feed inhibit 7	13.6	Feed inhibit 15
12.7	Feed inhibit 8	13.7	Feed inhibit 16

6.1.2 Overview of output signals NC-O (NC –> PLC)

NC-O	Signal designation	NC-O	Signal designation
14.0	Sercos system ready	15.0	Axis inhibited (test)
14.1	Drive ready	15.1	Axis override 0 %
14.2	Drive under control	15.2	Axis override 100 %
14.3	Travel command	15.3	Reset executed (auxiliary axis)
14.4	Traversing direction –	15.4	res.
14.5	Axis running	15.5	res.
14.6	Axis in position	15.6	Reference point was reached
14.7	res.	15.7	Reference point is known
16.0	Index of the master axis bit 0	17.0	Move to fixed stop active
16.1	Index of the master axis bit 1	17.1	Fixed stop reached
16.2	Index of the master axis bit 2	17.2	Axis near endpoint
16.3	Index of the master axis bit 3	17.3	Error diagnostics class 1
16.4	Index of the master axis bit 4	17.4	Class 2 diagnostics changed
16.5	Axis in grid position (Hirth axis)	17.5	Class 3 diagnostics changed
16.6	Coupling lag exceeded	17.6	Torque limit
16.7	Standstill torque exceeded	17.7	Drive-controlled interpolation
18.0	Axis position 1	19.0	Channel number bit 0
18.1	Axis position 2	19.1	Channel number bit 1
18.2	Axis position 3	19.2	Channel number bit 2
18.3	Axis position 4	19.3	Channel number bit 3
18.4	Axis position 5	19.4	res.
18.5	Axis position 6	19.5	res.
18.6	Axis position 7	19.6	res.
18.7	Axis position 8	19.7	res.
20.0	Customer output 1	21.0	SERCOS drive status bit 0
20.1	Customer output 2	21.1	SERCOS drive status bit 1
20.2	Customer output 3	21.2	SERCOS drive status bit 2
20.3	Customer output 4	21.3	SERCOS drive status bit 3
20.4	Customer output 5	21.4	SERCOS drive status bit 4
20.5	Customer output 6	21.5	SERCOS drive status bit 5
20.6	Customer output 7	21.6	SERCOS drive status bit 6
20.7	Customer output 8	21.7	SERCOS drive status bit 7
22.0	res.	23.0	res.
22.1	res.	23.1	res.
22.2	res.	23.2	res.
22.3	res.	23.3	res.
22.4	res.	23.4	res.
22.5	res.	23.5	res.
22.6	res.	23.6	res.
22.7	res.	23.7	res.
24.0	res.	24.0	res.
24.1	res.	25.1	res.
24.2	res.	25.2	res.
24.3	res.	25.3	res.
24.4	res.	25.4	res.
24.5	res.	25.5	res.
24.6	res.	25.6	res.
24.7	res.	25.7	res.

NC-O	Signal designation	NC-O	Signal designation
26.0	res.	27.0	res.
26.1	res.	27.1	res.
26.2	res.	27.2	res.
26.3	res.	27.3	res.
26.4	res.	27.4	res.
26.5	res.	27.5	res.
26.6	res.	27.6	res.
26.7	res.	27.7	res.

6.2 Signal description

Some signals which are required repeatedly in the context of various machine functions, e.g. **drive inhibit** or **drive OFF**, are available up to 16 times per axis.
This will enable project designers to expand the PLC sequential program step by step without having to change any logic operations of existing function modules.

☞ **In the iPCL and ICL 700 these individual signals are combined to one signal in the APS via an OR-link.**
These sum signals can only be created internally; the PLC program cannot access them.

For purposes of diagnostics, each signal can be shown individually in monitor operation.

6.2.1 Input signals NC-I

Manual +	**NC-I 0.0**
Manual –	**NC-I 0.1**

The manual+ and manual– signals are effective for channel axes (synchronous axes) in the **manual set-up** and **manual approaching the reference point** channel operating modes as well as for **auxiliary axes** (asynchronous axes) in the **axis operating modes with the same designations**.
In addition, the signals are used for manual jogging of synchronous axes when **moving away from the contour**. The active part program is stopped with feed hold in this context.

Signal level logic 1

Continuous jogging:

If the channel or axis operating mode **manual set-up** has been set and all other prerequisites have been met (feed hold, feed inhibit, etc. have not been set), **manual+ or manual–** will move the axis in the respective direction with the manual feed selected via the axis interface (**NC-I 1.0...1.3**). The acceleration of the axis is parametrized in MACODA parameter 1010 00006.
At the axis interface, the **travel command** and **axis running** output signals are set and the **traversing direction–** signal is set in accordance with the direction of movement. The axis in position signal is cancelled at the same time.

Incremental jogging:

The same as above, but the axis travels the **incremental step** preselected via the axis interface. The axis continues to move even when the manual+ or manual– signal is no longer active and the movement along the preselected path has not been completed. The axis moves along the path with the speed entered in MACODA parameters 1005 00007 (defined steps) or 1005 00008 (variable steps).
A new movement can only be triggered if manual+ or manual– previously had signal level logic 0 (edge triggering).

Approaching the reference point:

If the channel or axis operating mode **manual approaching the reference point** has been set and all other prerequisites have been met, drive-controlled referencing is triggered for the axis using manual+ or manual–. The drive's behavior is determined in the SERCOS referencing parameter (ID no. 147).

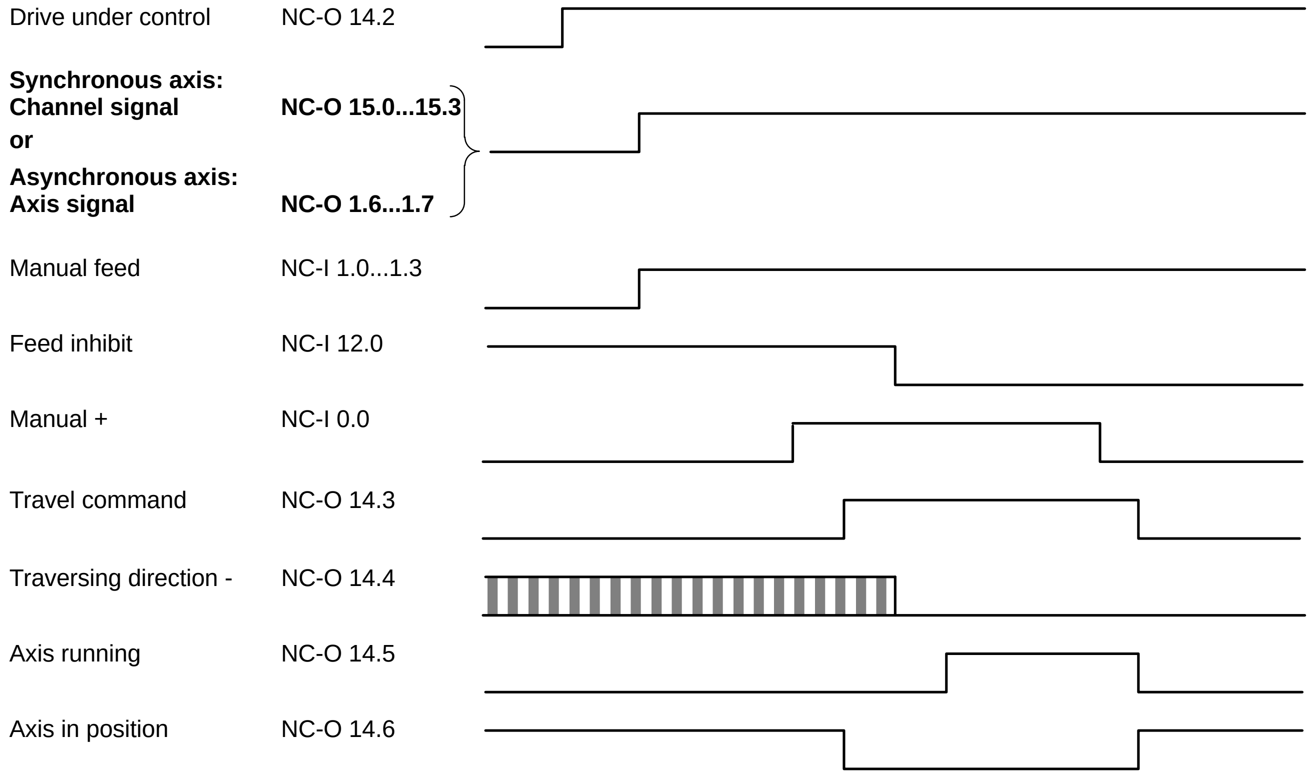
Signal level logic 0

Continuous jogging:

When the signal manual + or manual – is cancelled, the traversing movement is stopped with the deceleration set in MACODA. The travel command is reset. As soon as the axis is located in the in-position window, the axis running signal is reset and axis in position is output.

Example:

Moving an axis (synchronous or asynchronous) in positive direction in **manual** operating mode.


Drive under control NC-O 14.2
Synchronous axis:
Channel signal **NC-O 15.0...15.3**
or
Asynchronous axis:
Axis signal **NC-O 1.6...1.7**
Manual feed NC-I 1.0...1.3
Feed inhibit NC-I 12.0
Manual + NC-I 0.0
Travel command NC-O 14.3
Traversing direction - NC-O 14.4
Axis running NC-O 14.5
Axis in position NC-O 14.6

The control unit has been switched on, the control loop has been closed and torque has been applied on the drive (**drive under control** signal output).

Manual mode must be active on the respective channel in order to move a machining axis **manually** in a positive or negative direction. **Manual** mode is set in coded format by the signals NC-O 15.0 ... NC-O 15.3 being output at the channel-related interface.
For auxiliary axes, the mode selection with NC-I 1.6 ... 1.7 must have been set accordingly. **Manual feed** or the **incremental step** is set via the NC inputs 1.0 ... 1.3 in coded format.

When the **manual +** signal is set, the **travel command** and **traversing direction** signals are output.
When the **travel command** is output, the **axis in position** signal is reset.

No traversing movement is executed until **feed inhibit** is reset. Once the feed inhibit signal is reset, the axis moves as long as manual + is actuated or until the specified **incremental step** has been executed. While the axis is traversing, the **axis running** signal is output. If feed inhibit was reset at an earlier point in time, the axis will start moving as soon as the manual + signal is set.
The **travel command** signal remains set until the traversing movement has been completed. When the axis has come to a standstill, the **axis running** signal is reset and the **axis in position** signal is output if the axis is within the **in-position window**. The **traversing direction** signal is not reset until the next travel command in the opposite direction is output.

Override 100% (auxiliary axis) NC-I 0.2

The override 100% signal (auxiliary axis) is used to suppress the **axis override** of an asynchronous axis. In this case, the axis traverses at the preselected or programmed velocity. The signal refers to programmed asynchronous movements/external setpoint inputs as well as to manual mode (jogging). The signal has no effect on movements interpolated by the drive (e.g. traverse to reference point).

Signal level logic 1

Axis override is disabled.

Signal level logic 0

Axis override is enabled.

Inch incr. step NC-I 0.3

Setting the axis-related interface signal **Inch incr. step** allows jogging by “incremental steps in inches” or traversing with the handwheel while in manual mode. The incremental step specified via the interface is interpreted in inches.

The inch increment of an axis results from the multiplication of MACODA parameter 1015 00001 [axis] in inches by the factor 0.0001.

☞ **As regards rotary and endless axes, the interface signal ”Inch incr. step” makes no difference.**

Example: MACODA parameter = 1

1 incremental step in inch = 0.0001 inch

1 incremental step in metric units = 0.001 mm

Signal level logic 1

In jog mode or handwheel mode, the incremental step specified is interpreted in “inches”.

Signal level logic 0

In jog mode or handwheel mode, the incremental step specified is interpreted in “metric”.

Diameter incr. Step NC-I 0.4

Setting the axis-related interface signal incremental step as diameter allows jogging by ”diameter increments” or traversing with the handwheel while in manual mode. The incremental step also specified via the interface is then interpreted as diameter difference.
The unit of measurement of the increments is specified by the interface signal Inch incr. step NC-I 0.3.

Signal level logic 1

In jog mode or handwheel mode, the incremental step specified is interpreted as diameters difference.

Signal level logic 0

In jog mode or handwheel mode, the incremental step specified is interpreted as radius difference.

Handwheel direction of rotation NC-I 0.5

The handwheel direction of rotation signal allows you to specify the direction of axis rotation, regardless of the direction of handwheel rotation.

Signal level logic 1

The direction of axis rotation is opposite to the direction of handwheel rotation.

Signal level logic 0

The directions of axis and handwheel rotation are the same.

Manual feed/incremental step 2^0...2^3 NC-I 1.0...1.3

Effective in **manual** operating mode. If the axes are traversed manually with the jog button or handwheel, the **feedrates** or **incremental steps** which are assigned to the corresponding bit combination become active. Feedrates are defined in MACODA parameters 1005 00002 .. 1005 00008.

Furthermore, the **incremental steps** are important when the axes are traversed with the handwheel. In this context the **incremental steps** are evaluated as the **weighting factor** for the scale divisions of the handwheel.

The incremental steps for **x increments** are defined in MACODA parameter 1015 00002.

Manual feed/ incremental step	Input signal			
	Bit 3 NC-I 1.3	Bit 2 NC-I 1.2	Bit 1 NC-I 1.1	Bit 0 NC-I 1.0
No input	0	0	0	0
Rapid	0	0	0	1
Fast	0	0	1	0
Medium	0	0	1	1
Slow	0	1	0	0
x increments	0	1	0	1
1000 increments	0	1	1	0
100 increments	0	1	1	1
10 increments	1	0	0	0
1 increment	1	0	0	1

Travel limit range 2^0 ... 2^1 NC-I 1.4 ... 1.5

Selects one out of four possible limit switch ranges.

Bit combination 0: Travel limit range 1 active
Bit combination 1: Travel limit range 2 active
Bit combination 2: Travel limit range 3 active (not yet available)
Bit combination 3: Travel limit range 4 active (not yet available)

☞ **The NC input signal suppress software limit switches (NC-I 3.2) has effect on the selected travel limit range.**

Mode selection NC-I 1.6...1.7

The **mode selection** signal serves to define whether the auxiliary axes are to **jog** or **traverse to reference point** when manual +/– is actuated (provided that the axis has not been assigned to a channel by programming, i.e. it must not move). The selected modes of the channels are of no importance because asynchronous axes are not assigned to a specific channel.

Operating modes are coded as follows:

Operating modes	Mode selection	
	Bit 1 NC-I 1.7	Bit 0 NC-I 1.6
Programming	0	0
Manual set-up	0	1
Manual traverse to reference point	1	0
reserved	1	1

Handwheel assignment 2^0 NC-I 2.0

Handwheel assignment 2^1 NC-I 2.1

Two handwheels can be connected to an NC unit. As soon as one of the two signals is set, the axis can be traversed with the selected handwheel. The NC input signals **manual +** and **manual –** have no effect.

If the axis is traversed with the handwheel, the input signals **manual feed/ incremental step 2^0...2^3** are of importance. In this context, the incremental steps are used in the calculation to weight the scale marks of the handwheel.

Handwheel	**Handwheel assignment**	
	Bit 1 **NC-I 2.1**	**Bit 0** **NC-I 2.0**
no handwheel active	0	0
Handwheel 1	0	1
Handwheel 2	1	0
reserved	1	1

Traverse to next grid position NC-I 2.5

With the signal **traverse to next grid position**, the NC can move a Hirth axis to its next grid position. For this purpose, the NC must be in **manual set-up** mode and the Hirth axis must not being activated by **jog mode** at the same time (see also signal NC-O 16.5).

The traversing direction which was active last determines the direction in which the Hirth axis is moved. This direction is output at the axis-related interface signal NC-O 14.4 “traversing direction –”. If a Hirth axis has come to a halt between 2 grid positions (due to an emergency shutdown, e.g.), this signal may be used to move a Hirth axis to a grid position in the traversing direction specified.

Axis is discharged NC-I 3.0

When the **axis is discharged** signal is set, the respective axis is no longer checked within the inpos logic (e.g. G61, G0, etc.). The axis can thus remain in the configuration of a channel, it is only necessary to set the **drive OFF** and **axis is discharged** signals for the axis.

Control reset (auxiliary axis) NC-I 3.1

The **control reset** signal is used to cancel a movement of an asynchronous axis. When the axis has come to a standstill, the **reset executed** output signal (NC-O 15.3) is set.

Suppress software limit switches NC-I 3.2

Suppresses the positive and negative limit switches of the respective axis. In this case no position limit monitoring is active on the NC for this axis.

For as long as the reference point of the axis is not known (cf. NC-O 15.7), the Suppress software limit switches signal is not relevant. The internal checks performed by the NC are always switched off in this case.

Monitoring by the drive is not affected by the **suppress software limit switches** (e.g. ident. no. S-0-0049, position limit value positive; ident. no. S-0-0050 position limit value negative). Other safety-relevant functions must be assumed by the machine and/or PLC when the software limit switches are suppressed.
The **suppress software limit switches** signal is active in all operating modes.

Signal level logic 0

The software limit switches are not suppressed. The NC monitors the position limits as defined in the MACODA parameters in accordance with the travel limit range selected.

Signal level logic 1

The software limit switches are suppressed.

Cancel fixed stop NC-I 3.3

Setting the **cancel fixed stop** signal cancels an **active** fixed stop for an asynchronous axis.

Torque reduction NC-I 3.7

This signal allows the torque limit (**SERCOS ident. no. S-0-0092**) valid in the drive to be changed.

This signal is of particular significance in conjunction with the axis clamping function: With the axis clamped, the position control loop may remain closed while the torque of the drive can be reduced in order to minimized mechanical stress. To a large extent, this prevents mechanical axis displacement when the clamping is applied or released.

The value of the changed torque limit is set individually for each axis via MACODA parameter 1003 00010. Also, it can be programmed via an NC function (G177).

A torque change-over is permissible only when the axis is at a standstill.

If the torque limit is not accepted by the drive (e.g., because ident. no. is write-protected), no error message is displayed. The NC-O 17.6 torque limit signal remains unchanged in this case.

Signal edge

0 → 1: The torque reduction is transmitted to the drive via the SERCOS interface.

1 → 0: The original torque limit as specified in SERCOS ident. no. S-0-0092 is transmitted to the drive.

Override (auxiliary axis) NC-I 4.0 ... 4.4

The traversing speed of asynchronous axes can be influenced by the **override function**.

The **override** input is used to select the valid override weighting level for an auxiliary axis, provided that the override function has not been suppressed via the interface (NC-I 0.2) or the respective MACODA parameter.

The override function has effect on the **speeds programmed** in the part program and on the preselected speeds in **manual mode**.

As a maximum, the override switch has **32 positions**, which are queried in a **binary encoded format** by the NC.
Each binary code is assigned a value with MACODA parameter 7030 00010 of **channel 0 (!)**, which represents the respective speed as a percentage.

Coupling lag: NC error OFF NC-I 4.5

If a specified coupling lag is exceeded, the signal **coupling lag: NC error OFF** suppresses an internal error signal. In this case, it is necessary for a coupling lag monitoring to be programmed in the PLC.

☞ **If a coupling lag limit is exceeded, the signal NC-I 4.5 will be set, notwithstanding axis signal NC-I 2.6!**

Signal level logic 0

No coupling lag limit has been exceeded.

Signal level logic 1

In case a coupling lag limit is exceeded, the NC will **not** create an internal error signal. Exceeding the coupling lag limit is tolerated.

Standstill torque: NC error OFF NC-I 4.6

If a specified standstill torque is exceeded, the signal **standstill torque: NC error OFF** suppresses an internal error signal. In this case, it is necessary for a standstill torque monitoring to be programmed in the PLC.

☞ **If a standstill torque limit is exceeded, the signal NC-I 4.6 will be set, notwithstanding axis signal NC-I 2.7!**

Signal level logic 0

No standstill torque limit has been exceeded.

Signal level logic 1

In case the standstill torque limit is exceeded, the NC will **not** create an internal error signal. Exceeding of the standstill torque limit is tolerated.

Gantry slave to master position NC-I 4.7

Prerequisites:

- Master and slave have absolute encoders.
- Difference of positions less than the maximum following error configured in MACODA.

In case of a Gantry slave with absolute encoder, the command position of the master axis is transferred without any offset as command position of the slave axis.

Application:
In case of cyclically absolute encoders, the axis positions are not known prior to referencing. That is why the distance between the Gantry axes, which is determined from the position actual values provided by the encoders, is maintained until referencing for safety reasons. In case of axes with absolute encoders, the position actual values are also known without referencing. When the encoders have been commissioned, a possible skew can be recognized from the values they supply.

Signal level logic 0

No effect.

Signal level logic 1

The skew of the slave axis is compensated by jerk as soon as the drives are in operation.
If the distance is too great, a servo error will occur.

CAUTION
The signal must only be set when the encoders have been commissioned, otherwise mechanical damage may occur.

Hold command position NC-I 7.0

With the position control loop open, the **hold command position** signal ensures that the actual drive position is not corrected to the setpoint position.

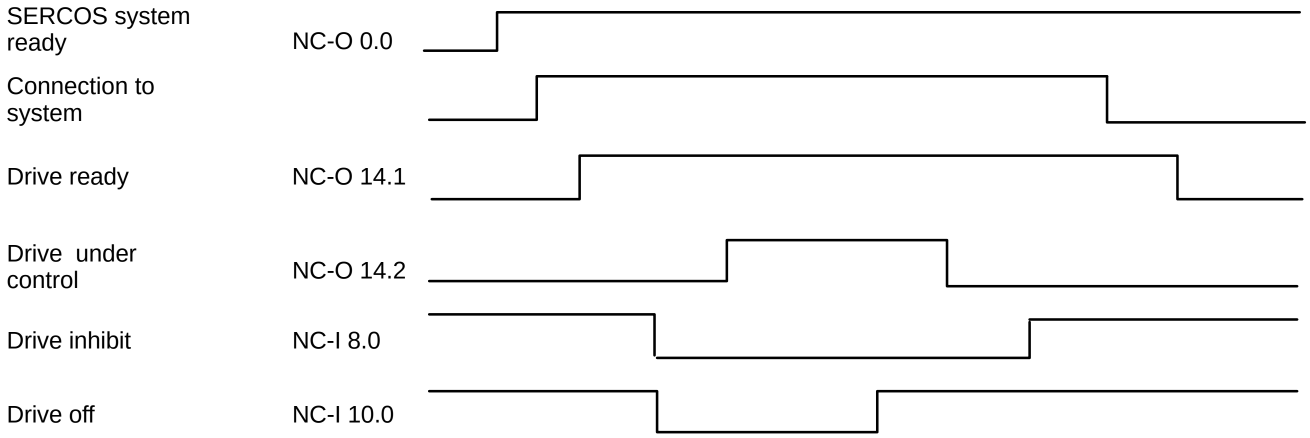
Signal edge

0 → 1: With the position control loop open, the actual drive position is input in the NC.

1 → 0: With the position control loop open, it is prevented that the control unit holds the actual position of the drive.

Drive inhibit	**NC-I 8.0..9.7**
Drive OFF	**NC-I 10.0 ... 11.7**
Sercos system ready	**NC-O 14.0**
Drive ready	**NC-O 14.1**
Drive under control	**NC-O 14.2**

These interface signals affect each other mutually and are of particular significance in the starting/shutdown cycles and the opening/closing of the position control loop.



Example: Functioning of the starting and shutdown cycles

When the machine is started or voltage is applied, the NC unit and the servo and spindle drives go through an initialization phase.
While the initialization phase of the individual components is running, also the digital interface for communication between the NC and the drives is established.

This digital interface corresponds to the settings of the **SERCOS interface.** Initialization of the **SERCOS interface** is in-system, automatic and consists of 4 phases. Once the interface is initialized and cyclic operation is active, the **SERCOS system ready** signal is output.

The **SERCOS system ready** signal enables the respective servo drive to be connected to the system.
When the servo drive is connected, the **drive ready** signal is output.
The position control loop is closed when the signals **drive inhibit** and **drive off are reset** by the PLC sequential program.
The output of the **drive under control** signal indicates that the position control loop is closed and torque is applied on the drive.

☞ **The ”drive OFF” and ”drive inhibit” signals are available 16 times each.**
In the iPCL and ICL 700, these individual signals are merged in the APS by an OR element to form a single signal.

The shutdown cycle or, resp., disconnection from the system, is initiated by the PLC sequential program setting the **drive OFF** signal.

If the axis is still in motion at this time, it is brought to a halt at the preset deceleration.
When the axis has come to a standstill, the position control loop is opened and the **drive under control** signal is reset. No torque is applied on the drive any more.

In the next step, the **drive inhibit** signal can be set and the load can be disconnected.
The **SERCOS system ready** signal remains set for as long as the machine components (NC and drives) stay initialized.

For bringing the drives into circuit again, the procedure described above applies.

Feed inhibit NC-I 12.0 ... 13.7

Feed inhibit is effective when the control loop is closed and torque is applied on the axis. This signal takes effect in all modes. The **feed inhibit** signal is available 16 times in parallel. The effects of the individual signals are identical. The individual signals are linked in the APS by an “OR function”.

Feed inhibit does not affect the signals **travel command, traversing direction +/–** or **axis in position**. Feed inhibit merely prevents the execution of a programmed or manually entered axis movement for as long as **feed inhibit** is set.

Signal level logic 1

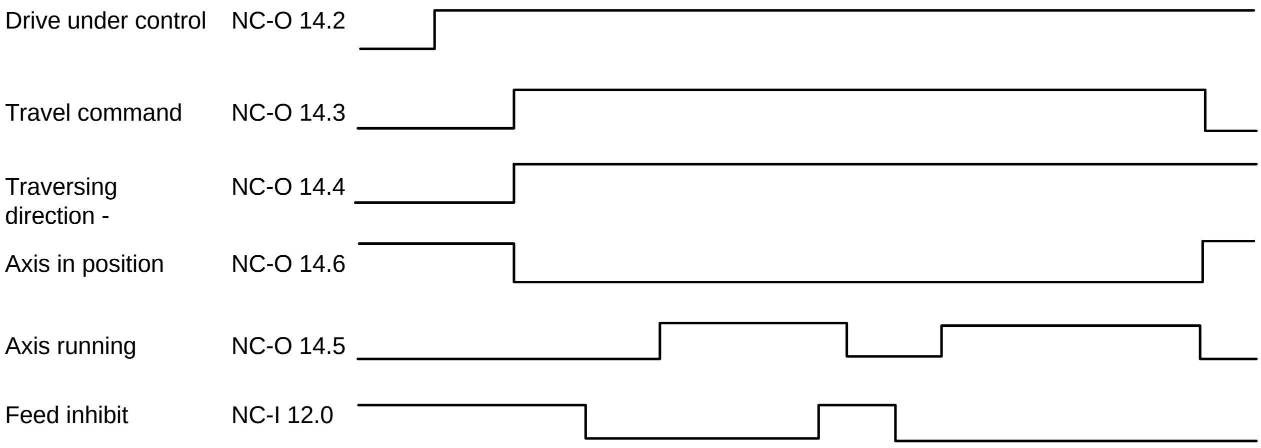
If **feed inhibit** is set by the PLC sequential program, the axis cannot be moved, or, if the axis is currently in motion, it will be stopped at the configured deceleration.
If **feed inhibit** is set for an axis which participates in an interpolation with other axes, the other axes cannot be traversed, either.

Signal level logic 0

Axis feed is not inhibited. If an axis travel command is initiated (via the NC part program or manually), the axis can start traversing right away.
If a traversing axis was brought to a standstill with **feed inhibit**, this axis will start accelerating up to its programmed speed as soon as the **feed inhibit** signal is reset.

Example:

The machine is switched on and the control loop has been closed. A negative traversing motion has been programmed for an axis via the NC part program.



The **drive under control** signal indicates to the PLC sequential program that the control loop is closed.

The **travel command** and **traversing direction** signals are output as soon as the NC block where the traversing motion is programmed is executed. The traversing direction is only up-to-date if a traversing command has been set.

The **axis in position** signal is reset when the travel command is activated.

As long as the **feed inhibit** signal remains set, the traversing movement will not be carried out and the **axis running** signal will not be set.

When feed inhibit is reset, the programmed traversing movement is performed and axis running is set.

If **feed inhibit** is set again during the traversing movement, the traversing movement is stopped and **axis running** is reset when the axis has come to a standstill.

When **feed inhibit** is reset, the axis accelerates up to the previously applied speed, **axis running** is set and the traversing movement is completed.

When the programmed end position is reached, the **travel command** and **axis running** signals are reset and the **axis in position** signal is set.

6.2.2 Output signals NC-O

SERCOS system ready NC-O 14.0

The SERCOS system ready signal is closely related to the signals **drive inhibit, drive OFF, drive ready** and **drive under control** (cf. S-0-0135 bit 14/15 in the SERCOS manual).
The functional relationship is set out in detail in the signal diagram for the **drive inhibit NC-I 8.0...9.7** signal.

Signal level logic 1

SERCOS system ready is output when the initialization phases of the SERCOS interface, the drive and the NC unit have been concluded without any errors occurring.
This signal is prerequisite to bringing the drive into circuit.

Signal level logic 0

The drive cannot be brought into circuit until the SERCOS system ready signal is set.

Drive ready NC-O 14.1

The drive ready signal is closely related to the signals **drive inhibit, drive OFF, SERCOS system ready** and **drive under control** (cf. S-0-0135, bit 14/15 in the SERCOS manual).
The functional relationship is set out in detail in the signal diagram for the **drive inhibit NC-I 8.0...9.7** signal.

Signal level logic 1

The drive ready signal is output when the respective drive has been brought into circuit. The signal remains set until the drive is cut out.

Signal level logic 0

The drive is not brought into circuit.

Drive under control NC-O 14.2

The **drive under control** signal is closely related to the signals **drive inhibit, drive OFF, SERCOS system ready** and **drive ready** (cf. S-0-0135, bit 14/15 in the SERCOS manual).
The functional relationship is set out in detail in the signal diagram for the **drive inhibit NC-I 8.0...9.7** signal.

Signal level logic 1

When the **drive under control** signal is output, the control loop is closed and torque is applied on the drive. The axis can be moved unless it is locked by any other function, e.g. **feed inhibit**.

Signal level logic 0

The control loop is open, no torque is applied on the drive.

Travel command NC-O 14.3

The signals **travel command, traversing direction +/–, axis running** and **axis in position** affect each other.

Examples of their relationships are shown in the signal diagrams and explanations of the signals **manual +** (NC-I 0.0) and **feed inhibit** (NC-I 12.0). The **traversing direction +/–** is output in conjunction with the **travel command.**

Signal level logic 1

The travel command signal is set whenever an axis is to execute a traversing movement. The traversing movement can be carried out by manual input or via the NC part program.

Signal level logic 0

No command for the traversing movement is present.

Traversing direction – NC-O 14.4

The **traversing direction–** signal is effective only in conjunction with the signal **travel command.** In the absence of a travel command signal, the **traversing direction–** signal is irrelevant.

Signal level logic 1

If traversing direction– and travel command are set, the respective axis is to execute a traversing movement in negative direction or that the previous movement of this very axis was executed in negative direction.

Signal level logic 0

If traversing direction– is not set and travel command is set, the respective axis is to execute a traversing movement in positive direction or that the previous movement of this very axis was executed in positive direction.

Axis running NC-O 14.5

The signals **travel command, traversing direction +/–, axis running** and **axis in position** affect each other.

Examples of their relationships are shown in the signal diagrams and explanations of the signals **manual +** (NC-I 0.0) and **feed inhibit** (NC-I 12.0).

If the traversing movement of an axis has been concluded from the NC point of view, no **travel command** is active any more, and the signals **axis in position** or **axis running** will be output.

Signal level logic 1

Axis running remains set for as long as the axis executes a programmed or manually preset movement.
This signal is not set in the case of **drive-controlled traversing to reference point** even if the axis is still in motion.

Signal level logic 0

The relevant axis performs no movement.

Axis in position NC-O 14.6

The signal is active when the control loop is closed (drive under control set) and torque is applied on the axis.

Signal level logic 1

Axis in position is set when the axis is in the parameterized in-position window and no travel command signal is set. The in-position window is specified in SERCOS parameter S-0-0057.
The “in-position window rough” for Servodyn drives is specified in SERCOS parameter S-0-0261, and that for other drives – in MACODA parameter 1015 00100.

Signal level logic 0

The axis is not in the parameterized in-position window.

Axis inhibited (test) NC-O 15.0

The **axis inhibited** signal indicates whether an axis has been switched to test mode. In test mode, an axis is still interpolated internally, however, the calculated setpoints are only shown on the display but are not transmitted to the drive. The axis is held in the same position where it was when test mode was activated. You can switch to test mode by using the softkey or you can switch automatically with drives disengaged.

Override 0% (auxiliary axis) NC-O 15.1

The **override 0%** signal indicates that the axis override of an asynchronous axis has been set to 0% and that no override 100% has been selected via the interface (NC-I 0.2).

Signal level logic 1

0% act on the axis.

Signal level logic 0

A value not equal to 0% acts on the axis.

Override 100% (auxiliary axis) NC-O 15.2

The **override 100%** signal indicates that the axis override of an asynchronous axis has been set to 100% and/or that override 100% has been selected via the interface (NC-I 0.2).

Signal level logic 1

A value of 100% acts on the axis.

Signal level logic 0

A value not equal to 100% acts on the axis.

Reset executed (auxiliary axis) NC-O 15.3

The **reset executed** signal indicates that the asynchronous axis accepted the control reset task and that the asynchronous axis is ready to accept new inputs for motions.

Reference point was reached NC-O 15.6

In contrast to the **reference point is known** signal, the **reference point was reached** signal is set only when traversing to the reference point has been successfully completed. If traversing to the reference point was cancelled, none of the two signals will be set. While a subsequent reset command causes a status check of the actual position values to be performed in the drive (ident. no. S-0-0403) and sets (or clears) the **reference point is known** signal, as applicable, the **reference point was reached** signal remains unchanged.

This signal is reset at the beginning of any traverse to a reference point.

Signal level logic 1

The last traverse to the reference point was properly completed by reaching the reference point.

Signal level logic 0

Either no reference points have been traversed to so far, or traversing to the reference point is still active and the reference point has not yet been reached, or the last traverse to the reference point was cancelled.

Reference point is known NC-O 15.7

The **reference point is known** signal is set as soon as the reference point of the axis has been reached, or, if absolute encoders are used, as soon as they have been recognized by the NC. For this purpose, the NC evaluates the **actual position value status** SERCOS interface signal S-0-0403. Before the **reference point is known** signal is set, no limit switch monitoring is performed by the NC.

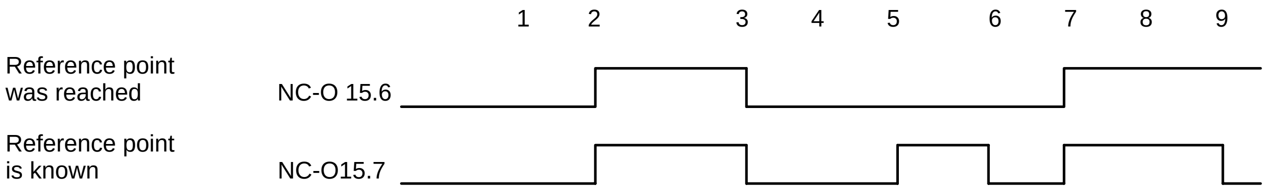
☞ **Once the reference point is known signal has been set it will only be reset when a new SERCOS phase run-up has been initiated, or in the case of a drive fault (e.g. defective encoder).**

Signal level logic 1

The reference point has been reached, or the axis is equipped with absolute encoders.

Signal level logic 0

No reference point has been traversed to yet.



1 Start traversing to reference point
2 Traversing to reference point completed
3 Start traversing to reference point
4 Cancel traversing to reference point
(reference point has already been recognized by the drive)
5 Control reset
6 Start traversing to reference point
7 Traversing to reference point completed
8 Drive lost reference point due to a fault
9 Control reset

Index of the master axis bit 2^0 ... 2^4 NC-O 16.0 ... 16.4

Each axis is given the axis index of its master axis in binary code, and the following applies:

- The index of the master axis must be less than 32.
- If the axis is not coupled, the index of the master axis equals 0.
- If the axis is the master itself, the index of the master axis is identical with the axis index.

Axis in grid position NC-O 16.5

The **axis in grid position** signal indicates that the Hirth axis has reached a grid position.
The level of this signal is 0 as long as no traverse to the reference point has been performed yet. This signal is set to 1 as soon as the axis is within the selected in-position window.

Signal level logic 1

The Hirth axis is within the selected in-position window.

Signal level logic 0

Indicates that the Hirth axis has not traversed to a reference point before.

Coupling lag exceeded NC-O 16.6

The **coupling lag exceeded** signal serves to monitor the coupling lag specified for a group of coupled axes.

Signal level logic 0

The maximum coupling lag specified for an axis has not been exceeded.

Signal level logic 1

The maximum coupling lag specified for an axis has been exceeded.

Standstill torque exceeded NC-O 16.7

The **standstill torque exceeded** signal serves to monitor the standstill torque specified for an axis.

☞ **In the case of Gantry axes, this signal refers to the sum total of the standstill torque values (regardless of the sign) of all the axes of a group of coupled axes. The signal is set only for the master axis.**

Signal level logic 0

The standstill torque specified for an axis has not been exceeded.

Signal level logic 1

The standstill torque specified for an axis has been exceeded.

Move to fixed stop active NC-O 17.0

The **move to fixed stop active** signal is output as long as the traversing movement to the fixed stop is active.

A movement to the fixed stop is triggered by the G375 ”measuring fixed stop” or G475 ”move to fixed stop” functions.

Signal level logic 0

No move to fixed stop active.

Signal level logic 1

Axis moves to fixed stop.

Fixed stop reached NC-O 17.1

The **fixed stop reached** signal is output as long as the fixed stop is active.

The fixed stop is generated by the G475 ”move to fixed stop” function. The fixed stop becomes inactive when the axis leaves the ”MP 1003 00032” monitoring window fixed stop or the fixed stop is cancelled. (G476).

Signal level logic 0

Fixed stop is inactive.

Signal level logic 1

Fixed stop is active.

Axis near endpoint NC-O 17.2

The **axis near endpoint** signal indicates that this axis is near its endpoint in an area set via MACODA parameter 1015 00010.

Signal level logic 0

Axis command position outside of the indicated distance or not configured or traversing to reference point active or handwheel active.

Signal level logic 1

Axis command position within the indicated distance.

Error diagnostics class 1 NC-O 17.3

The **error diagnostics class 1** signal (drive cutoff with interlock) is a direct map of the corresponding signal in the SERCOS drive status word of the axis in question. The interlock can only be canceled by the command ”reset diagnostics class 1”, which is set automatically by the NC upon a **control reset**.

The pending error can be evaluated manually using the SERCOS monitor, or automatically by having the PLC directly read ident. no. S-0-0011.

Ident. no. S-0-0011 comprises 2 bytes and has the following structure:

Bit 0:	reserved	
Bit 1:	Amplifier shut-off temperature	S-0-0203
Bit 2:	Motor shut-off temperature	S-0-0204
Bit 3:	reserved	
Bit 4:	Control voltage fault	
Bit 5:	Feedback error (encoder, measuring system)	
Bit 6:	Error in electronic commutation system	
Bit 7:	Overcurrent	
Bit 8:	Overvoltage	
Bit 9:	Undervoltage fault	
Bit 10:	Phase fault in power supply	
Bit 11:	Excessive controller deviation	S-0-0159
Bit 12:	Communication error	
Bit 13:	Position limit value exceeded	S-0-0049
Bit 14:	reserved	
Bit 15:	Manufacturer-specific error	S-0-0129

Class 2 diagnostics changed NC-O 17.4

The **class 2 diagnostics changed** signal (switch-off prewarning) is a direct map of the corresponding signal in the SERCOS drive status word of the axis in question. A change in diagnostics class 2 sets the bit, which is reset again when the status has been read using ident. no. S-0-0012. The status can be read manually using the SERCOS monitor or automatically by having the PLC directly read ident. no. S-0-0012.

Ident. no. S-0-0012 comprises 2 bytes and contains the following warnings:

Bit 0:	reserved	
Bit 1:	Amplifier overtemperature warning	S-0-0311
Bit 2:	Motor overtemperature warning	S-0-0312
Bit 3:	reserved	
Bit 4:	reserved	
Bit 5:	reserved	
Bit 6:	reserved	
Bit 7:	reserved	
Bit 8:	reserved	
Bit 9:	reserved	
Bit 10:	reserved	
Bit 11:	reserved	
Bit 12:	reserved	
Bit 13:	reserved	
Bit 14:	reserved	
Bit 15:	Manufacturer-specific warning	S-0-0181

The effect of switch-off prewarnings concerning the alteration bit can be suppressed on the screen for **diagnostics class 2** (ident. no. S-0-0097).

Class 3 diagnostics changed NC-O 17.5

The **class 3 diagnostics changed** signal (messages concerning operating statuses) is a direct map of the corresponding signal in the SERCOS drive status word of the axis in question. A change in diagnostics class 3 sets the bit, which is reset again when the status has been read using ident. no. S-0-0013. The status can be read manually using the SERCOS monitor or automatically by having the PLC directly read ident. no. S-0-0013.

Ident. no. S-0-0013 comprises 2 bytes and contains the following messages:

Bit 0:	$n_{act} = n_{set}$	S-0-0330
Bit 1:	$n_{act} = 0$	S-0-0331
Bit 2:	$\|n_{act}\| < \|n_x\|$	S-0-0332
Bit 3:	$\|Md\| \geq \|Md_x\|$	S-0-0333
Bit 4:	$\|Md\| \geq \|Md_{limit}\|$	S-0-0334
Bit 5:	$\|n_{set}\| > \|n_{limit}\|$	S-0-0335
Bit 6:	In Position	S-0-0336
Bit 7:	$\|P\| \geq \|P_x\|$	S-0-0337
Bit 8:	reserved	
Bit 9:	$\|n_{act}\| \geq$ min. spindle speed	S-0-0339
Bit 10:	$\|n_{act}\| \geq$ max. spindle speed	S-0-0340
Bit 11:	In position rough	S-0-0341
Bit 12:	Target position reached	S-0-0342
Bit 13:	Interpolator halted	S-0-0343
Bit 14:	reserved	
Bit 15:	Manufacturer-specific operating statuses	S-0-0182

The effect of the operating statuses concerning the alteration bit can be suppressed on the screen for **diagnostics class 3** (ident. no. S-0-0098).

Torque limit NC-O 17.6

The **torque limit** signal indicates that the axis concerned is running at a reduced maximum torque.

Signal level logic 1

Maximum torque of the axis is reduced.

Signal level logic 0

Maximum torque of the axis is not reduced.

Drive-controlled interpolation NC-O 17.7

The **drive-controlled interpolation** signal indicates that the relevant axis is currently being interpolated directly by the drive.

Signal level logic 1

Axis is interpolated by the drive.

Signal level logic 0

Axis is controlled by the control unit.

Axis position 1 ... 8 NC-O 18.0 ... 18.7

The **axis position** signal serves to monitor the actual position of an axis. An axis may be assigned a maximum of 8 axis monitoring positions (electric limit switches).

Using MACODA parameters 2010 00100 through 2010 00130, you can define up to 64 axis monitoring positions for the whole system.

By **combining** two signals, you can create ”monitoring windows”.
For further information, please see the manual ”Configuration Parameters and MACODA Parameter Description”.

Signal level logic 1

The axis actual position is greater than or equal to the monitoring position.

Signal level logic 0

The axis actual position is smaller than the monitoring position.

Channel number 2^0 ... 2^3 NC-O 19.0 ... 19.3

Using the channel number 2^0 ... 2^3 signals the binary-coded channel number of the corresponding channel is output.

”0” is output if the axis is not assigned to any channel (e.g. asynchronous axis).

Channel number	**Input signal**			
	Bit 3 **NC-O 19.3**	**Bit 2** **NC-O 19.2**	**Bit 1** **NC-O 19.1**	**Bit 0** **NC-O 19.0**
No channel	0	0	0	0
Channel 1	0	0	0	1
...	...	...	...	...
Channel 12	1	1	0	0

If axes are borrowed to another channel, they are output in the following manner:

Status	**Channel information**
Channel is activated (upon program selection or manual data input) with one axis.	Number of this channel
Channel is deactivated (upon program deselection or end of manual data input) with one axis.	Number of this channel
Channel releases axis with G512 while axis has not been released on its original channel.	Number of the original channel
Channel releases axis with G512 and axis has been released on its original channel.	0

Definitions:

- “Borrowed axis”: an axis is borrowed if a channel “accesses an axis of an inactive channel” while the axis was not released on this channel with G512.
- “Original channel”: upon NC start-up, all synchronous axes belong to an original channel.

7 Spindle-related interface signals

7.1 Overview of spindle-related interface signals

Spindle-related interface signals have an effect on those NC functions that are related to the control of a machining spindle.
An interface is available for each projected spindle, i.e. for n spindles the spindle-related interface is also mapped n times on the corresponding marker.

In the case of spindles assigned to a channel, also the channel-related interface is of importance.

The start addresses of spindle-related interfaces are defined in MACODA parameter 2060 00006.

In the following description, the interface signals are outlined for one spindle.

7.1.1 Overview of input signals NC-I (PLC –> NC)

NC-I	Signal designation	NC-I	Signal designation
0.0	Spindle M3 manual	1.0	Spindle speed jog bit 0
0.1	Spindle M4 manual	1.1	Spindle speed jog bit 1
0.2	Spindle M5 manual	1.2	Spindle speed jog bit 2
0.3	res.	1.3	res.
0.4	Spindle jog M3	1.4	res.
0.5	Spindle jog M4	1.5	res.
0.6	res.	1.6	C axis ON
0.7	res.	1.7	C axis OFF
2.0	Override bit 0	3.0	res.
2.1	Override bit 1	3.1	res.
2.2	Override bit 2	3.2	Spindle orientation manual
2.3	Override bit 3	3.3	Spindle override 100%
2.4	Override bit 4	3.4	Reset (spindle)
2.5	res.	3.5	res.
2.6	res.	3.6	res.
2.7	res.	3.7	res.
4.0	res.	5.0	Gear range 1 acknowledge
4.1	res.	5.1	Gear range 2 acknowledge
4.2	res.	5.2	Gear range 3 acknowledge
4.3	res.	5.3	Gear range 4 acknowledge
4.4	res.	5.4	res.
4.5	res.	5.5	res.
4.6	res.	5.6	res.
4.7	res.	5.7	Gear idle motion acknowledge

NC-I	Signal designation	NC-I	Signal designation
6.0	Customer input 1	7.0	res.
6.1	Customer input 2	7.1	res.
6.2	Customer input 3	7.2	res.
6.3	Customer input 4	7.3	res.
6.4	Customer input 5	7.4	res.
6.5	Customer input 6	7.5	Drive inhibit, general
6.6	Customer input 7	7.6	Drive OFF, general
6.7	Customer input 8	7.7	Spindle inhibit, general
8.0	Drive inhibit 1	9.0	Drive inhibit 9
8.1	Drive inhibit 2	9.1	Drive inhibit 10
8.2	Drive inhibit 3	9.2	Drive inhibit 11
8.3	Drive inhibit 4	9.3	Drive inhibit 12
8.4	Drive inhibit 5	9.4	Drive inhibit 13
8.5	Drive inhibit 6	9.5	Drive inhibit 14
8.6	Drive inhibit 7	9.6	Drive inhibit 15
8.7	Drive inhibit 8	9.7	Drive inhibit 16
10.0	Drive OFF 1	11.0	Drive OFF 9
10.1	Drive OFF 2	11.1	Drive OFF 10
10.2	Drive OFF 3	11.2	Drive OFF 11
10.3	Drive OFF 4	11.3	Drive OFF 12
10.4	Drive OFF 5	11.4	Drive OFF 13
10.5	Drive OFF 6	11.5	Drive OFF 14
10.6	Drive OFF 7	11.6	Drive OFF 15
10.7	Drive OFF 8	11.7	Drive OFF 16
12.0	Spindle inhibit 1	13.0	Spindle inhibit 9
12.1	Spindle inhibit 2	13.1	Spindle inhibit 10
12.2	Spindle inhibit 3	13.2	Spindle inhibit 11
12.3	Spindle inhibit 4	13.3	Spindle inhibit 12
12.4	Spindle inhibit 5	13.4	Spindle inhibit 13
12.5	Spindle inhibit 6	13.5	Spindle inhibit 14
12.6	Spindle inhibit 7	13.6	Spindle inhibit 15
12.7	Spindle inhibit 8	13.7	Spindle inhibit 16

7.1.2 Overview of output signals NC-O (NC –> PLC)

NC-O	Signal designation	NC-O	Signal designation
14.0	SERCOS system ready	15.0	Spindle orientation active
14.1	Drive ready	15.1	Spindle is oriented
14.2	Drive under control	15.2	Progr. spindle speed reached
14.3	Spindle command	15.3	Spindle speed 100 %
14.4	Direction of rotation M4	15.4	Idling speed reached
14.5	Gear range switching active	15.5	Spindle stopped
14.6	Spindle in position	15.6	res.
14.7	res.	15.7	res.
16.0	Output gear range 1	17.0	C axis is active
16.1	Output gear range 2	17.1	Spindle speed limited
16.2	Output gear range 3	17.2	Spindle override 0%
16.3	Output gear range 4	17.3	Error diagn. class 1
16.4	res.	17.4	Class 2 diagnostics changed
16.5	res.	17.5	Class 3 diagnostics changed
16.6	res.	17.6	Position control active
16.7	Output gear in idle motion	17.7	C axis switching active
18.0	Index of coupling bit 0	19.0	res.
18.1	Index of coupling bit 1	19.1	res.
18.2	Index of coupling bit 2	19.2	res.
18.3	Spindle is master	19.3	res.
18.4	Synchronous 1	19.4	Control reset acknowledge (Spindle)
18.5	Synchronous 2	19.5	res.
18.6	Coupling error	19.6	res.
18.7	res.	19.7	res.
20.0	Customer output 1	21.0	SERCOS drive status bit 0
20.1	Customer output 2	21.1	SERCOS drive status bit 1
20.2	Customer output 3	21.2	SERCOS drive status bit 2
20.3	Customer output 4	21.3	SERCOS drive status bit 3
20.4	Customer output 5	21.4	SERCOS drive status bit 4
20.5	Customer output 6	21.5	SERCOS drive status bit 5
20.6	Customer output 7	21.6	SERCOS drive status bit 6
20.7	Customer output 8	21.7	SERCOS drive status bit 7
22.0	res.	23.0	res.
22.1	res.	23.1	res.
22.2	res.	23.2	res.
22.3	res.	23.3	res.
22.4	res.	23.4	res.
22.5	res.	23.5	res.
22.6	res.	23.6	res.
22.7	res.	23.7	res.
24.0	res.	25.0	res.
24.1	res.	25.1	res.
24.2	res.	25.2	res.
24.3	res.	25.3	res.
24.4	res.	25.4	res.
24.5	res.	25.5	res.
24.6	res.	25.6	res.
24.7	res.	25.7	res.

NC-O	Signal designation
26.0	res.
26.1	res.
26.2	res.
26.3	res.
26.4	res.
26.5	res.
26.6	res.
26.7	res.

NC-O	Signal designation
27.0	res.
27.1	res.
27.2	res.
27.3	res.
27.4	res.
27.5	res.
27.6	res.
27.7	res.

7.2 Signal description

Some signals which are required repeatedly in the context of various machine functions, e.g. **drive ready** or **drive OFF** are available up to 16 times per spindle.
This will enable project designers to expand the PLC sequential program step by step without having to change any logic operations of existing function modules.

☞ **In the iPCL and ICL 700 these individual signals are combined to one signal in the APS via an OR-link.**
These sum signals can only be created internally; the PLC program cannot access them.

For purposes of diagnostics, each signal can be shown individually in monitor operation.

Depending on their function, NC input signals are either **static signals** with levels logic 0 or logic 1, or they are **edge-triggering signals**, i.e. their signal edges 0→1 or 1→0 are evaluated by the NC.

7.2.1 Input signals NC-I

Spindle M3 manual NC-I 0.0
Spindle M4 manual NC-I 0.1

These signals are alternative to the NC part program (programming M3, M4) in determining the sense of rotation of the spindle:

- Spindle M3 manual: spindle clockwise
- Spindle M4 manual: spindle counter-clockwise

The **spindle M5 manual** signal is significant in conjunction with signals spindle M3 manual and spindle M4 manual (see also example on page 7–6).

Signal edge

0 → 1: Using this edge, a spindle speed and the corresponding sense of rotation is input for the spindle drive. The spindle will start to rotate if the control loop is closed (**drive under control**) and no **spindle inhibit** signal is set.
The spindle speed corresponds to the speed programmed last.

1 → 0: No effect.

Spindle M5 manual NC-I 0.2

This signal is an alternative to the NC part program (programming M5) in activating the spindle stop function.

The signals **spindle M3 manual** and **spindle M4 manual** are significant in conjunction with spindle M5 manual.

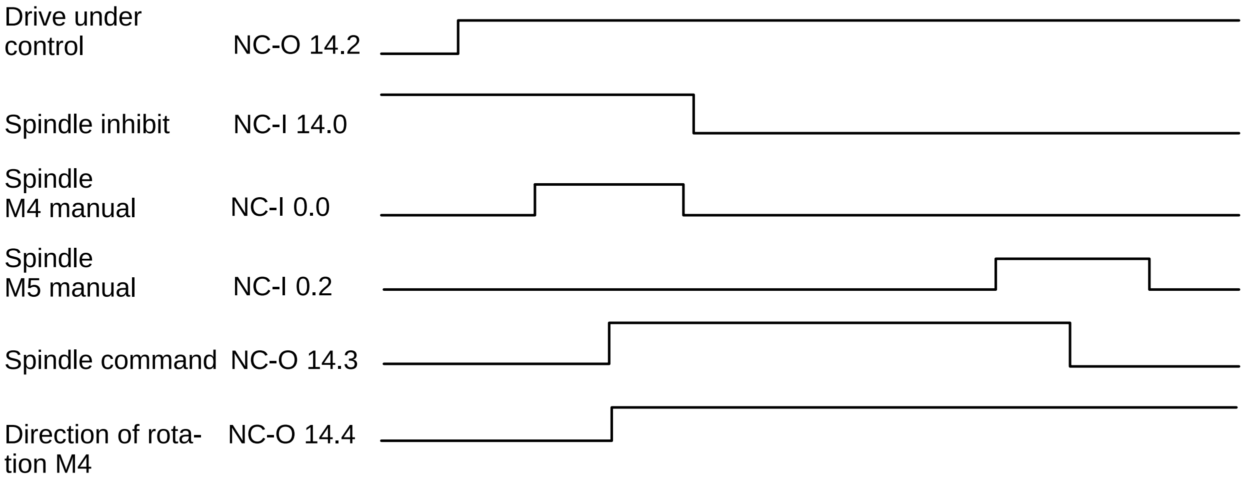
Signal edge

0 → 1: Stops the spindle with the parametrized deceleration. The gear range and spindle speed that were active previously remain stored in the NC. The control loop remains closed and torque is still applied on the spindle drive.

1 → 0: No effect.

Example:

By manual data input, the spindle is to be accelerated in counter-clockwise rotation up to the speed that was programmed last.



The **drive under control** signal indicates that the control loop is closed and torque is applied on the spindle.

When the **spindle M4 manual** signal is set, the speed programmed last is transmitted to the spindle drive.
As a result of this speed input, the NC sets the **spindle command** and **direction of rotation M4** signals.

Upon the output of the **spindle command** signal, the **spindle M4 manual** signal can be reset. The spindle does not rotate while **spindle inhibit** is set.

Only when the **spindle inhibit** signal has been reset will the spindle start rotating and the **spindle rotating** signal is set. By setting the **spindle M5 manual** signal, the rotation of the spindle is stopped and the **spindle command** and **direction of rotation M4** signals are reset by the NC.

Spindle jog M3 NC-I 0.4
Spindle jog M4 NC-I 0.5

The signals spindle jog M3 and spindle jog M4 have an effect in the **manual** operating mode of the NC and are functionally related with the coded NC input signals **spindle speed jog 2^0**...**spindle speed jog 2^2.**

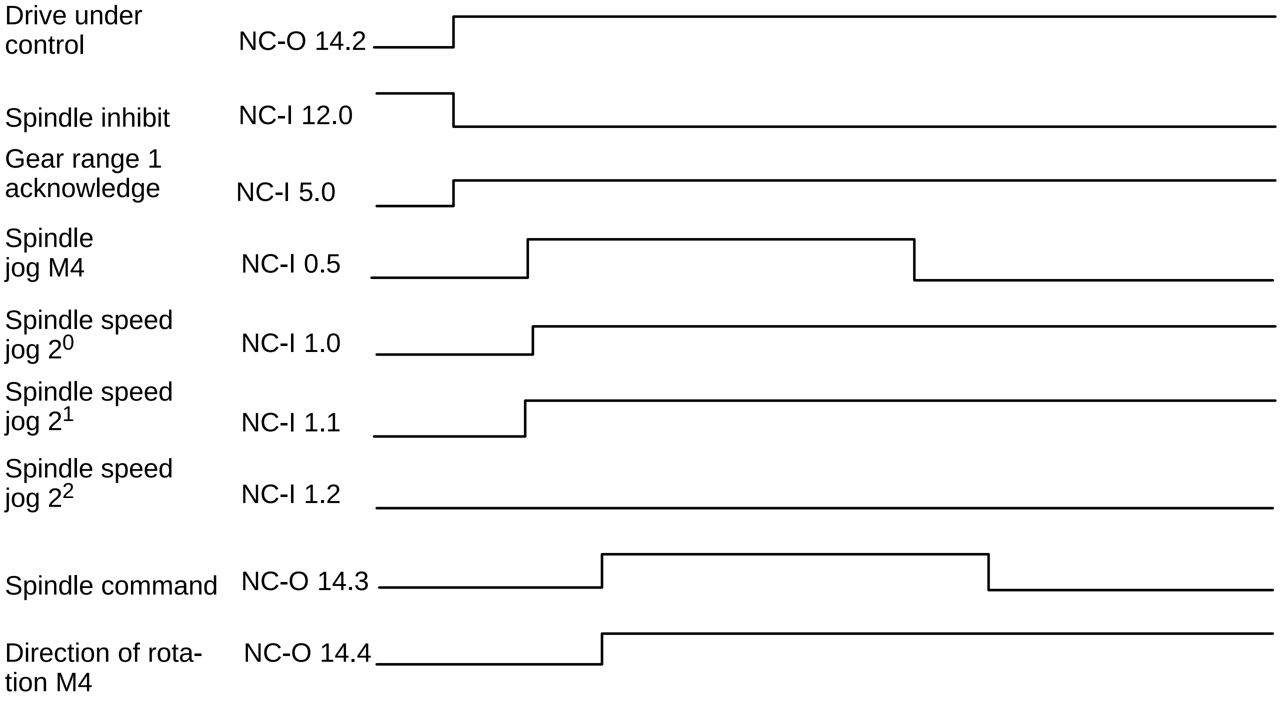
Signal level

1: As a prerequisite, a spindle speed must be input via the coded input signals spindle speed jog 2^0...spindle speed jog 2^2. Furthermore, the control loop must be closed (drive under control set), spindle inhibit must be reset and a gear range must be activated. As long as the **spindle jog M3** or **spindle jog M4** signal is set, the spindle rotates at the respective speed in the selected direction of rotation.

0: The spindle is decelerated until it comes to a standstill and the control loop remains closed.

Example:

In manual mode, the spindle is to be traversed in clockwise direction at a predefined speed.



The control loop is closed (**drive under control** set), **spindle inhibit** is reset and gear range 1 (**gear range 1 acknowledge**) is activated.

At the same time, the **spindle jog M3** signal is set, a spindle speed must be input (**spindle speed jog 2^0...spindle speed jog 2^2**). The jogging speed is defined for each spindle individually in MACODA parameter 1040 00015. The **spindle command** is set by the NC and the **direction of rotation M4** is output in conjunction with the **spindle command.**
The spindle accelerates to the preselected spindle speed because the **spindle inhibit** has already been reset.

If the **spindle inhibit** signal is still set when the **spindle command** is output, the **direction of rotation M4** will not be output until the **spindle inhibit** is reset.

As soon as the **spindle jog** signal is reset, the spindle decelerating cycle is initiated. When the spindle has come to a standstill, the **spindle command** signal is reset.

CAUTION
In normal operation, the spindle will generally be traversed in controlled mode only and not in the position control circuit. This means that it is possible that the spindle command has been reset while the spindle is still decelerating. In order to ensure that the spindle is stationary, the spindle stopped signal must also be queried.

Spindle speed jog 2^0 NC-I 1.0
Spindle speed jog 2^1 NC-I 1.1
Spindle speed jog 2^2 NC-I 1.2

These signals are effective exclusively in conjunction with the **spindle jog M3** or **spindle jog M4** input signals.

Depending on the coding, the spindle speed defined in MACODA parameter 1040 00015 will be activated. If the preselected spindle speed is not in the range of the current gear range, it will be limited to the minimum or maximum spindle speed of the activated gear range.

MACODA parameter	1040 00015 [1]	1040 00015 [2]	1040 00015 [3]	1040 00015 [4]	1040 00015 [5]	1040 00015 [6]	1040 00015 [7]	1040 00015 [8]
Spindle speed jog 2^0 NC-I 1.0	0	1	0	1	0	1	0	1
Spindle speed jog 2^1 NC-I 1.1	0	0	1	1	0	0	1	1
Spindle speed jog 2^2 NC-I 1.2	0	0	0	0	1	1	1	1

C axis ON NC-I 1.6

Using the C axis ON signal, the PLC switches to C axis operation.

Application:
In special cases (spindle/tool revolver), it may be necessary to switch on the C axis operation right after control unit start-up.

Signal edge

0 → 1: The running spindle is decelerated and switched into C axis operation.
The sequence within the NC is influenced by the settings in MACODA parameter 1040 00060.

1 → 0: No function.

C axis OFF NC-I 1.7

Using the C axis OFF signal, the PLC switches back to spindle mode.

Signal edge

0 → 1: The drive is switched back to spindle mode.
The switching sequence within the NC is influenced by the settings in MACODA parameter 1040 00060.

1 → 0: No function.

Override 2^0 .. 2^4 NC-I 2.0 .. 2.4

The current spindle speed can be influenced by using the **override function**. The override function is effective both on the spindle speeds programmed in the part program and in manual mode.

The **override switch** can have a maximum of **32 positions**, which are queried by the NC in **binary-coded format**.

A value representing the actual speed input as a percentage of the programmed speed is assigned to each binary code via MACODA parameter 1040 00041.

Spindle orientation manual NC-I 3.2

This signal or the NC part program (M19) calls up the ”spindle orientation” function. For this purpose, the control loop has to be closed and the drive under control.
The output signals **spindle in position**, **spindle orientation active** and **spindle is oriented** are significant in conjunction with spindle orientation manual.

☞ **This function is not available for analog spindles (MACODA parameter 1040 00001).**

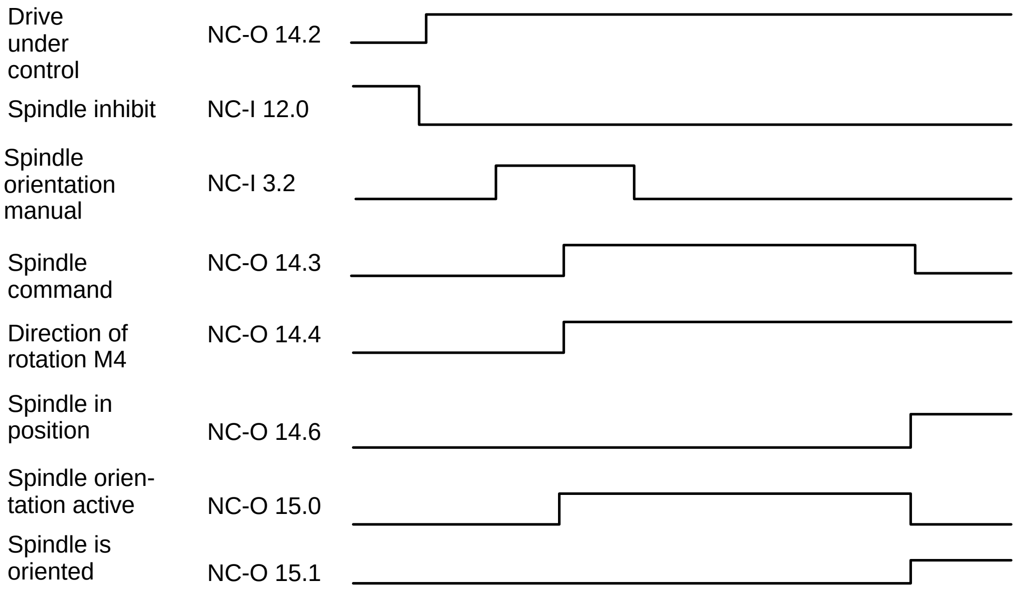
Signal edge

0 → 1: Spindle orientation is initiated. Spindle orientation is always performed on the zero pulse of the measuring system. The direction of rotation depends on SERCOS parameter S-0-0154.

1 → 0: No effect

Example:

The spindle is to be oriented manually, the control loop is closed.



The control loop is closed (**drive under control** is set) and **spindle inhibit** is reset.

When the input signal **spindle orientation manual** is set, the spindle receives the command to rotate, **spindle command** is set.
Depending on the direction of rotation, **direction of rotation M4** is set in this case where the spindle is rotating counter-clockwise (M4).

Upon the output of the **spindle command** signal, the input signal **spindle orientation manual** can be reset.
As soon as the spindle is rotating, **spindle orientation active** is set.

Once the spindle has been oriented in line with the marker position, the signals **M19 spindle is oriented**, and **spindle in position** are set and the **spindle command** signal is reset.

Spindle override 100% NC-I 3.3

The override function may be cancelled with this signal.

Signal edge

0 → 1: If the spindle override function is cancelled, the spindle traverses at the programmed speed.

1 → 0: The override function is active.

Reset (spindle) NC-I 3.4

Using the **reset (spindle)** signal, the current movement input or an active command (e.g. spindle orientation) can be canceled for the respective spindle. The spindle acknowledges the receipt of the reset instruction by setting the **control reset acknowledge** output signal (NC-O 19.4).

Gear range acknowledge 1 ... 4 NC-I 5.0 .. 5.3
Gear idle motion acknowledge NC-I 5.7

The “gear range acknowledge” signals are used to report the actual gear range. In the case of a gear range inconsistency, “idling” mode will be activated when the next spindle command is output.
At control startup, the gear range acknowledged is adopted in the NC.

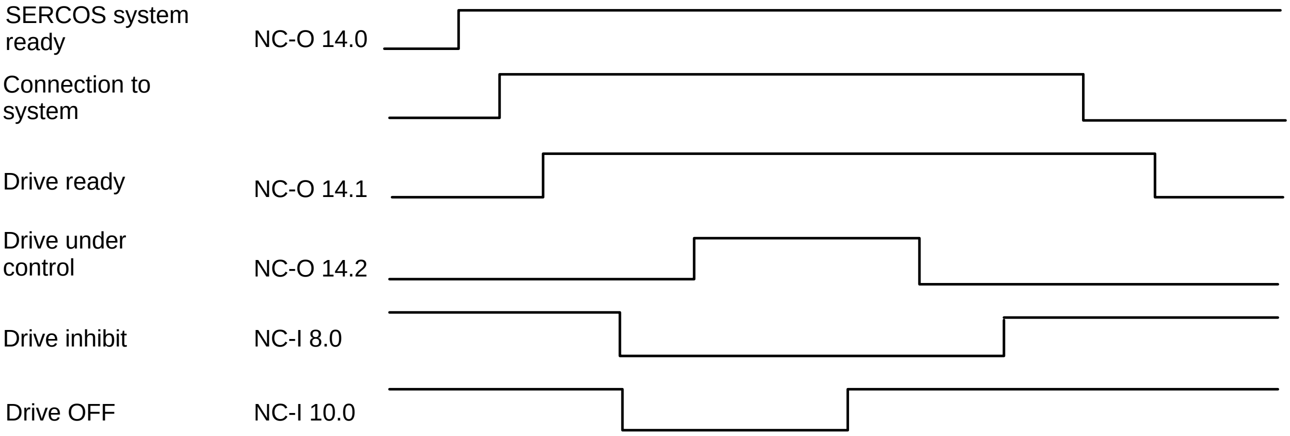
No analog spindle (1040 00001) gear changes are supported. In the case of analog spindles, the first gear range must be acknowledged.

Drive inhibit	**NC-I 8.0...9.7**
Drive OFF	**NC-I 10.0 ... 11.7**
(SERCOS system ready	**NC-O 14.0**
Drive ready	**NC-O 14.1**
Drive under control	**NC-O 14.2)**

These interface signals affect each other mutually and are of particular significance in the starting/shutdown cycles and the opening/closing of the position control loop.

☞ **The whole sequence does not apply to analog spindles.**

Example: Functioning of the starting and shutdown cycles



When the machine is started or voltage is applied, the NC unit and the servo and spindle drives go through an initialization phase. While the initialization phase of the individual components is running, also the digital interface for communication between the NC and the drives is established. (This digital interface corresponds to the settings of the SERCOS interface.)

Initialization of the SERCOS interface is in-system, automatic and consists of 4 phases. Once the interface is initialized and cyclic operation is active, the **SERCOS system ready** signal is output. The SERCOS system ready signal enables the respective spindle drive to be connected to the system.

When the spindle drive is connected, the **drive ready** signal is output. The position control loop is closed when the signals **drive inhibit** and **drive OFF** are reset by the PLC sequential program.
The output of the **drive under control** signal indicates that the position control loop is closed and torque is applied on the drive.

☞ **The ”drive OFF” and ”drive inhibit” signals are available 16 times each.**
In the iPCL and ICL 700, these individual signals are merged in the APS by an OR element to form a single signal.

The shutdown cycle or, resp., disconnection from the system, is initiated by the PLC sequential program setting the **drive OFF** signal. If the spindle is still in motion at this time, it is brought to a halt at the preset deceleration. When the spindle has come to a standstill, the control loop is opened and the **drive under control** signal is reset. No torque is applied on the drive any more.

In the next step, the **drive inhibit** signal can be set and the load can be disconnected.
The **SERCOS system ready** signal remains set for as long as the machine components (NC and drives) stay initialized. For bringing the spindle drive into circuit again, the procedure described above applies.

Spindle inhibit NC-I 12.0 ... 13.7

Spindle inhibit is effective when the control loop is closed and torque is applied on the spindle. This signal takes effect in all modes.
This spindle inhibit signal is available 16 times in parallel. The effects of the individual signals are identical. In the iPCL and ICL 700, these individual signals are merged in the APS by an OR element to form a single signal.

The spindle inhibit prevents the execution of a programmed or manually entered spindle movement. If spindle inhibit is set, the idling speed is not carried out.
The signals spindle command, direction of rotation M4 or spindle in position are not affected.

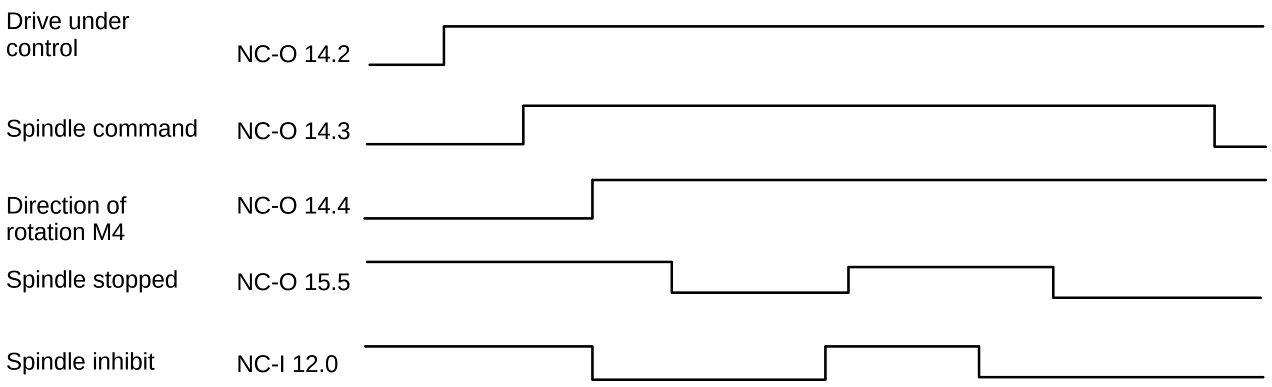
Signal level

1: The spindle cannot be rotated when spindle inhibit is set. If a rotating movement is active already, the spindle will be stopped at the parametrized deceleration.

0: Spindle rotation is not inhibited. If a spindle command is initiated (via the NC part program or manually) and the appropriate gear range is active, the spindle can start rotating right away. If a rotating spindle was brought to a standstill with **spindle inhibit**, this spindle will now start accelerating up to the initial speed.

Example:

The machine is switched on and the control loop has been closed. A rotary motion, e.g. M3 S1000, has been programmed via the NC part program.



The **drive under control** signal indicates to the PLC sequential program that the control loop is closed. As soon as the NC block in which the rotary movement was programmed is executed, the **spindle command** is output.

As long as **spindle inhibit** is still set, the rotary movement will not be carried out and the signals **direction of rotation M4** and **spindle stopped** will not be set. The direction of rotation can be queried only if a spindle command has been set.

When **spindle inhibit** is reset, the programmed rotary movement is executed, the **direction of rotation M4** is set according to the movement and **spindle stopped** is reset.

If **spindle inhibit** is set again during the rotary movement, the rotary movement is stopped and **spindle stopped** is set when the spindle has come to a standstill.

When **spindle inhibit** is reset, the spindle accelerates up to the previously applied speed, **spindle stopped** is reset and the rotary motion is continued.

Spindle command is reset when the spindle is stopped with the **M5** spindle stop function (NC part program or interface signal).

7.2.2 Output signals NC-O

SERCOS system ready NC-O 14.0

This signal is closely related to the signals **drive inhibit, drive OFF, drive ready** and **drive under control.**

The functional relationship is set out in detail in the signal diagram on page 7–12.

Signal level

1: SERCOS system ready is output when the initialization phases of the SERCOS interface, the spindle drive and the NC unit have been concluded without any errors occurring. This signal is prerequisite to bringing the drive into circuit (connecting the drive to the system).

0: The drive cannot be brought into circuit until the SERCOS system ready signal is set.

☞ **This signal is not generated for analog spindles (1040 00001).**

Drive ready NC-O 14.1

This signal is closely related to the signals **drive inhibit, drive OFF, SERCOS system ready** and **drive under control.**

The functional relationship is set out in detail in the signal diagram on page 7–12.

Signal level

1: The drive ready signal is output when the respective spindle drive has been brought into circuit. The signal remains set until the drive is cut out.

0: The spindle drive is not brought into circuit.

☞ **This signal is not generated for analog spindles (1040 00001).**

Drive under control NC-O 14.2

This signal is closely related to the signals **drive inhibit, drive OFF, SERCOS system ready** and **drive ready.**
The functional relationship is set out in detail in the signal diagram on page 7–12.

☞ **This signal is not generated for analog spindles (1040 00001).**

Signal level

1: When the drive under control signal is output, the control loop is closed and torque is applied on the drive. The spindle can be rotated unless it is locked by any other function, e.g. spindle inhibit.

0: The control loop is open, no torque is applied on the drive.

☞ **When changing from logic 1 to logic 0, spindle stop (M5) is triggered within the NC.**

Spindle command NC-O 14.3

The **spindle command, direction of rotation M4, spindle stopped** and **spindle in position** signals affect each other mutually.

Examples of their relationships are shown in the signal diagrams and explanations on the spindle M3 manual, spindle M4 manual, spindle M5 manual and spindle inhibit signals.

The **direction of rotation M4** signal is output in conjunction with the **spindle command.**

Signal level

1: The spindle command is set whenever the spindle is to carry out a rotary motion, which is input either **manually** (e.g. spindle M3 manual) or via the **NC part program** (e.g. M3 S1500). Prerequisite is that the control loop is closed, i.e. the **drive under control** signal must have been set.

0: Spindle rotation was stopped by a programmed M5 spindle stop function or by a spindle M5 manual input signal (NC-I 0.2). The spindle command signal is not set if **spindle orientation** or **positioning** or **gear range switching** is active.

Direction of rotation M4 NC-O 14.4

This signal is effective only in conjunction with the **spindle command** signal, i.e. the **direction of rotation M4** and **spindle command** signals must be linked by an AND element in the PLC sequential program.

In the absence of a spindle command signal, the direction of rotation M4 signal is irrelevant.

No direction of rotation M4 signal is output or, resp., its status remains unchanged with the functions gear range switching, idling speed and spindle orientation.

Signal level

1: Direction of rotation M4 set and spindle command set means that the spindle should carry out a rotary movement in a **counter-clockwise direction (M4)**.

0: Direction of rotation M4 not set and spindle command set means that the spindle should carry out a rotary movement in a **clockwise direction (M3)**.

Gear range switching active NC-O 14.5

Effective only in conjunction with the gear switchover function, triggered by the **M functions M40, 41, 42**.

Signal level

1: A change of gear range is executed.

0: The change of gear range has been executed.

Spindle in position NC-O 14.6

This signal is active when the control loop is closed (drive under control set) and torque is applied on the spindle.

The spindle in position signal is relevant only in conjunction with the **spindle orientation** or **spindle positioning** functions.

Signal level

1: The **spindle in position** signal is set when the spindle has been oriented or positioned and is located in the in-position window defined in the drive.

0: The spindle has not been oriented and/or is not in the parameterized in-position window.

Spindle orientation active NC-O 15.0

Effective only in conjunction with the **spindle orientation** function. The functional relationship is set out in detail in the signal diagram on page 7–10.

Signal level

1: Spindle orientation active is set if the **spindle orientation** function has been activated by the NC part program (e.g. programming M19) or the NC input signal **spindle orientation manual**.

0: Spindle orientation is not activated.

Spindle is oriented NC-O 15.1

The output signals **spindle orientation active** and **spindle in position** are of importance in conjunction with the spindle is oriented signal. The functional relationship is set out in detail in the signal diagram on page 7–10.

Signal level

1: The spindle orientation function has been executed and the spindle is in the oriented stop position.

0: The spindle orientation function has been executed but the spindle is no longer in its previous ”oriented stop” position.

Progr. spindle speed reached NC-O 15.2

Signal level

1: The actual spindle speed is identical with the spindle speed programmed and/or is within the speed tolerance band as defined for every spindle by the superposition of MACODA parameters 1040 00020 (spindle speed window in rpm) and 1040 00021 (spindle speed window in %).
If the speed is limited due to adjustments made (speed limits or gear range speed limits that have been programmed), this signal is output when the set limit is reached. In addition, the position of the override switch is taken into account.

0: The actual spindle speed is not within the speed tolerance band.

Spindle speed 100% NC-O 15.3

Effective in conjunction with spindle **override 2^0...override 2^4**.

Signal level

1: If the bit combination for **override = 100%** or the input signal spindle override 100% is present at the binary-coded input signals **override 2^0...override 2^4**, the spindle speed 100% output signal is set.

0: The bit combination of input signals **override 2^0...override 2^4** does not match the 100% coding.

Idling speed reached NC-O 15.4

This signal is effective in conjunction with the gear switchover and the checkback signal of the actual gear range.
The spindle override setting is not taken into account when the idling speed is output.

Signal level

1: If the actual spindle speed matches the idling speed, **idling speed reached** is set. This signal is normally used to switch gear ranges.

0: The idling speed has not yet been reached.

Spindle stopped NC-O 15.5

The **spindle stopped** signal refers to the spindle speed.

Signal level

1: Spindle stopped is set if the function M5 has been executed (NC part program or interface), speed 0 (S0) has been set via the NC part program, or the **spindle inhibit** or **override 0** signals have been set.

0: The spindle is turning at a speed that is above the defined tolerance band.

☞ **This signal is not generated for analog spindles!**

Output gear range 1 ... 4 NC-O 16.0 ... 16.3

Depending on the gear range selection (manual or via the NC part program), the preselected gear range is output.

The gear range can be selected directly by programming **M41...M44** (gear range 1...gear range 4) via the NC part program. The gear range is derived from the programmed **speed S ....** in conjunction with **automatic gear range recognition M40.**
If the speed ranges in the individual gear ranges overlap, the gear range with the higher spindle drive speed is output.

In the context of gear range switching, the **NC-O 15.4 idling speed reached** output signal is of importance.

Behavior after power-up:

After power-up or start-up of the NC, the gear range which is reported as an **acknowledgement** by the PLC program via the relevant inputs is output. If no gear range is acknowledged, no gear range will be output either. In this case, no gear range is output until a gear range is selected manually or via the NC part program.
If **automatic gear range recognition M40** is active upon power-up, the gear range is only output after a **speed S...** has been programmed in conjunction with a **direction of rotation M3** or **M4.**

Signal edge

0 → 1: Change of gear is to be performed with this edge. In order to minimize the mechanical impact of the gear change, the **spindle idling** function (if available in the spindle drive) is activated in the spindle drive via the SERCOS interface. As soon as the idling speed (possibly accompanied by spindle-gear meshing) is reached, the **NC-O 15.4, idling speed reached** output signal is set and the mechanical switching operation can be carried out. When the gear-switching operation has been executed, the actual gear range must be signaled back to the relevant input **NC-I gear range 1 acknowledge ... gear range 4 acknowledge**. Activation of the respective parameter block via the SERCOS interface concludes the gear-switching operation.

1 → 0: The gear range selected previously is now deselected.

Output gear in idle motion NC-O 16.7

Depending on the gear range selection (manual or via the NC part program), the preselected gear range or idling is output.

Idling can be selected directly by programming the **M48** function in the NC part program. Neither a direction of rotation, **M3** or **M4**, nor **spindle orientation, M19**, nor spindle speed **S...** may be programmed in conjunction with **M48**.

In the context of gear range switching, the **NC-O 15.4, idling speed reached** output signal is of importance.

Signal edge

0 → 1: This signal edge is used to switch the gear in idle motion. In order to minimize the mechanical impact of the gear change, the **spindle idling** function (if available in the spindle drive) is activated in the spindle drive via the SERCOS interface.
As soon as the idling speed (possibly accompanied by spindle-gear meshing) is reached, the **NC-O 15.4, idling speed reached** output signal is set and the mechanical switching operation can be carried out.
When the gear-switching operation has been executed, the fact that the gear is in neutral must be signaled back to the respective input, **NC-I 4.7, gear idle motion acknowledge**. This concludes the gear switching operation.

1 → 0: The gear in idle motion function is deselected.

C axis is active NC-O 17.0

The C axis is active signal indicates whether the respective spindle is in axis mode (position interface) or in spindle mode (speed interface). Spindles which can be switched over to C-axis operation have **a normal axis interface in addition** to the spindle interface. The C axis is active signal indicates the relevant interface.
An example of a signal sequence is shown on page 5–25 (G32 active).

Signal level

1: The spindle is in axis mode. The axis interface is valid, the spindle interface is not served by the NC.

0: The spindle is in spindle operation: The spindle interface is served by the NC, the axis interface is not valid.

Spindle speed limited NC-O 17.1

This signal indicates that the programmed spindle speed is limited either by the active gear range or by the programmed spindle speed limit (G192, G292).

Signal level

1: The programmed spindle speed is limited.

0: The programmed spindle speed is not limited.

Spindle override 0% NC-A 17.2

The **spindle override 0%** signal indicates that the override of the spindle is set to 0%.

Signal level

1: 0% act on the spindle.

0: A value not equal to 0% acts on the spindle.

Error diagn. class 1 NC-O 17.3

The error diagnostics class 1 signal (drive cutoff with interlock) is a direct map of the corresponding signal in the SERCOS drive status word of the spindle in question. The interlock can only be canceled by the command ”reset diagnostics class 1”, which is set automatically by the NC upon a control reset.

The pending error can be evaluated manually using the SERCOS monitor, or automatically by having the PLC directly read S-0-0011.

The parameter S-0-0011 comprises 2 bytes and has the following structure:

Bit 0:	–	
Bit 1:	Amplifier shut-off temperature	S-0-0203
Bit 2:	Motor shut-off temperature	S-0-0204
Bit 3:	–	
Bit 4:	Control voltage fault	
Bit 5:	Feedback error (encoder, measuring system)	
Bit 6:	Error in electronic commutation system	
Bit 7:	Overcurrent	
Bit 8:	Overvoltage	
Bit 9:	Undervoltage fault	
Bit 10:	Phase fault in power supply	
Bit 11:	Excessive controller deviation	S-0-0159
Bit 12:	Communication error	
Bit 13:	Position limit value exceeded	S-0-0049
Bit 14:	–	
Bit 15:	Manufacturer-specific error	S-0-0129

☞ **This signal is not generated for analog spindles (1040 00001).**

Signal level

1: A diagnostics class 1 error has occurred.

0: There is no diagnostics class 1 error.

☞ Refer to the **SERCOS Interface Specifications** for more information on diagnostics class 1 errors.

Class 2 diagnostics changed NC-O 17.4

The **class 2 diagnostics changed** signal (switch-off prewarning) is a direct map of the corresponding signal in the SERCOS drive status word of the spindle in question. A change in diagnostics class 2 sets the bit, which is reset again when the status has been read using parameter S-0-0012. The status can be read manually using the SERCOS monitor or automatically by having the PLC directly read S-0-0012.

The parameter S-0-0012 comprises 2 bytes and contains the following **warnings**:

Bit 0:	–	–
Bit 1:	Amplifier overtemperature warning	S-0-0311
Bit 2:	Motor overtemperature warning	S-0-0312
Bit 3 through 14:	–	
Bit 15:	Manufacturer-specific warning	S-0-0181

The effect of switch-off prewarnings concerning the alteration bit can be suppressed on the screen for **diagnostics class 2** (S-0-0097).

☞ **This signal is not generated for analog spindles (1040 00001).**

Class 3 diagnostics changed NC-O 17.5

The **class 3 diagnostics changed** signal (messages of operating statuses) is a direct map of the corresponding signal in the SERCOS drive status word of the spindle in question. A change in diagnostics class 3 sets the bit, which is reset again when the status has been read with using parameter S-0-0013. The status can be read manually using the SERCOS monitor or automatically by having the PLC directly read S-0-0013.

The parameter S-0-0013 comprises 2 bytes and contains the following **messages**:

Bit 0:	$n_{act} = n_{set}$	S-0-0330
Bit 1:	$n_{act} = 0$	S-0-0331
Bit 2:	$\|n_{act}\| < \|n_x\|$	S-0-0332
Bit 3:	$\|Md\| \geq \|Md_x\|$	S-0-0333
Bit 4:	$\|Md\| \geq \|Md_{limit}\|$	S-0-0334
Bit 5:	$\|n_{set}\| > \|n_{limit}\|$	S-0-0335
Bit 6:	In position	S-0-0336
Bit 7:	$\|P\| \geq \|P_x\|$	S-0-0337
Bit 8:	–	
Bit 9:	$\|n_{act}\| \geq$ min. spindle speed	S-0-0339
Bit 10:	$\|n_{act}\| \geq$ max. spindle speed	S-0-0340
Bit 11:	In position rough	S-0-0341
Bit 12:	Target position reached	S-0-0342
Bit 13:	Interpolator halted	S-0-0343
Bit 14:	–	
Bit 15:	Manufacturer-specific operating statuses	S-0-0182

The effect of operating statuses concerning the alteration bit can be suppressed on the screen for **diagnostics class 3** (S-0-0098).

☞ **This signal is not generated for analog spindles (1040 00001).**

Position control active NC-O 17.6

Signal level

1: The spindle is activated via the position interface.

0: The spindle is activated via the speed interface.

C axis switching active NC-O 17.7

The “C-axis switching active” signal indicates that the spindle is currently being switched from spindle mode to C-axis mode or vice versa.
The C axis switching signal is always accompanied by a simultaneous spindle command.

An example of a signal sequence is shown in the context of the **channel** output signal, **G32 active** (NC-O 20.2).

Signal level

1: Spindle is being switched over to/from C-axis mode.

0: Spindle is not being switched over to/from C-axis mode.

Index of coupling bit 0 .. 2 NC-O 18.0 .. 18.2

When spindle coupling is active, the coupling index of every member of a group of coupled spindles is output with the **index of coupling bit 0 .. 2** signal.

Spindle is master NC-O 18.3

This signal identifies the master of a group of coupled spindles (see also signal NC-O18.0 .. 18.2).

Signal level

1: The spindle is the master spindle.

0: The spindle is a slave spindle.

Synchronous 1 NC-O 18.4

With spindle coupling active (NC-O 18.0 .. 18.2 signal is set) for slave spindles (NC-O 18.3 is not set), the **synchronous 1** signal provides the following information on the angular offset:

Signal level

1: The synchronous mode window is being complied with and a programmed angular offset has been executed.

0: The synchronous mode window is not being complied with or a programmed angular offset is just being executed.

Synchronous 2 NC-O 18.5

With spindle coupling active (NC-O 18.0 .. 18.2 signal is set) for slave spindles (NC-O 18.3 is not set), the **synchronous 2** signal provides the following information on the fault window:

Signal level

1: The synchronous mode error window is being complied with.

0: The synchronous mode error window is not being complied with.

Coupling error NC-O 18.6

With spindle coupling active (NC-O 18.0 .. 18.2 signal is set) for master spindles (NC-O 18.3 is set), the **coupling error** signal provides information on the current status of the group of coupled spindles.

Signal level

1: A coupling error has occurred in the group of coupled spindles (NC-O 18.0 .. 18.2 signal is set). The error can be cleared by a ”master spindle control reset” or by an ”overall control reset”.

0: The status of the group of coupled spindles is normal.

Control reset acknowledge NC-O 19.4

The **control reset acknowledge** signal indicates that the spindle accepted the control reset task and that the spindle is ready to accept new inputs for motions.

SERCOS drive status bit 0 ... Bit 7 NC-O 21.0......21.7

The bit signals as configured in the cyclic telegram from the spindle drive are transferred to the NC in the signal status word (parameter S-0-0144).

The status signals can be configured in parameter S-0-0026. Parameter S-0-0144 must be included in the configuration list of the drive telegram (S-0-0016) to enable transmission of the signals in the cyclic telegram.

Configuration example:

- S-0-0026 = (330, 331,0, 333, 0, 335, 336, 337):
  Configuration of the signal status word
- S-0-0016 = (40, 144):
  Configuration of the drive telegram

The following status messages are allocated to the individual spindle output signals:

NC-O 21.0	: S-0-0330	Message $n_{act} = n_{set}$
NC-O 21.1	: S-0-0331	Message $n_{act} = 0$
NC-O 21.2	: -	
NC-O 21.3	: S-0-0333	Message $M_d \geqq M_{dx}$
NC-O 21.4	: -	
NC-O 21.5	: S-0-0335	Message $n_{set} \geqq n_{limit}$
NC-O 21.6	: S-0-0336	”In position” message
NC-O 21.7	: S-0-0337	Message $P \geqq P_x$

☞ **For a detailed description of all status messages available, please see the SERCOS Interface Specifications.**

Notes:

8 External spindle speed

If an external spindle is defined within the system, the actual spindle speed is transferred for display on the control panel by the PLC program. The PLC program writes the spindle speed value to an area within the PLC operands. This area is defined by MACODA parameters 2060 00002 [8] (operand type) and 2060 00003 [8] (operand address). Every spindle speed value uses 4 bytes.

If several external spindles are defined, their speeds are stored consecutively under this address.

The NC reads the speed to be displayed on the operator panel from this area.

Notes:

9 Auxiliary functions

Auxiliary functions are programmed in the NC part program.

All **bit-coded auxiliary functions** programmed in **one** NC block are mapped in parallel via the APS to markers.

Parallel data transfer helps avoid machine down-time.

Auxiliary function output is cyclic with each PLC cycle provided an auxiliary function has been programmed in the NC part program.

The following conventions apply to auxiliary functions:

- On every channel, there are several auxiliary functions available. The designations (types of auxiliary functions) of these auxiliary functions (M, S, T, ...) are defined in MACODA parameter 3010 00030.
- An auxiliary function can be mapped by the APS onto an operand area (MACODA parameter 3010 00040) as a
  - **bit-coded auxiliary function**
  - **32-bit BCD-coded auxiliary function**
  - **64-bit BCD-coded auxiliary function**
  - **32-bit BCD-coded channel-independent auxiliary function**
  - **64-bit BCD-coded channel-independent auxiliary function**
- Several function types can be declared as bit-coded auxiliary functions for each channel (MACODA parameter 3010 00030), e.g. the auxiliary function with the designation **M** and the auxiliary function with the designation **H.**
- The start addresses of bit-coded auxiliary functions are set individually for each channel in MACODA parameter 2060 00009.

9.1 Bit-coded auxiliary functions

Auxiliary functions are required in the PLC sequential program most often in the form of individual signals, e.g. to switch the coolant on or off, to open or close a clamping mechanism, or to extend or retract a cylinder.

Auxiliary functions are programmed in the NC part program in order to trigger these functions. The decoded auxiliary functions are mapped on an operand range (e.g. markers). The absolute size of this range is dependent on the number of auxiliary functions which are to be decoded as well as on the number of channels (MACODA parameter 9040 00001).

The operand type (e.g. marker) is defined in MACODA parameter 2060 00002[5].

The start address of the operand area is defined individually for each channel in MACODA parameter 2060 00009.

Example:

There are 3 channels defined in the NC; for each channel, there are 128 (MACODA parameter 3010 00010 [2]) decoded type **M** auxiliary functions declared (MACODA parameter 3010 00030) and the operand area is to be at marker M200.0 for channel 0 (MACODA parameter 2060 00009[1]). The auxiliary functions of the other channels follow one after another (MACODA parameter 2060 00009[2] –> M240, MACODA parameter 2060 00009[3] –> M280).

Owing to these parameters, the marker field has the following structure:

Channel	M function	Marker field
0	M31... M0	M203.7 ... M200.0
	M128... M96	M215.7 ... M212.0
1	M31... M0	M243.7 ... M240.0
	M128... M96	M255.7 ... M252.0
2	M31... M0	M283.7 ... M280.0
	M128... M96	M295.7 ... M292.0

In this example, a 16-byte marker field is defined for every channel. In these marker fields, **every bit-coded auxiliary function occupies one marker**.

Bit-coded auxiliary functions **may or may not require acknowledgement.**

MACODA parameter 3010 00020 defines which auxiliary function requires acknowledgement.

Acknowledgement is made by the PLC user program resetting the respective bit-coded auxiliary function.

As long as an auxiliary function which requires acknowledgement is not reset by the PLC user program, the following NC block will not be executed.

Auxiliary functions which do not require acknowledgement must likewise be reset by the PLC user program in order to recognize the changed signal level from logic 0 to logic 1 when the auxiliary function is output again.

Bit-coded auxiliary functions are first and foremost channel-related, i.e. the assigned bit on the cannel concerned is set. With M10, for example, the 11th bit of the channel on which the auxiliary function was output is set.

This way, even large auxiliary function indexes can be assigned to unused auxiliary functions in the lower area. Thus, M999 can address bit 109, e.g.

Depending on the application, operation of the PLC user program can be made easier by assigning a **global effect** to some bit-coded auxiliary functions.
For this purpose, a specific auxiliary function of a random channel is assigned to the global auxiliary function in the marker field.

9.2 BCD-coded channel-independent auxiliary functions

Apart from the bit-coded format (type M, e.g.), auxiliary functions can be output also in BCD code with 32 or 64 bits (type S, e.g.).

Channel-independent BCD-coded auxiliary functions are not assigned to any specific channel and therefore have the same effect on every channel.

For every BCD-coded type of auxiliary function, an operand area is defined. Every operand area has either 8 bytes (32-bit BCD value) or 12 bytes (64-bit BCD value).

The address locations of these operand areas are defined in MACODA parameters 2060 00002 (operand type) and 2060 00003 (operand address).

Example:

The ”S” auxiliary function has been declared to be a 32-bit-coded auxiliary function.

The start address of the marker field of the ”S” auxiliary function is to begin at the M20 marker byte.

Marker M23 reserved	Marker M22 reserved	Marker M21 reserved	Marker M20 Control code
Marker M27 MSB	32 bit BCD value		Marker M24 LSB

The control code in marker byte M20 has the following meaning:

Marker M20
Control code

Bit 0	Change signal
Bit 1	Sign
Bit 2	reserved
Bit 3	reserved
Bit 4	reserved
Bit 5	reserved
Bit 6	reserved
Bit 7	reserved

The example above shows that when a BCD-coded auxiliary function is output to the byte M24 through M27 marker field, also the M20.0 change signal is set. The PLC user program must acknowledge the output of the auxiliary function by resetting the change signal on marker M20.0.

If the auxiliary function has been programmed with a negative sign in the part program, the M20.1 marker will be set, too.

All BCD-coded auxiliary functions have the same structure as shown in the example above. For 64-bit auxiliary functions, the total length of the corresponding marker fields is 12 bytes instead of 8 bytes.

Notes:

10 PROFIBUS-DP (DP master) interface

The PNC-P and the osa dc interface module of the PNC-R have a PROFIBUS-DP interface, which may be used as the **DP master interface** in combination with the iPCL/ICL 700.
This interface is designed for decentralized connection of peripheral PROFIBUS-DP modules.

10.1 Commissioning of PROFIBUS-DP interface

Installation steps for the **PNC-R** (with ICL and KNS) or **PNC-P** (with KNS):

1. Create a **master parameter set** (see WinDP documentation) using the **WinDP** configuration tool by Rexroth (runs under Window 95/NT).
2. Use the ”load” function of the WinDP tool to store a **master parameter file** (e.g. bmlist.mxa) on the PC where you are running WinDP.
   Subsequently, you must transfer this file to the PNC-R file system because the osa dc interface module does not support remote configuration.

☞ **Save the name of this master parameter file in format 8.3 (evaluation by the NC is case-sensitive).**

3. Next, you must adapt the group 4080 MACODA parameters, where, among other entries, the file name of the master parameter set must be entered. For a description of these parameters, see the ”Configuration Parameters and MACODA Parameter Description” manual.
4. A PNC-R start-up has to be initiated to check whether the installation was successfully concluded.
   When the start-up has been concluded, the green LED P1 on the front panel of the osa dc interface module lights up. If this is not the case, error messages appearing on the PNC-R operator panel will indicate possible sources of error.
   If there is a data exchange failure with connected DP slaves, an error message or a warning will be displayed on the PNC-R operator panel. The decentrally connected peripheral modules can only be **accessed properly** if no error message or warning indicating a data exchange failure is displayed any more on the PNC-R operator panel.
5. In addition, you must adapt the PLC user program of the ICL 700 before you can commission the DP master. This adaptation is described in Section 10.2 below.

Installation steps for the **PNC-R** (with iPCL) or **PNC-P** (with iPCL):

1. Create a **master parameter set** (see WinDP documentation) using the **WinDP** configuration tool by Rexroth (runs under Window 95/NT).
2. Use the ”load” function of the WinDP tool to automatically store the master parameter set as a file in the PNC-P under the file name entered in MACODA (group 4080).

☞ **For a description of these parameters, see the ”Configuration Parameters and MACODA Parameter Description“ manual.**

3. Subsequently, the PNC-P has to be started again using RUN.
   Possible error messages and information on possible error sources are displayed on the operating panel of the PNC-P.
   If there is a data exchange failure with connected DP slaves, an error message or a warning will be displayed on the operating panel.
   The diagnostics tools of WinDP (see ”WinDP” documentation) can be used for analysis of the error cause.
   The decentrally connected peripheral modules can only be accessed properly if no error message or warning indicating a data exchange failure is displayed any more on the operating panel.
4. You must adapt the PLC user program of the iPCL before you can commission the DP master. This adaptation is described in Section 10.2.

10.2 Determining the DP master status

Behavior of the ICL:

The status of the DP master is determined and certain measures in the PLC user program are initiated with the S34 and S36 data words:

- Data word S34: Entry of errors reported by the DP master after initialization or during operation.
- Data word S36: Control of the DP master status from the PLC user program.

The DP master status is stored in the system data area of the ICL 700.

Example: In the simplest case, all you have to do is insert the following two lines in your cyclic PLC user program:

```
L    W    S34,A        ; load DP master status (S34) in register A
T    W    S36          ; and transfer it to data word S36
```

Behavior of the iPCL:

In case of the iPCL, the DP master status is stored in the S 114 data word. The slave diagnostics is returned in the system area S 240 – S 255. It is used to determine all faulty slaves (see ”iPCL System Description and Programming Manual”).
These data are used to determine the state of the bus and allow for the PLC program to respond to possible errors.

For the evaluation in the PLC user program, see the following table, ”DP Master Status, Bits and Errors”:

Bit	Meaning
0	Bus master error
1	Classified slave diagnostics
2	System diagnostics
5	DP bus set to CLEAR
6	PG retains DP bus in STOP status (cannot happen)
7	DP bus master is active
8	Slave(s) not available
9	Slave(s) signal(s) configuration error
10	Slave(s) signal(s) static diagnostics
11	Slave(s) signal(s) extended diagnostics
12	Slave(s) not ready
13	Slave(s) invalid

Errors that are signaled by the DP master in the DP master status do not necessarily lead to an iPCL/ICL 700 STOP. Any error handling, if required, must be done in the PLC user program.

10.3 Digital and analog osa dc interface I/Os (PNC-R with SNCI4)

Access routines to digital or analog I/Os from the PLC user program are adapted in MACODA parameter group 4075 (see the ”Configuration Parameters and MACODA Parameter Description” manual). This adaptation involves setting the I/O signal assignments in the PLC and the application of PLC addresses.

Digital and analog I/O signals are displayed on the PLC like decentralized peripherals.

The digital and analog I/Os are connected to ports X12 and X11 on the osa dc interface module (DCIO).

An extension card with digital I/Os is available for the PNC-P.

☞ **In case of read access to the digital inputs of the DCIO, please note that the signals appear inverted on the iPCL!**

11 Machine error and status display (MSD)

This function facilitates the display of a total of 2048 machine errors, warnings, messages or machine statuses on the NC control panel, thereby supporting rapid error search when malfunctions occur in the machine operation.

The respective messages are defined in plain text in one file (MSD file) and activated when the respective marker is set by the PLC sequential program.

11.1 MSD files and displays

Function and structure of the MSD file

The same overall conditions apply to creating and handling the MSD file as to creating an NC part program.

The MSD file receives the name “**mzatexte.???**”, with “???” standing for the respective language to be used, e.g. 044 for English.

Every line in the MSD file must end with a ”**)**”.

A total of **2048 messages** can be defined. Every message is assigned a serial number (1...2048). There may be gaps in the list of serial numbers.

The various types of messages are distinguished as follows:

Machine errors	designation **E**
Machine warnings	designation **W**
Machine messages	designation **M**
Machine status	designation **Z**
Operating status	designation **B**

In addition to this classification by machine error, machine status and operating status, these messages can be classed by **groups** in the MSD file, to increase clarity in the display unit and facilitate error search when malfunctions occur. A group is defined by the **designation G**.

Designation I xxxx is used to designate an area of “**fast signals**”, with xxxx representing a serial number from 1 to a max. of 2017 (in steps of 32). A total of 32 messages can be defined as fast signals. Programming “**(I33)**” means that the fast signals are stored under serial numbers **33 .. 64.**
These fast signals are checked for changes by the APS after each PLC cycle. This ensures that short-time changes present for at least one PLC cycle are transferred to the host and, if necessary, to the display unit. Up to 48 messages with their group designations can be shown on each screen of the display.

Display:

Machine errors, warnings and messages are displayed in the INFO dialog and in the status line of the OI. The status line displays the highest priority message. In addition, machine errors are displayed in the DIAGNOSTICS operating area in the MSD dialog function.
Operating status messages differ from machine status messages insofar as they are not only displayed in the MSD dialog but are, in addition, transferred like machine errors and warnings via the communication interface to the host computer.

Formatting and length of MSD texts:

- The MSD dialog displays a maximum of 38 characters of an MSD text. The character sequence ”\n” in the text has the effect that the text following this character sequence is not displayed.
- A maximum of 2 lines with 18 characters each can be displayed in the NC status line. The text following ”\n” is ignored.
- A maximum of 41 characters are displayed in the info list of the INFO dialog. The text following ”\n” is ignored.
- In the field ”selected info” of the INFO dialog, the MSD text can be displayed in several lines. 34 characters per line are allowed. An ”\n” in the MSD text results in a line break.

  Tip:
  2-part structure of the MSD text (separated by ”\n”:
  1. The text of the machine error, warning or message.
  2. Additional information (e.g. elimination text)
     The additional information would be displayed in the INFO dialog when the corresponding MSD message occurs.

  Example:
  (E20, emergency stop actuated from the magazine table\nsee error description\ngroup1)

Example of an MSD file structure:

(G1, protective devices)
(E1, EMERGENCY stop actuated)
(E2, touch guards actuated)
(E3, light barrier actuated)
(E4, door opened)
(G2, lubrication)
(E20, no oil pressure)
(E21, oil level too low)
(E22, oil cooler inactive)
(W29, oil contaminated)
(M30, oil slightly contaminated)
(Z31, oil has reached operating temperature range)
(Z32, oil level sufficient)
(G3, collision monitor)
(I97)
(E100, part 1 and 2 temporarily collided)
(Z110, inject adhesive)

	Identifier	MSD dialog	Status line	INFO dialog	Host
Error	E	yes	yes	yes	yes
Warning	W	yes	no	yes	yes
Message	M	yes	no	yes	no
Status	Z	yes	no	no	no

Operating status	B	yes	no	no	yes
Group	G	yes	no	no	no

11.2 Activating messages

Machine errors, warnings, messages, statuses or operating statuses can be activated by setting defined bit signals or deleted by resetting the respective markers. The range of these bit signals is defined by MACODA parameters 2060 00003 [4] (operand type) and 2060 00002 [4] (operand index). **Marker area M4000.0 ... M4255.7** is currently reserved for a total of **2048** messages.

These markers are set or reset exclusively by the PLC sequential program.

Marker **M4000.0** is assigned to the message with serial number 1, marker **M4000.1** to the message with serial number 2.

Notes:

12 Program modules for iPCL and ICL700

12.1 General structure of program modules

The APS contains a number of program modules, which can be called up by the user in the PLC sequential program cyclically and with parameter settings.
Apart from setting the parameters required for a function (e.g. a program number for the external program selection), the user must also define the data area where the data is to be stored in each case.

Within this data area, there are a status word and an error code defined if an error occurs.

Thus, you must set two parameters for every program module call.
The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.
If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

☞ **The program examples shown in the context of APS program module descriptions show the PROFI software syntax. For PLC user programs that have been prepared with WinSPS, this syntax must be adapted accordingly.**

As a rule, these data areas, the addresses of which are defined in parameter P0, have the following structure:

MSB LSB

Bit 31 0

DW0	Error code (2 bytes)	Status (1 byte)	Job (1 byte)
DW1	Length of the total data area reserved (4 bytes)		
DW2	Channel number	Offset from the beginning of the data area to the 1st data feedback address	
DW3	Transfer parameter for the activated function, e.g. program number for ext. program selection. The length of this data area is dependent on the function concerned. The entry is completed with the NUL character if input is made in the form of a character string.		
DWx	Data feedback, e.g. actual axis values, CPL variables, etc.		

DW: Designation of the operand (input, output, marker, entry in data module) as a double word with a length of 32 bits.

Double words DW0...DW2 appear in every program module data area but they are defined only depending on the respective function. If, e.g., no channel number is required, constant K-1 must be entered instead.

The same applies to entering the offset if there is no data feedback involved in a function.

Double word DW0

Double word DW0 contains the error code, status and job.

The **job** has a length of 1 byte and may contain the following functions:

MSB LSB

Bit 31 . . . . 16 15 . . . . 8 7 0

Error code	Status	Job

Bit 0	Start function (=1)
Bit 1	to be defined depending on the respective function
Bit 2	”
Bit 3	”
Bit 4	”
Bit 5	”
Bit 6	”
Bit 7	”

The **status** has a length of 1 byte and may contain the following functions:

MSB LSB

Bit 31 16	15 8	7 0
Error code	Status	Job

Bit 0 to be defined depending on the respective function
Bit 1 ”
Bit 2 ”
Bit 3 ”
Bit 4 Function active (=1)
Bit 5 to be defined depending on the respective function
Bit 6 ”
Bit 7 Error (see also error code)

☞ **When the PLC user program is saved on a PCMCIA card or in the FEPROM on the SM, data areas (e.g. data modules) are saved, too, which may contain data areas for client module calls. If the function active bit is set when the PLC user program is being saved, the function active status is saved, too, with the effect that the respective function is not carried out when the user program is loaded the next time.**
The same problem arises with data areas located in a retentive marker area or a data module because, when the NC is restarted, these areas are not automatically deleted by the NC.
To avoid this, data areas used for communication should be preassigned valid values by the PLC in a start-up module (OM5 or OM7) or in the 1st PLC cycle.

The **error code** has a length of 2 bytes:

MSB LSB

Bit 31 16	15 8	7 0
Error code	Status	Job

Error code, see description of the function concerned

The type of error or status is represented in the error code in the form of a **binary-coded, signed number**.

There are two categories of errors and statuses:

1. Errors and statuses caused by incorrect parameter settings or an internal NC error.
   These messages are listed under **general errors and statuses** in the following.
2. Errors and statuses signaled as an acknowledgement upon the execution of a function.
   These messages are listed under **function-specific errors and statuses** in the following.

An error code is output whenever an error occurs in the execution of a function which prevents the proper execution of the respective function.

Errors may be caused not least by incorrect parameterization, e.g. an invalid channel number. Also, system errors may occur, which will also be displayed with the respective error code.

Because some of the error messages are function-specific, they are explained in the description of the respective function in this manual.

Error codes are defined by Rexroth. They are updated as required in the context of new releases.

Double word DW1

Double word DW1 is used to define the size of the data area required to transfer or receive the data of the respective function.

The length of the data area is stated in bytes to be entered as an integer.

Example:

The length of the data area is to be 40 bytes. In double word DW1, constant K 28_H must be entered.

Double word DW2

In double word DW2, an offset must be entered which defines the length from the beginning of the data area down to the address where the data feedback is to be stored.

In addition, a channel number must be defined here if the function to be executed requires a channel number.

Entries are made in binary code.

MSB LSB
Bit 31 . . . . 16 15 . . . . 8 7 0

Channel number	Offset from the beginning of the data area to the 1st data feedback address

The offset length is stated in bytes to be entered as an integer. If there is no data feedback involved in the respective function, constant K-1 must be entered in byte 0 and in byte 1 of double word DW2.

Example:
The data area begins with the start address at marker byte M200.0. The data feedback is to be stored from marker byte M232.0 on. Thus, the offset is 32 bytes.

Constant K 20_H must be entered in byte 0 and byte 1 of double word DW2.

The channel number must be entered as an integer. If no channel number is required for the function to be executed, constant K-1 must be entered in byte 2 and byte 3 of double word DW2.

Example:
The actual values of the axes on channel 1 are to be scanned.

Constant K B_H must be entered in byte 2 and byte 3 of double word DW2.

Double word DW3

From double word 3 on, the address area is used to define the parameters that must be transferred additionally for the respective function to be executed, e.g. for the **external program selection** module.

In the case of the external program selection module, the parameter required for the execution of this function is the part program number, and for CPL variable declarations, their permanent variable numbers are required, etc.

If no parameters need to be transferred for the execution of a function, no definition of double word DW3 is required. Instead, data feedback may be stored already from here on.

Double word DWx

This address is dynamic and is determined indirectly by the offset defined in double word DW2.

Data feedback, e.g. actual axis values or CPL variables, is stored in this area.

12.2 Read actual axis values (-B02AXPOS) program module

Function

The program module **read actual axis values (-B02AXPOS)** supplies the **actual axis values** of **all** axes as at the time of the request.

Accordingly, enough storage space must be reserved for this purpose in the data module or marker field.

The **actual axis value** feedback is scaled to 0.1µ and represented as an integer with 4 bytes.

As regards cyclic calls of this module, please note that the number of calls should be limited to what is necessary to reduce the strain on system capacity. This can be achieved, e.g., by timer programming, time-controlled modules or by calling the module only every nth cycle.

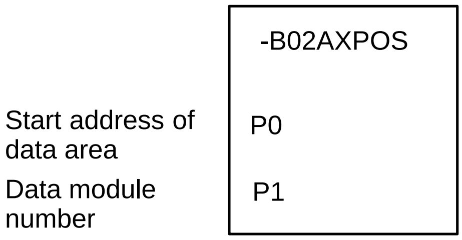
12.2.1 Module call

This module may be called once or repeatedly in a PLC cycle, and calls may be unconditional or conditional.

If the module is called again before the data from the previous call is received, please note that a different data area must be activated.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, parameter P1 contains the data module number.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B02AXPOS**, the data area, the address of which is defined in parameter P0, has the following structure:

	MSB		LSB
	Bit 31		0
DW0	Error code (2 bytes)	Status (1 byte)	Job (1 byte)
DW1	Length of the total data area reserved (4 bytes)		
DW2	Channel number	Offset from the beginning of the data area to the 1st data feedback address	
DW3	1st actual axis data feedback		
. . . .	:		
DW4	last actual axis data feedback		

This function is activated when bit 0, **start function**, is set in the job byte and remains set for at least one PLC cycle.

Start function may be reset by the PLC sequential program with the **function active** message.

If not reset by the PLC sequential program, **start function** is automatically reset when the function is finished.

Status bit 4, **function active**, is set. It remains set for one PLC cycle at least, or until the function has been executed.

Execution of this function is complete as soon as the data feedback is received. Then, status bit 4, **function active**, is reset.

This module provides actual data feedback of every axis, regardless of the channel number transferred.

To ensure compatibility also in the future, –1 should be entered for the channel number.

12.2.2 Configuration example

There are 3 axes defined on channel 1. The actual axis values are to be scanned with the positive edge of PLC input I0.2.

The data area is to begin at marker 200.

Given these requirements, a **-B02AXPOS** program module call may be designed as follows:

```
AN      B       I0.2
A       B       M302.0
R       B       M302.0              ; resetting the edge marker
A       B       M302.7              ; evaluate answer ?
JPC             -checkRsp
A       B       I0.2                ; scan input 0.2
AN      B       M302.0              ; formation of positive edge
JPI             -end
S       B       M302.0              ; setting the edge-trigger marker
S       B       M302.7              ; setting the evaluate answer marker
S       B       M200.0              ; setting start function
L       DW      K24,A               ; definition of parameters to be transferred
T       DW      A,M204              ; total length of data area = 24 bytes
L       W       K12,A               ; 12 bytes offset from beginning of data area
T       W       A,M208              ; actual axis value of 1st axis,
L       W       K1,A                ; request for
T       W       A,M210              ; actual axis value of channel 1.
CM              -B02AXPOS,2         ; call module -B02AXPOS
P0              M200                ; the data area begins at marker 200
P1              K-1                 ; not in a data module
JP              -end
                -checkRsp
L       DW      M200,A              ; job, status and error codes in reg. A
A       B       A.12                ; wait while function is still active
JPC             -end
A       B       M302.7
R       B       M302.7              ; resetting of evaluate answer
A       B       A.15                ; error flag set ?
JPI             -end
                                    ; error handling
EM                                  ; processing of received data
                -end
EM
```

The actual axis value of the

1st axis is stored on markers	212...215,
2nd axis is stored on markers	216...219,
3rd axis is stored on markers	220...223.

12.2.3 Status messages

When the **read actual axis values** function has been called and an error or a status occurs which prevents the proper execution of this function, **bit 7** is set in **status**.

In addition, the error code indicates the error type or, resp., the status in binary-coded format.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$: Parameter errors caused by incorrect parameter input in the data area.

FFFE$_H$: Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.

FFFD$_H$: Data overflow, the reserved data area is used to capacity, not enough space to store all the data.

FFFC$_H$: Module **B01APSMN** has not been called.

FFFB$_H$: Incorrect offset: The offset is too large (> reserved area) or too small (overwriting of job data occurs).

8000$_H$: NC-internal error, cannot be influenced by user.

7FFF$_H$: PxRos error, cannot be influenced by user.

7FFE$_H$: NCS error, cannot be influenced by user.

Function-specific errors or statuses occurring with the acknowledgement that the function has been executed:

1$_H$: Invalid channel number.

12.3 External program selection/deselection (-B03PANAB) program module

Function

The external program selection/deselection module facilitates the selection and deselection of an NC part program via the PLC sequential program.

The **program change without control reset** function allows selecting another program without a modal function reset. You can, e.g., select another program while the spindle is still running if you delete ”M5” in the init string. This applies on the condition that the program selected previously has been fully executed or has not been started yet.

☞ **The function ”program change without control reset” is applicable from NC software version V2.3.4 up.**

12.3.1 Module call

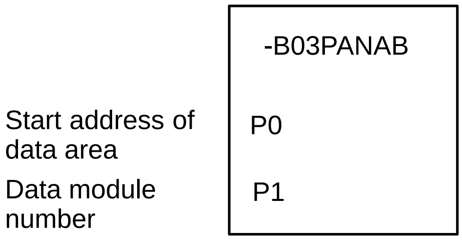
This module can be called unconditionally or conditionally.

If several channels are activated, the module can also be called several times per PLC cycle.

If the module is called repeatedly, please note that a different data area must be parameterized for each call.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B03PANAB**, the data area, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 0

DW0	Error code (2 bytes)	Status (1 byte)	Job (1 byte)
DW1	Length of the total data area reserved (4 bytes)		
DW2	Channel number	Offset from the beginning of the data area to the 1st data feedback address	
DW3	Program number in BCD or binary code		
DW4	Program name, max. 30 ASCII characters, with a trailing NUL (00_H) ASCII character for termination		
. . . .	Start block, max. 9 ASCII characters, with a trailing NUL (00_H) ASCII character for termination		
DWx	End block, max. 9 ASCII characters, with a trailing NUL (00_H) ASCII character for termination		

The job byte is structured as follows:

Bit	**Function**	
0	Start function	
1	reserved	
2	Program number is BCD-coded	
3	Program selection with start block	
4	Program selection with end block	
5-7	Function number:	
	0:	reserved
	1:	reserved
	2:	Program deselection
	3:	reserved
	4:	Program selection without automatic cancellation and modal function resetting
	5:	Program selection with automatic cancellation and modal function resetting
	6:	Program change without modal function resetting
	7:	reserved

An NC part program can be selected only if either **automatic or single block** operating mode is active.

The function is started when bit 0 is set to logic 1 in the job byte.

Bit 0 may be reset via the PLC sequential program with the **function active** message (bit 4 in status). Otherwise, the ”start function” signal is reset automatically when the function has been carried out.

Bit 4 in the status byte, **function active**, remains set until the function has been carried out, at least, however, until the subsequent call of the **-B01APSMN** program module.

As soon as a program has been selected/deselected, this function is completed and bit 4, **function active**, in the status byte is reset.

Programs can be selected by entering the program number in BCD or binary code or, optionally, by entering the program name with a max. of 30 ASCII characters.

If the program is to be selected by inputting the program number in BCD code, bit 2 must be set in the job and the NUL ASCII character must be set in byte 0 in DW3.

If a start block and an end block are defined, bit 3 and bit 4 must be set in the job.

For deselecting a program, only the program number or the program name need to be entered.

The option as to whether a program is to be selected or an active program is to be deselected is made by setting bit 6 or 7 in the job.

The size of the data area required for the entire function is defined in double word DW1.

Because the **-B03PANAB** module does not return any data, no offset needs to be defined in double word DW2.

Specification of the channel number (as an integer), likewise defined in double word DW2, is, however, absolutely necessary.

The program number is input in BCD or binary code in double word DW3 (provided that a part program with a sequence of numbers and a preceding P, such as **P1234**, has been stored in the part program memory).

The **-B03PANAB** module converts the program number into ASCII characters, as the part programs in PNC-R are stored not numerically but according to character.

If a program to be selected is specified by its name in the form of an ASCII character string, please note that the program name is limited to 29 characters, with a trailing NUL ASCII character for termination. If the search path for selecting a program is to be different from the NC-internal search path defined in the MACODA parameters, the respective directory containing the program can be transferred together with the program name. However, the input length limited to 30 characters must not be exceeded by the program name plus the name of the respective directory. If you want, e.g., the **test.cpl** program always to be selected from the user FEPROM, the program name to be transferred must read: **“/usrfep/test.cpl”**.

If the part program is to be started at a certain block, the start block can be entered at the end of the program name, with a trailing NUL ASCII character for termination. The start block can be an NC block according to DIN 66025 (always begins with “N”) or a CPL block (no “N” in front of the block number).

The start block must thus be specified as a character string. The start block length is limited to 9 ASCII characters, with a trailing NUL ASCII character for termination.

If execution of a part program is to be stopped at some point prior to its completion (M30), an end block may be defined. The same rules as for defining start blocks apply to defining end blocks.

If a new program is to be selected while another one is still active, all you have to do is select the new program because function no. 5 or no. 6 is already specified in the job byte. The previous program is deselected automatically provided the NC is in **ready** or **stopped** status.

12.3.2 Configuration example

NC part program P1327 is to be activated on channel 1. Because the program name consists of the letter P and a sequence of numbers, you can select the program by entering its program number in BCD code.

The data area is to begin at byte 28 in data module DM1.

Module **-B03PANAB** could be called as follows:

```
AN    B     I0.3             ;
A     B     M303.0
R     B     M303.0           ; resetting the edge-trigger marker
CM          -DM01PANA        ; call data module DM1
A     B     M303.7           ; evaluate answer ?
JPC         -checkRsp
A     B     I0.3             ; scan input 0.3,
AN    B     M303.0           ; formation of positive edge
EMI
S     B     M303.0           ; setting the edge-trigger marker
S     B     M303.7           ; setting the evaluate answer marker
S     B     D28.2            ; program number in BCD code
S     B     D28.7            ; definition of program selection
S     B     D28.0            ; setting start function
L     DW    K60,A            ; definition of parameters to be transferred
T     DW    A,D32            ; total length of data area = 60 bytes,
L     W     K0,A             ; no offset required as no data
T     W     A,D36            ; are returned,
L     W     K1,A             ; program selection/deselection
T     W     A,D38            ; to be made on channel 1
L     DW    K1327H,A         ; program number to be input
T     DW    A,D40            ; in BCD code,
L     BY    K0H,A            ; complete program number input
T     BY    A,D44            ; with trailing NUL ASCII character
CM          -B03PANAB,2      ; -B03PANAB module call
P0          K28              ; the data area begins at data byte 28
P1          K1               ; in data module DM1
EM
            -checkRsp
L     DW    D28,A            ; job, status and error codes in reg. A
A     B     A.12             ; wait while function is still active
EMC
A     B     M303.7
R     B     M303.7           ; resetting of evaluate answer
A     B     A.15             ; error flag set
EMI                          ; error handling
EM
```

12.3.3 Status messages

If an error or status occurs after calling the ext. program selection/deselection function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$: Parameter error caused by incorrect parameter input in the data area.
FFFE$_H$: Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.
FFFD$_H$: Data overflow, the reserved data area is used to capacity, not enough space to store all the data.
FFFC$_H$: Module **B01APSMN** has not been called.
FFFB$_H$: Incorrect offset: The offset is too large (> reserved area) or too small (overwriting of job data occurs).
8000$_H$: NC-internal error, cannot be influenced by user.
7FFF$_H$: PxRos error, cannot be influenced by user.
7FFE$_H$: NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

1: Invalid channel number.

Errors or statuses occurring upon program selection:

2$_H$: Channel is busy.
3$_H$: Control reset is currently being executed on the channel.
4$_H$: Program name is too long.
5$_H$: Axes have not been referenced.
6$_H$: Program cannot be opened.
7$_H$: Error during program selection.
8$_H$: Either the wrong operation mode or none at all has been selected.

Function-specific errors and statuses occurring upon program deselection:

1$_H$: Invalid channel number.
3$_H$: Control reset is currently being executed on the channel.
0A$_H$: No program has been selected.
0B$_H$: Program is running and feed hold is not present.
19$_H$: Syntax error in the selected program.

12.4 NC block input (-B04SATZV) program module

☞ **This program module is not available for KNS.**

Function

The **NC block input** module facilitates the input of an NC block via the PLC sequential program. This enables the PLC program, e.g., to transfer set-point values of auxiliary axes or auxiliary functions to the NC. NC block input can be made even while the program is running provided no axis addresses are input that would have an impact on the running machining process.

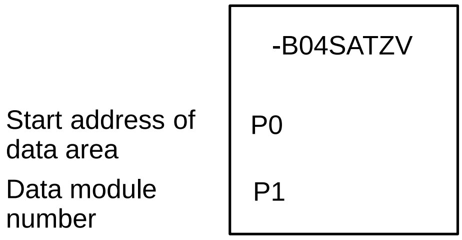
12.4.1 Module call

The module can be called unconditionally or conditionally.

If the module is called repeatedly, please note that a different data area must be parameterized for each call.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B04SATZV**, the data area, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 0

DW0	Error code (2 bytes)	Status (1 byte)	Job (1 byte)
DW1	Length of the total data area reserved (4 bytes)		
DW2	Channel number	Offset from the beginning of the data area to the 1st data feedback address	
DW4 ... DWx	Max. length of NC block: 128 ASCII characters, with a trailing NUL ASCII character (00$_H$) for termination		

The following functions can be activated in the job:

Bit	**Function**
0	Start function
1	reserved

The function is started when bit 0 is set to logic 1 in the job byte.
Bit 4, **function active**, remains set in the status until the function has been carried out, at least, however, until the subsequent call of the **-B01APSMN** program module.

The start function is reset:

- automatically when the function is completed, or
- via the PLC sequential program, upon setting ”function active” (bit 4 in the status).

As soon as NC block input has been executed, this function is completed and bit 4, **function active**, in the status byte is reset. Execution of NC block input is complete when the respective block has become active. If NC blocks are input that involve asynchronous motions the end point of which has not been reached yet, this ensures that the axis travel command is already set at the time the function is completed.

Because the **-B04SATZV** module does not return any data, no offset needs to be defined in double word DW2.
Specification of the channel number (as an integer), likewise defined in the double word DW2, is, however, absolutely necessary.

☞ **For entering asynchronous motions, you can also use the internal ”Automatic start-up channel”. It has the channel number <number of channels>+1. ”Number of channels” is defined in MACODA parameter 9040 00001.**

The NC block is input from double word DW3 on.
Please note that the NC block must end in a trailing NUL ASCII character (00$_H$).

The NC executes the NC block immediately after receipt without an additional NC start signal.

12.4.2 Configuration example

NC block **B123** is to be transferred on channel 1. The data area is to begin at byte 4 in data module DM4.

Module **-B04SATZV** could be called as follows:

```
AN     B      I0.4              ;
A      B      M304.0
R      B      M304.0            ; resetting the edge-trigger marker
CM            -DM4SatzV         ; call data module DM4
A      B      M304.7            ; evaluate answer ?
JPC           -checkRsp
A      B      I0.4              ; scan input 0.4,
AN     B      M304.0            ; formation of positive edge
JPI           -end
S      B      M304.0            ; setting the edge-trigger marker
S      B      M304.7            ; setting the evaluate answer marker
S      B      D4.0              ; setting start function
L      DW     K128,A            ; total length of data area = 128 bytes
T      DW     A,D8
L      W      K0,A              ; no offset required as no data
T      W      A,D12             ; are returned,
L      W      K1,A              ; channel number 1
T      W      A,D14
L      DW     K’321B’,A         ; note order of characters “B123”!
T      DW     A,D16
L      BY     K0,A              ; end of string
T      BY     A,D20
CM            -B04SATZV,2       ; call module -B04SATZV
P0            K4                ; the data area begins at data byte 4
P1            K4                ; in data module DM4
EM
              -checkRsp
L      DW     D4,A              ; job, status and error codes in reg. A
A      B      A.12              ; wait while function is still active
EMC
A      B      M304.7
R      B      M304.7            ; resetting of evaluate answer
A      B      A.15              ; error flag set
EMI                             ; error handling
EM
```

12.4.3 Status messages

If an error or status occurs after calling the NC block input function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$: Parameter error caused by incorrect parameter input in the data area.
FFFE$_H$: Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.
FFFD$_H$: Data overflow, the reserved data area is used to capacity, not enough space to store all the data.
FFFC$_H$: Module **B01APSMN** has not been called.
8000$_H$: NC-internal error, cannot be influenced by user.
7FFF$_H$: PxRos error, cannot be influenced by user.
7FFE$_H$: NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

1$_H$: Invalid channel number.
2$_H$: Channel is busy.
3$_H$: The channel is just being reset.
4$_H$: NC block is too long.
5$_H$: Axes have not been referenced.
8$_H$: Either the wrong operation mode or none at all has been selected.
19$_H$: In the case of an NC block input with automatic start, a runtime error, e.g. a syntax error, was detected.

12.5 Temperature compensation (-B05TKOMP) program module

Function

The **temperature compensation** program module allows the user to set a compensation value via the PLC sequential program to offset temperature impacts. The PLC program transfers the temperature compensation value for one axis at a time (as of V5.1.8 also for **several axes** at a time) with temperature compensation values becoming active immediately upon transfer. Compensation values are transferred on a 0.0001 mm scale and are applied irrespective of axis positions.

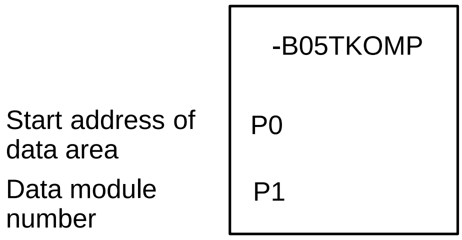
12.5.1 Module call

The module can be called unconditionally or conditionally.

If the module is called repeatedly, please note that a different data area must be parameterized for each call.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B05TKOMP**, the data area, the address of which is defined in parameter P0, has the following structure:

	MSB Bit 31 . . .		LSB 0
DW0	Error code (2 bytes)	Status (1 byte)	Job (1 byte)
DW1	Length of the total data area reserved (4 bytes)		
DW2	Channel number	Offset from the beginning of the data area to the 1st data feedback address	
DW3	Number of compensation values	Physical axis index	
DW4	1st compensation value (scaled to 0.1µ)		
:	:		
DWx	nth compensation value (scaled to 0.1µ)		

The following functions can be activated in the **job**:

Bit	**Function**
0	Start function
1–3	reserved
4–7	Function number

Permissible parameters:

0: temperature compensation input for one axis only (function available up to V5.1.7)

1: temperature compensation input for one or several axes (function available as of V5.1.8)

This function is activated when bit 0, **start function**, is set in the job byte and remains set for at least one PLC cycle.

Bit 0 may be reset via the PLC sequential program with the **function active** message.
Otherwise, the **start function** signal is reset automatically when the function has been carried out.

Bit 4, **function active**, remains set in the status until the function has been carried out, at least, however, for the duration of the PLC cycle.
When the NC has acknowledged the function, the function is completed and bit 4, **function active**, is reset in the status.

Because the **-B05TKOMP** module does not return any data, no **offset** needs to be defined in double word DW2.

Likewise, no **channel number** needs to be specified in double word DW2 because this function is axis-related.

The **physical axis index** and the **number of compensation values** must be entered in double word DW3. The NC takes the temperature compensation values transferred to it into account as of the axis with the index ”axis index”. The axis indices start at 0 for the 1st axis.

☞ **If a temperature compensation value is entered for just one axis (function number 0), 0 must be entered for the number of compensation values. This allows for the module to be integrated in an existing PLC program without first having to make any changes in the PLC user program.**

The temperature compensation is immediately accepted by the NC.

12.5.2 Configuration example

To compensate the temperature impact, a compensation value of 100 µ is to be input for the 4th axis.

The data area is to begin at byte 12 in data module DM5.

Module **-B05TKOMP** could be called as follows:

```
AN   B    I0.5
A    B    M305.0
R    B    M305.0          ; resetting the edge-trigger marker
CM        -DM5TKOMP       ; call data module DM5
A    B    M305.7          ; evaluate answer ?
JPC       -checkRsp
A    B    I0.5            ; scan input 0.5
AN   B    M305.0          ; formation of positive edge
JPI       -end
S    B    M305.0          ; setting the edge-trigger marker
S    B    M305.7          ; setting the evaluate answer marker
S    B    D12.0           ; setting start function
L    DW   K20,A           ; definition of parameters to be transferred
T    DW   A,D16           ; total length of data area = 20 bytes,
L    W    K0,A            ; no offset required as no data
T    W    A,D20           ; is returned,
L    W    K-1,A           ;no channel no. because function is axis-related
T    W    A,D22
L    DW   K4,A            ; for the 4th axis
T    DW   A,D24
L    W    K1000,A         ; 100 µ (0.1 µ scaling)
T    W    A,D28
CM        -B05TKOMP,2     ; call module -B05TKOMP
P0        K12             ; the data area begins at data byte 12
P1        K5              ; in data module DM5
JP        -end
          -checkRsp
L    DW   D12,A           ; job, status and error codes in reg. A
A    B    A.12            ; wait while function is still active
JPC       -end
A    B    M305.7
R    B    M305.7          ; resetting of evaluate answer
A    B    A.15            ; error flag set
                          ; error handling
JP        -end            ; processing of received data
          -end
EM
```

12.5.3 Status messages

If an error or status occurs after calling the **temperature compensation** function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code displayed in binary code indicates the kind of error or status that has occurred.

There are two categories of errors or statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$: Parameter error caused by incorrect parameter input in the data area.

FFFE$_H$: Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.

FFFD$_H$: Data overflow, the reserved data area is used to capacity, not enough space to store all the data.

FFFC$_H$: Module **B01APSMN** has not been called.

8000$_H$: NC-internal error, cannot be influenced by user.

7FFF$_H$: PxRos error, cannot be influenced by user.

7FFE$_H$: NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

10$_H$: Too many temperature compensation values (displayed with the axis number, if applicable)

18$_H$: Invalid axis number

12.6 Logic analyzer (-B06LGANA) program module

Function

The **logic analyzer** program module contains the entire logic for recording the logic analyzer signals within the diagnostics on the NC side.

If this diagnostics function is used, the module must be included in the PLC program.

12.6.1 Module call

☞ **The logic analyzer is called from within the APS modules and, therefore, must not be called from the user program. Otherwise, the NC would stop operations with a system error.**

12.7 Read MACODA parameters (-B07MPARA) program module

Function

The **read MACODA parameters (-B07MPARA)** program module returns the values of a MACODA parameter (block).

The memory space required in the data module or marker field must be defined sufficiently large for all individual parameters to be stored there.

Because this module does not convert value formats, real values are returned as real numbers.

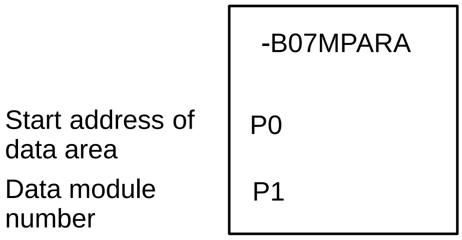
12.7.1 Module call

This module may be called once or repeatedly in a PLC cycle, and calls may be unconditional or conditional.

If the module is called again before actual data feedback from the previous call is received, please note that a different data area must be activated.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B07MPARA**, the data area, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 0

DW0	Error code (2 bytes)	Status (1 byte)	Job (1 byte)
DW1	Length of the total data area reserved (4 bytes)		
DW2	Channel number	Offset from the beginning of the data area to the 1st data feedback address	
DW3	No. of MACODA parameter (block)		
DW4	1st individual parameter returned		
. . . . . .	:		
DWx	last individual parameter returned		

This function is activated when bit 0, **start function**, is set in the job byte and remains set for at least one PLC cycle.

Bit 0 may be reset via the PLC sequential program with the **function active** message.

Otherwise, the **start function** signal is reset automatically when the function has been carried out.

Bit 4, **function active**, remains set in the status until the function has been carried out, at least, however, for the duration of the PLC cycle.

When the data has been returned, the function is completed and bit 4, **function active**, is reset in the status.

Because the **-B07MPARA** module does return data, an offset in double word DW2 must be defined, indicating the beginning of the returned data.

No channel number needs to be specified in double word DW2 because this function is not channel-related.

The number of the MACODA parameter (block) to be read must be entered in double word DW3.

12.7.2 Configuration example

The values of MACODA parameter 2060 00001 are to be scanned upon the formation of the positive edge at PLC input I0.7.

The data area is to begin at marker 700.

Given these requirements, a **-B07MPARA** program module call may be designed as follows:

```
AN   B    I0.7
A    B    M307.0
R    B    M307.0           ; resetting the edge-trigger marker
A    B    M307.7           ; evaluate answer ?
JPC       -checkRsp
A    B    I0.7             ; scan input 0.7,
AN   B    M307.0           ; formation of positive edge
EMI
S    B    M307.0           ; setting the edge-trigger marker
S    B    M307.7           ; setting the evaluate answer marker
S    B    M700.0           ; setting start function
L    DW   K24,A
T    DW   A,M704           ; total length of data area = 24 bytes,
L    W    K16,A            ; 16-byte offset from the beginning of the data
T    W    A,M708           ; area to 1st individual parameter
L    W    K-1,A            ; no channel number
T    W    A,M710
L    DW   K206000001,A     ; requesting MACODA parameter
T    DW   A,M712           ;  no. 2060 00001
CM        -B07MPARA,2      ; call module -B07MPARA
P0        M700             ; the data area begins at M700
P1        K-1              ; not in a data module
EM
          -checkRsp
L    DW   M700,A           ; job, status and error codes in reg. A
A    B    A.12             ; wait while function is still active
EMC
A    B    M307.7
R    B    M307.7           ; resetting of evaluate answer
A    B    A.15             ; error flag set
JPI       -FktOk
                           ;    error handling
EM
          -FktOk           ;    processing of received data
EM
```

The individual parameters returned are stored as follows:

1st individual parameter on markers	716...719,
2nd individual parameter on markers	720...723,
3rd individual parameter on markers	724...727.

12.7.3 Status messages

If an error or status occurs after calling the **read MACODA parameter** function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_{H}$: Parameter error caused by incorrect parameter input in the data area.
FFFE$_{H}$: Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.
FFFD$_{H}$: Data overflow, the reserved data area is used to capacity, not enough space to store all the data.
FFFC$_{H}$: Module **B01APSMN** has not been called.
8000$_{H}$: NC-internal error, cannot be influenced by user.
7FFF$_{H}$: PxRos error, cannot be influenced by user.
7FFE$_{H}$: NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

72 : No MACODA parameter (block) defined

12.8 Open/close serial interface (-B10SEROP) program module

Function

The **open/close serial interface** program module facilitates the opening and closing of a serial interface of the NC.
For opening a serial interface, the interface parameters to be used to initialize the selected interface may be entered optionally. If no such parameters are entered, the interface parameters used in the previous opening process or those defined in MACODA are applied.

☞ **The PNC-P does not support accesses to the serial interface, i.e. this module must not be used with the PNC-P.**

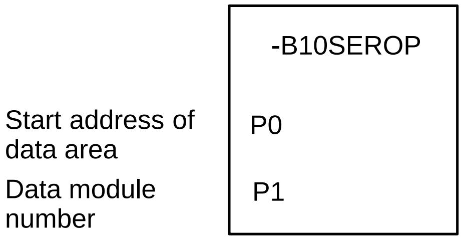
12.8.1 Module call

This module may be called once or repeatedly in a PLC cycle, and calls may be unconditional or conditional.

If the module is called again before the data from the previous call is received, please note that a different data area must be activated.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B10SEROP** program module, the data area, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 . . . . 16 15 0

Error code (2 bytes)	Status (1 byte)	Job (1 byte)
Length of the total data area reserved (4 bytes)		
Channel number	Offset	
Driver	Interface	
Baud rate		
Log. interface parameters	Data format	No. of data bits
Abort string (4 bytes)		
Read-in timeout [s]		
Read-out timeout [s]		
Identifier		
:		
Data		

The following functions can be activated in the job:

Bit	**Function**
0	Start function
1-3	reserved
4-7	Function number: 1: Opening the interface with unchanged parameters 2: Opening the interface with new interface parameters 3: Closing the interface

Start function

This function is activated when bit 0, **start function**, is set in the job byte and remains set for at least one PLC cycle. Bit 0 may be reset via the PLC sequential program with the **function active** message. Otherwise, the **start function** signal is reset automatically when the function has been carried out.

Status

Bit 4, **function active**, remains set in the status until the function has been carried out, at least, however, for the duration of one PLC cycle. When the interface has been opened or closed, the function is completed and bit 4, **function active**, is reset in the status.

Size of data area

In order to prevent accidental overwriting of data, the user enters here the maximum number of bytes expected to occur with this function.

Offset

The offset is the space between the beginning of the area (job) and the address where the received data identification is to be stored or, resp., where the identification for closing the interface is stored.

Channel no.

This parameter is irrelevant to this module.

Interface no.

The interface number can be used to select one of the serial interfaces for the function.

The following values are permitted:

1 - Interface X32 on the BOR module
2 - Interface X31 on the BOR module
4 - Interface X35 on the osa master module

Driver number

Only the TTY driver can be selected at present for data exchange via a serial interface (driver number = 0).

☞ **The following parameters are required only if also the interface parameters are to be transferred in the process of opening an interface. These parameters are irrelevant to any other function.**

Baud rate

Baud rate at which the interface is to operate.

The following values are permitted:

110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bauds

Number of data bits

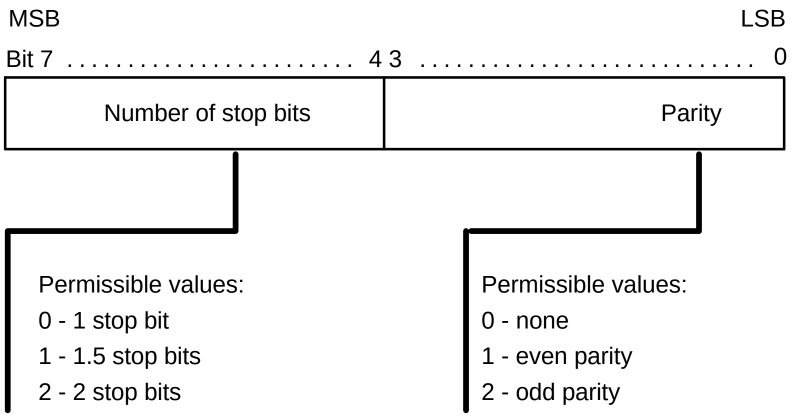
This parameter defines the number of data bits assigned to every packet received.

The following values are permitted:

5, 6, 7, 8

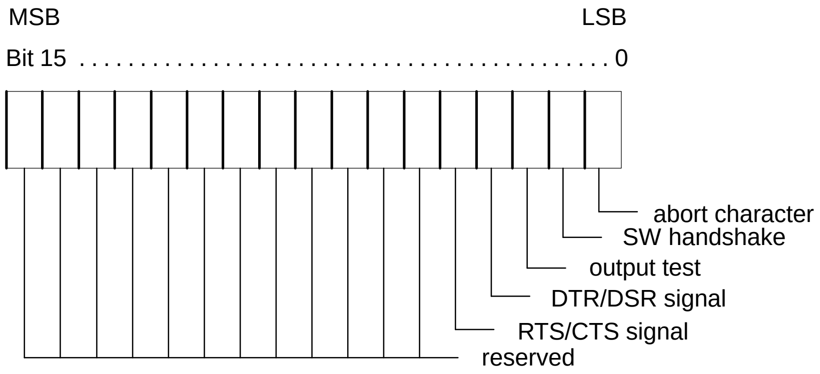
Data format (parity, number of stop bits)

This parameter is used to define the parity and the number of stop bits.



Logical interface parameters

Serial interface performance can be controlled by setting the logic interface parameters. Setting the respective bit will activate the function.


MSB
LSB
Bit 15 0
abort character
SW handshake
output test
DTR/DSR signal
RTS/CTS signal
reserved

Abort character: When the respective control character (< 20_H or > $7f_H$) is received, the loading process is aborted and the characters read in by then are transferred.

SW handshake (XON/XOFF): The interface is controlled by software handshake.

Output test: For test purposes, all control characters are output as displayable characters (e.g., “<0x24>” for the output of a <CR> character at the line end)

DTR/DSR signals: For hardware handshakes, the DSR and DTR signal lines are used for data transfer control.

RTS/CTS signals: For hardware handshakes, the RTS and CTS signal lines are used for data transfer control.

Abort character

Up to 4 characters may be defined to control data traffic. When they are received, the loading process is aborted and the characters read in by then are transferred to the user together with the abort character being the last character received.

Read-in timeout

The check time for data read-in must be entered in seconds. If a timeout occurs in the course of a read-in process, the characters read-in by then are transferred to the user.

Read-out timeout

The check time for data read-out must be entered in seconds. If a timeout occurs in the course of a read-out process, the characters read-out by then are transferred to the user.

Identifier

The function returns an identifier in acknowledgement of the successful execution of the interface opening process. This identifier must be entered for all interface functions (read-in, read-out, closing). The memory area where this identifier is stored results from the start address of the data area transferred and the offset entered.

12.8.2 Configuration example

Example:

Interface X32 on the BOR module is to be opened with the following parameters:

9600 bauds, 8 data bits, even parity and 2 stop bits. Data flow is to be controlled by software handshake (XON/XOFF):

The function is to be started with the rising edge on input I10.0. The data area containing the job-specific parameters is to begin at byte 12 in data module DM10.

Module **-B10SEROP.PAH** could be called as follows:

```
AN    B     I10.0
A     B     M310.0
R     B     M310.0          ; resetting the edge-trigger marker
CM          -DM10serOp      ; call data module DM10
A     B     M310.7          ; evaluate answer ?
JPC         -checkRsp
A     B     I10.0           ; scan input 10.0
AN    B     M310.0          ; formation of positive edge
EMI
S     B     M310.0          ; setting the edge-trigger marker
S     B     M310.7          ; setting the evaluate answer marker
L     BY    K2,A            ; open with input of the interface
                            ; parameters
SLL   BY    A,4             ; function no. is entered in bit 4 in the job
O     BY    K1,A            ; setting start function
T     BY    A,D12           ; save job byte
L     DW    K64,A           ; definition of parameters to be transferred
T     DW    A,D16           ; total length of data area = 64 bytes,
L     W     K36,A           ; offset from beginning of data area
T     W     A,D20           ; to identifier
L     W     K-1,A           ; no channel-specific function
T     W     A,D22
```

```
L      W      K4,A              ; interface X35 on the osa master module
T      W      A,D24
L      W      K0,A              ; TTY driver
T      W      A,D26
L      DW     K9600,A           ; 9600 bauds
T      DW     A,D28
L      BY     K8,A              ; 8 data bits
T      BY     A,D32
L      BY     K1,A              ; even parity
L      BY     K2,B              ; 2 stop bits
SLL    BY     B,4               ; display stop bits in the upper half
                                ; of the byte
O      BY     B,A
T      BY     A,D33
L      W      K2,A              ; SW handshake (XON/XOFF)
T      W      A,D34
L      DW     KFFFFFFFFH,A      ; no abort characters
T      DW     A,D36
L      DW     K2,A              ; read-in timeout = 2 s
T      DW     A,D40
L      DW     K2,A              ; read-out timeout = 2 s
T      DW     A,D44
CM            -B10SEROP,2       ; call module B10SEROP.PAH
P0     K12                      ; the data area begins at data byte 12
P1     K10                      ; in data module DM10
EM
       -checkRsp
L      DW     D12,A             ; job, status and error codes in reg. A
A      B      A.12              ; wait while function is still active
EMC
A      B      M310.7
R      B      M310.7            ; resetting of evaluate answer
A      B      A.15              ; error flag set
EMI                             ; error handling
EM
```

12.8.3 Status messages

If an error or status occurs after calling the **open/close serial interface** function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$: Parameter error caused by incorrect parameter input in the data area.
FFFE$_H$: Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.
FFFD$_H$: Data overflow, the reserved data area is used to capacity, not enough space to store all the data.
FFFC$_H$: Module **B01APSMN** has not been called.
FFFB$_H$: Incorrect offset: The offset is too large (> reserved area) or too small (overwriting of job data occurs).
8000$_H$: NC-internal error, cannot be influenced by user.
7FFF$_H$: PxRos error, cannot be influenced by user.
7FFE$_H$: NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

2 : Invalid job parameters
3 : Invalid identifier
7 : Device driver does not exist
8 : Invalid interface number
9 : Interface is already open (possibly by the operator via the operator interface or by the DNC)
10 : Changing the interface parameters was canceled when the interface was closed.
12 : There is no operational equipment available at present to which this function could be applied
35 : Other error within the input/output logic

12.9 Data read-in/read-out (-B11SERIO) program module

Function

The **data read-in/read-out** program module allows data to be read in and read out via a serial NC interface.

☞ **The PNC-P does not support accesses to the serial interface, i.e. this module must not be used with the PNC-P.**

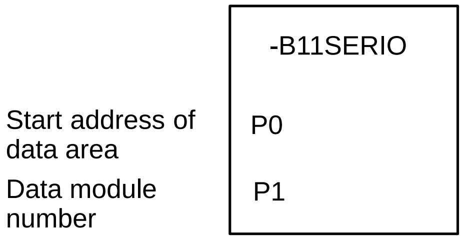
12.9.1 Module call

This module may be called once or repeatedly in a PLC cycle, and calls may be unconditional or conditional.

If the module is called again before the data from the previous call is received, please note that a different data area must be activated.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B11SERIO**, the data area, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 . . . . 16 15 0

Error code (2 bytes)	Status (1 byte)	Job (1 byte)
Length of the total data area reserved (4 bytes)		
Channel number	Offset	
Identifier		
Volume of data to be read in/read out (in bytes)		
Data to be read out / volume of data to be read in		
Data read-in		

The following functions can be activated in the job:

Bit	**Function**
0	Start function
1-3	reserved
4-7	Function number:

1:	Reading-in data at the serial interface
2:	Reading-out data at the serial interface
3:	Cancellation of data read-in
4:	Cancellation of data read-out

This function is activated when bit 0, **start function**, is set in the job byte and remains set for at least one PLC cycle.

Bit 0 may be reset via the PLC sequential program with the **function active** message.

Otherwise, the **start function** signal is reset automatically when the function has been carried out.

Status

Bit 4, **function active**, remains set in the status until the function has been carried out, at least, however, for the duration of one PLC cycle.

When the data has been loaded or output or the respective transaction has been cancelled, the function is completed and bit 4, **function active**, is reset in the status.

Size of data area

In order to prevent accidental overwriting of data, enter here the maximum number of bytes you want to provide for this function.

Offset

The offset is the space between the beginning of the area (job) and the address where the received data is to be stored or, resp., where the data to be read out is stored.

Channel no.

This parameter is irrelevant to this module.

Identifier

The identifier which was returned when the interface was opened.

Volume of data to be read in / read out

Data volume in bytes.

Output data / number of characters read in

When data is read in, the number of data bytes read-in is stored here.

When data is to be read out to the interface, data is transferred from this point on.

Received data

When data is read in, the data bytes received are stored here.

12.9.2 Configuration example

A total of 64 characters are to be loaded from the serial interface.

The identifier that was returned when the interface was opened was entered in D48 in data module DM10. The data area containing the job parameters is to begin at byte 0 in data module DM11 and the function is to be started with the rising edge on input I11.0.

Module **-B11SERIO.PAH** could be called as follows:

```
AN    B     I11.0
A     B     M311.0
R     B     M311.0          ; resetting the edge-trigger marker
CM          -DM11serIO      ; call data module DM11
A     B     M311.7          ; evaluate answer ?
JPC         -checkRsp
A     B     I11.0           ; scan input 11.0,
AN    B     M311.0          ; formation of positive edge
EMI
S     B     M311.0          ; setting the edge-trigger marker
S     B     M311.7          ; setting the evaluate answer marker
L     BY    K1,A            ; data read-in
SLL   BY    A,4             ; function no. is entered in bit 4 in the job
O     BY    K1,A            ; setting start function
T     BY    A,D0            ; save job byte
L     DW    K88,A           ; total length of data area = 88 bytes,
T     DW    A,D4
L     W     K20,A           ; offset from beginning of data area to identifier
T     W     A,D8            ;
L     W     K-1,A           ; no channel-specific function
T     W     A,D10
CX    DM10                  ; open 2nd data module
L     DW    DX48,A          ; read identifier from DM10
T     DW    A,D12
L     DW    K64,A           ; request read-in of 64 characters
T     DW    A,D16
CM          -B11SERIO,2     ; call module B11SERIO.PAH
P0    K0                    ; the data area begins at data byte 0
P1    K11                   ; in data module DM11
EM
      -checkRsp
L     DW    D0,A            ; job, status and error codes in reg. A
A     B     A.12            ; wait while function is still active
EMC

A     B     M311.7

R     B     M311.7          ; resetting of evaluate answer

A     B     A.15            ; error flag set

JPC   -error

                            ; processing of the data received

EM

      -error                ; error handling

EM
```

12.9.3 Status messages

If an error or status occurs after calling the **data read-in/read-out** function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$: Parameter error caused by incorrect parameter input in the data area.

FFFE$_H$: Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.

FFFD$_H$: Data overflow, the reserved data area is used to capacity, not enough space to store all the data.

FFFC$_H$: Module **B01APSMN** has not been called.

FFFB$_H$: Incorrect offset: The offset is too large (> reserved area) or too small (overwriting of job data occurs).

8000$_H$: NC-internal error, cannot be influenced by user.

7FFF$_H$: PxRos error, cannot be influenced by user.

7FFE$_H$: NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

2 : Invalid job parameters

3 : Invalid identifier

10 : Data read-in/read-out was canceled when the interface was closed.

12 : There is no operational equipment available at present to which this function could be applied

20: Timeout. Data read-in before the timeout has occurred is transferred to the user. Thus, the user can examine the timeout process by comparing the data volume set in the job parameters and the number of characters read-in.

21 : Overrun error in UART

22 : Parity error in UART

23 : Framing error in UART

35 : Other error within the input/output logic

12.10 Program module handwheel data (-B14HANDW)

Function

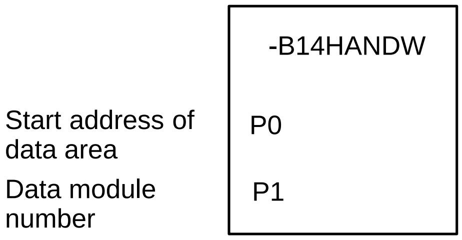
For handwheels in connection with PROFIBUS-DP, the **handwheel data** program module transmits the current value of the handwheel including handwheel number to the NC.
Since the data are transmitted via PROFIBUS-DP, the data from the input area have to be transferred to the NC with the aid of this APS module.

12.10.1 Module call

This module may be called once or repeatedly in a PLC cycle, and calls may be unconditional or conditional.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

P1 contains the data module number if the data area is located in a data module. In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K–1 must be entered in parameter P1.

This function is activated when bit 0, start function, is set in the job byte and remains set for at least one PLC cycle.
Bit 0 may be reset via the PLC sequential program with the function active message. Otherwise, the start function signal is reset automatically when the function has been carried out.

In the case of program module -B14HANDW, the data area, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 . . . . 16 15 0

Error code (2 bytes)	Status (1 byte)	Job (1 byte)
Length of the entire reserved data area (4 bytes)		
Channel no.	Offset	
Input value of the handwheel	Handwheel number	

Job

This function is activated when bit 0, **start function**, is set in the job byte and remains set for at least one PLC cycle.
Bit 0 may be reset via the PLC sequential program in dependence on the **function active** message. Without resetting via the PLC sequential program, **start function** is reset automatically when the function has been carried out.

Status

The value of status indicates the current condition of the job:

Bit	**Status**
0-3	reserved
4	function active
5-6	reserved
7	error flag

As long as bit 4 function active is set, the job has not been completed.
If an error occurs during execution of a job, the module sets bit 7 error flag.

Error code

If an error occurs, the module transfers the error code indicating the cause of the error (see page 12–42).

Size of data area

To prevent data from being overwritten accidentally, the maximum number of bytes, earmarked for this function, are transmitted here.

Offset channel no.

Not relevant for this module.

Handwheel number

Transmitting the number of the handwheel for which the data are transferred: (Permissible range: 1 – 2)

Input value of the handwheel

Current value in increments.

12.10.2 Configuration example

Handwheel 1 is active and the values of the handwheel are positioned starting at input 56.
The data area is to begin at marker 1400.
The module is called cyclically when the handwheel mode is active.

Module **-B14HANDW.PAH** could be called as follows:

```
A       B       –Handr              ; handwheel mode active ?
JPCI            –end

S       B       M1400.0             ; setting of start function
L       DW      K24,A               ; definition of parameters to be transferred
T       DW      A,M1404             ; total length of data area = 24 bytes
L       W       K–1,A               ; dummy offset
T       W       A,M1408
L       W       K–1,A               ; dummy channel number
T       W       A,M1410
L       W       K1,A                ; handwheel number 1
T       W       A,M1412
L       W       E56,A               ; value of the handwheel
T       W       A,M1414
CM      –B14HANDW                   ; -B14HANDW module call
P0      M1400                       ; the data area begins at marker 1400
P1      K–1                         ; not in a data module

; check whether an error has occurred
; the error check is performed directly after the call because data are transferred
; directly
A       B       A.15                ; error flag set ?
JPCI            –end
;error handling

–end
EM
```

12.10.3 Status messages

If an error or status occurs after calling the function which prevents proper execution of this function, **bit 7** is set in **status** and the type of error or status is represented in the form of a binary-coded number in the **error code**.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC errors:

FFFF$_H$: Parameter error caused by incorrect parameter input in the data area.
FFFE$_H$: Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.
FFFC$_H$: Module **B01APSMN** has not been called.
8000$_H$: NC-internal error, cannot be influenced by user.

Function-dependent errors or statuses occurring with the acknowledgement that the function has been executed:

1 : Invalid handwheel number.

12.11 Read/Input SERCOS ident. number (-B15IDTNR) program module

Function

The program module **read/input SERCOS ident. number** allows reading and inputting ident. numbers from and to the drive modules. The format of values to be read or input is a 4-byte, unsigned integer, i.e. the format of values as saved in the drive modules. Any conversion of these values must be done in the user program. When reading a value, this functions also returns the attribute that describes the ident. number. When inputting values in the drive modules, also the attribute must be entered.

As of V5.1.8 also the values (for all axes) of an ident no. transferred from the cyclic message to the NC can be read. For an ident. number of an axis not listed in the cyclic message, the value 80000000_H is returned for the respective axis.

12.11.1 Module call

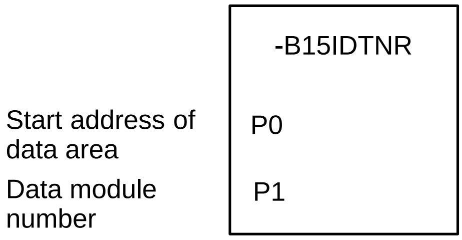
The module can be called unconditionally or conditionally.

If this module is called repeatedly, please note that a different data area must be configured for each call.

☞ **Repeated module calls with requests for data transfer directly on the service channel (non-cyclical data) may strain the capacity of the service channel! Therefore, a 500 ms interval between two module calls is recommended!**

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B15IDTNR**, the data area, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 . . . . 16 15 0

<table>
<tr><td>Error code (2 bytes)</td><td>Status (1 byte)</td><td>Job (1 byte)</td></tr>
<tr><td colspan="3">Length of the total data area reserved (4 bytes)</td></tr>
<tr><td>Channel number</td><td colspan="2">Offset from the beginning of the data area to the 1st address of data returned.</td></tr>
<tr><td colspan="3">Physical axis number</td></tr>
<tr><td colspan="3">Standard <b>(S)</b> / manufacturer-dependent <b>(P)</b> ident. number</td></tr>
<tr><td colspan="3">Ident. number</td></tr>
<tr><td colspan="3">Attribute</td></tr>
<tr><td colspan="3">Value to be read / input</td></tr>
</table>

Read-out from the cyclic message:

As of version 5.2.2, the NC can read the values of ident. number (as specified in the cyclic message transmitted from the drives to the NC) without any **transmission delay time.**

In the case of program module **-B15IDTNR**, the data area for read-outs from the cyclic message, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 . . . . 16 15 0

<table>
<tr><td>Error code (2 bytes)</td><td>Status (1 byte)</td><td>Job (1 byte)</td></tr>
<tr><td colspan="3">Length of the total data area reserved (4 bytes)</td></tr>
<tr><td>Channel number</td><td colspan="2">Offset from the beginning of the data area to the 1st address of data returned.</td></tr>
<tr><td colspan="3">Physical axis number</td></tr>
<tr><td colspan="3">Standard <b>(S)</b> / manufacturer-dependent <b>(P)</b></td></tr>
<tr><td colspan="3">Ident. number</td></tr>
<tr><td colspan="3">Ident. number value of 1st physical axis</td></tr>
<tr><td colspan="3">Ident. number value of 2nd physical axis</td></tr>
<tr><td colspan="3">:</td></tr>
<tr><td colspan="3">Ident. number value of nth physical axis</td></tr>
</table>

The **job byte** is structured as follows:

Bit	**Function**
0	Start function
1-3	reserved
4-7	Function number:

1: Read the attribute and the current value of the ident. no.
2: Input a new ident. no. value.
3: Read the ident. no. attribute only.
4: Read an ident no. from the cyclic message. Starting at the address defined by the specified offset, read-outs from the cyclic message provide the values of all axis ident. numbers transferred. For ident. numbers not listed in the cyclic message, the value 80000000_{hex} is returned for the respective axis. The minimum interval for value read-outs is 100 ms.
5: Same as above, 4, but the values are provided **without** any transmission time delay.

This function is activated when bit 0, **start function**, is set in the job byte and remains set for at least one PLC cycle.

Bit 0 may be reset via the PLC sequential program with the **function active** message. Otherwise, the **start function** signal is reset automatically when the function has been carried out.

Bit 4, **function active**, remains set in the status until the function has been carried out, at least, however, for the duration of one PLC cycle.

When the data has been returned or input, the function is completed and bit 4, **function active**, is reset in the status.

Offset

The offset is the space between the beginning of the area (job) and the address where the attribute and the current value of the ident. no. are to be stored, or, resp., where the attribute and the new value of the ident. no. are currently stored.
With read-outs of ident. numbers from the cyclic telegram, this is where the values (starting with the 1st axis) are stored.

Channel no.

This parameter is irrelevant to this module.

Physical axis number

Physical axis number of the axis whose ident. no. is to be read or input.
This parameter is irrelevant in the context of read-outs from the cyclic message.

Standard/manufacturer-dependent ident. no.

To distinguish between standard and manufacturer-dependent ident. numbers to be read out or input, ASCII characters **S** or **P** must be entered here.

Ident. no.

The ident. no. whose value is to be read or input.

Attribute

Attribute of the returned ident. no.

Value

Current or new value of the specified ident. no.

Ident. no. value of the nth physical axis

Current or new value of the ident. no. read out from the cyclic message.

☞ **To reduce high system loads due to communication, ident. numbers should be read out from cyclic messages using function “4” if the respective application allows this.**

12.11.2 Configuration example

The current value of ident. no. **S-0-0108, Feedrate Override** of physical axis 1 is to be set at 50%. The data area containing the job parameters is to begin at byte 0 in data module DM15 and the function is to be started with the rising edge on input I15.0. The attribute of the respective ident. no. has already been read out and stored beginning with D24 of the data module.

Module **-B15IDTNR.PAH** could be called as follows:

```
AN    B     I15.0
A     B     M315.0
R     B     M315.0         ; resetting the edge-trigger marker
CM          -DM15IDTNR     ; call data module DM15
A     B     M315.7         ; evaluate answer ?
JPC         -checkRsp
A     B     I15.0          ; scan input 15.0,
AN    B     M315.0         ; formation of positive edge
EMI
S     B     M315.0         ; setting the edge-trigger marker
S     B     M315.7         ; setting the evaluate answer marker
L     BY    K2,A           ; shifting the function no. of ident. no. input
SLL   BY    A,4            ; to the upper half of the byte
O     BY    K1,A           ; setting start function
T     BY    A,D0           ; save job byte
L     DW    K32,A          ; total length of data area = 32 bytes,
T     DW    A,D4
L     W     K24,A          ; 24-byte offset to attribute
T     W     A,D8           ; of the ident. number
L     W     K-1,A          ; channel no. is irrelevant
T     W     A,D10
L     DW    K1,A           ; physical axis number 1
T     DW    A,D12
L     W     K’ S’,A        ; standard ident. number (note
T     W     A,D18          ; sequence of characters!)
L     DW    K108,A         ; ident. no. 108
T     W     A,D16
L     DW    K5000,A        ; 50% (unit in 0.01%)
T     DW    A,D28
CM          -B15IDTNR,2    ; module call B15IDTNR.PAH
P0    K0                   ; the data area begins at data byte 0
P1    K15                  ; in data module DM15
      -checksp
```

```
L       DW      D0,A              ; job, status and error codes in reg. A
A       B       A.12              ; wait while function is still active
EMC
A       B       M315.7
R       B       M315.7            ; resetting of evaluate answer
A       B       A.15              ; error flag set
EMI                               ; function completed without errors
        -error
                                  ; error handling
                                  ; e.g., repeat job if the service channel for
                                  ; the drive was busy (error no. 9H).
EM
```

12.11.3 Status messages

If an error or status occurs after calling the **read/input SERCOS ident. number** function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$:	Parameter errors caused by incorrect parameter input in the data area.
FFFE$_H$:	Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.
FFFD$_H$:	Data overflow, the reserved data area is used to capacity, not enough space to store all the data.
FFFC$_H$:	Module **B01APSMN** has not been called.
8000$_H$:	NC-internal error, cannot be influenced by user.
7FFF$_H$:	PxRos error, cannot be influenced by user.
7FFE$_H$:	NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

3$_H$:	NCS channel currently busy
6 $_H$:	Unknown ident. no.
9 $_H$:	Access conflict on service channel
E $_H$:	Handshake timeout
F$_H$:	Busy timeout
16$_H$:	Data transmitted was too short
17 $_H$:	Data transmitted was too long
18 $_H$:	Element cannot be changed
19 $_H$:	Element cannot be changed at the moment (different phase)
1A $_H$:	Data smaller than min. value
1B $_H$:	Data greater than max. value
1C $_H$:	Invalid operating data (e.g. bit combination)
1D $_H$:	SERCOS ring currently unavailable (run-up)
1E $_H$:	Incorrect attribute received
1F $_H$:	Invalid or non-SERCOS (dummy) system axis
21 $_H$:	Data length exceeds service container size
28 $_H$:	Max. number of simultaneously active commands exceeded
29 $_H$:	Command change bit timeout
32 $_H$:	Conversion error (conversion to SERCOS representation or vice versa)
1F4 $_H$:	System axis not available as drive
1F5 $_H$:	Configuration error (system axis no. and axis no. on ring are inconsistent)

12.12 Edit tool list (-B20WZLST) program module

Function

With the **edit tool list (-B20WZLST)** module, you can request lists of tool data records or delete lists of tool data records.

- **Searching for tool data records**
  To search for tool data records, you may use up to 7 search criteria to be linked by logic AND operators. If less than 7 search criteria are used, 0 must be entered in the search word for any search criteria remaining unused.
  With this module you can also search for the next data record in the memory location that follows a reference memory location you have specified. If no such data record is found, this module returns **-1** for the next memory location number. If no search for the next memory location is to be conducted, -1 must be entered for the reference sector.
- **Deleting tool data records**
  To delete data records, you may specify a list of data records to be deleted. This module will then delete each of these data records by overwriting their contents with 0, with the exception of the sector and location numbers.
  The space between the beginning of the whole data area and the start index address must be entered in the offset. This allows for the same data area to be used both for requesting and deleting a list of tool data records. Thus, the very same tool data list requested on the basis of specific search criteria can then be used to delete these data records.
  With the start index, the user also transfers the index of the 1st data record on the list to be deleted as well as the number of data records to be deleted.

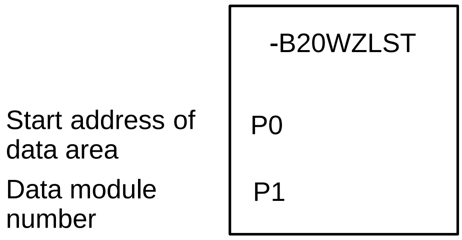
12.12.1 Module call

This module may be called once or repeatedly in a PLC cycle, and calls may be unconditional or conditional.

If the module is called again before the data from the previous call is received, please note that a different data area must be activated.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B20WZLST**, the data area, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 . . . . 16 15 0

Error code (2 bytes)	Status (1 byte)	Job (1 byte)
Length of the total data area reserved (4 bytes)		
Channel number	Offset	
Table no.		
Start sector		
Start location		
Reference sector		
Reference location		
1st search word		
1st search contents		
1st search mask		
2nd search word		
2nd search contents		
2nd search mask		
:		
7th search word		
7th search contents		
7th search mask		
only for search word=6 (query_string) character string (max. 32 bytes)		
Number of data records	Next memory location/start index	
1st data record: location no.	1st data record: sector no.	
2nd data record: location no.	2nd data record: sector no.	
3rd data record: location no.	3rd data record: sector no.	
:	:	
Last data record: location no.	Last data record: sector no.	

The following functions can be activated in the job:

Bit	**Function**
0	Start function
1	reserved
2-7	Function number:

1:	Requesting a tool list, with the tool table to be searched from its very beginning
2:	Requesting a tool list, with the tool table to be searched from the specified start sector/start location on
3:	Deleting data records in accordance with the list
4-7:	reserved

This function is activated when bit 0, **start function**, is set in the job byte and remains set for at least one PLC cycle.

Bit 0 may be reset via the PLC sequential program with the **function active** message.

Otherwise, the **start function** signal is reset automatically when the function has been carried out.

Bit 4, **function active**, remains set in the status until the function has been carried out, at least, however, for the duration of one PLC cycle.

When the data has been returned, the function is completed and bit 4, **function active**, is reset in the status.

Search word

The following values may be entered for selecting a search word:

0	Ignore search word	
1	Sector	
2	Tool number	(query_int 1)
3	Duplo number	(query_int 2)
4		(query_int 3)
5		(query_int 4)
6	Tool name	(query_string)
7	Tool identification	(bit_field)

12.12.2 Configuration example

A search is to be conducted for all tools with tool number 123 in the tool holding magazine (sector 2) the tool life of which has expired.

The area containing the job data is to begin at byte 0 in data module DM20.

The function is to be started with the rising edge on input **I20.0**.

Module **-B20WZLST** could be called as follows:

```
AN    B     I20.0
A     B     M320.0
R     B     M320.0          ; resetting the edge-trigger marker
CM          -DM20WzgL       ; call data module DM20
A     B     M320.7          ; evaluate answer ?
JPC         -checkRsp
A     B     I20.0           ; scan input 20.0,
AN    B     M320.0          ; formation of positive edge
JPI         -end
S     B     M320.0          ; setting the edge-trigger marker
S     B     M320.7          ; setting the evaluate answer marker
S     B     D0.0            ; setting start function
                            ; definition of parameters to be transferred
L     BY K1,A               ; function number for search from
                            ; beginning of table
SLL   BY A,2                ; shifting the function no. to the respective
                            ; location in the job byte
O     BY    K1,A            ; start function
T     BY    A,D0
L     DW    K256,A          ; total length of data area = 256 bytes
T     DW    A,D4
L     W     K116,A          ; offset = 148 bytes
T     W     A,D8,
L     W     K-1,A           ; no channel-specific function
T     W     A,D10
L     DW    K1,A            ; Table no.1
T     DW    A,D12
L     DW    K-1,A           ; no start sector
T     DW    A,D16
L     DW    K-1,A           ; no start location
T     DW    A,D20
L     DW    K-1,A           ; no reference sector
T     DW    A,D24
L     DW    K-1,A           ; no reference location
T     DW    A,D28
L     DW    K1,A            ;1st search word is ”sector”
T     DW    A,D32
L     DW    K2,A            ; search is limited to sector 2 (tool magazine)
T     DW    A,D36
L     DW    K0,A            ; no mask, as search word not “identifier”
T     DW    A,D40
L     DW    K2,A            ; 2nd search word is “tool number”
T     DW    A,D44
L     DW    K123,A          ; search for tool number 123
T     DW    A,D48
L     DW    K0,A            ; no mask, as search word not “identifier”
T     DW    A,D52
L     DW    K7,A            ; 3rd search word is “identifier”
T     DW    A,D56
L     DW    K4H,A           ; search for tools identified as ”defective”
```

```
T      DW     A,D60
L      DW     K4,A               ; defective tool identifier bit only
T      DW     A,D64
L      DW     K0,A               ; 4th search word is ”not used”
T      DW     A,D68
L      DW     K0,A               ; 5th search word is “not used”
T      DW     A,D80
L      DW     K0,A               ; 6th search word is “not used”
T      DW     A,D92
L      DW     K0,A               ; 7th search word is “not used”
T      DW     A,D104
CM            -B20WZLST,2        ; call module -B20WZLST
P0            K0                 ; the data area begins at data byte 0
P1            K20                ; in data module DM20
JP            -end
              -checkRsp
L      DW     D0,A               ; job, status and error codes in reg. A
A      B      A.12               ; wait while function is still active
JPC           -end
A      B      M320.7
R      B      M320.7             ; resetting of evaluate answer
A      B      A.15               ; error flag set
JPI           -FktOk
                                 ; error handling
JP            -end
              -FktOk
                                 ; processing of received data
              -end
EM
```

12.12.3 Status messages

If an error or status occurs after calling the **edit tool list** function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$:	Parameter error caused by incorrect parameter input in the data area.
FFFE$_H$:	Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.
FFFD$_H$:	Data overflow, the reserved data area is used to capacity, not enough space to store all the data.
FFFC$_H$:	Module **B01APSMN** has not been called.
FFFB$_H$:	Incorrect offset: The offset is too large (> reserved area) or too small (overwriting of job data occurs).
8000$_H$:	NC-internal error, cannot be influenced by user.
7FFF$_H$:	PxRos error, cannot be influenced by user.
7FFE$_H$:	NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

68 :	Tool table is empty
72 :	Data record not found
108 :	The last data record was read with *inhibit* and not reenabled
115 :	The last data record was not read with *inhibit*
1001:	Invalid table no.

12.13 Edit tool data (-B21WZDAT) program module

Function

The **edit tool data (-B21WZDAT)** program module offers various functions for reading and changing tool data.

The various functions can be selected by entering the respective function number in the job byte.

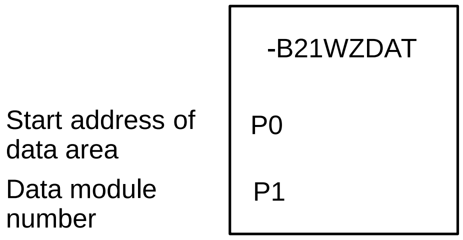
12.13.1 Module call

This module may be called once or repeatedly in a PLC cycle, and calls may be unconditional or conditional.

If the module is called again before the data from the previous call is received, please note that a different data area must be activated.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B21WZDAT**, the data area, the address of which is defined in parameter P0, has the following structure:

<table>
<tr><td>MSB</td><td colspan="2">LSB</td></tr>
<tr><td>Bit 31 . . . . 16</td><td colspan="2">15 . . . . 0</td></tr>
<tr><td>Error code (2 bytes)</td><td>Status (1 byte)</td><td>Job (1 byte)</td></tr>
<tr><td colspan="3">Length of the total data area reserved (4 bytes)</td></tr>
<tr><td>Channel number</td><td colspan="2">Offset</td></tr>
<tr><td colspan="3">Table number (source)</td></tr>
<tr><td colspan="3">File name (with functions 8 or 9 only) / sector number (source)</td></tr>
<tr><td colspan="3">Location number (source)</td></tr>
<tr><td colspan="3">Table number (destination, with functions 5 or 6 only)</td></tr>
<tr><td colspan="3">Sector number (destination, with functions 5 or 6 only)</td></tr>
<tr><td colspan="3">Location number (destination, with functions 5 or 6 only)</td></tr>
<tr><td>Location number</td><td colspan="2">Sector number</td></tr>
<tr><td colspan="3">Tool number (query_int 1)</td></tr>
<tr><td colspan="3">Duplo number (query_int 2)</td></tr>
<tr><td colspan="3">(query_int 3)</td></tr>
<tr><td colspan="3">(query_int 4)</td></tr>
<tr><td colspan="3">Tool identification (bit field)</td></tr>
<tr><td colspan="3">Tool name (query_string)
32 characters</td></tr>
<tr><td colspan="3">Tool data (data_int)
40 values of 4 bytes, ea.</td></tr>
<tr><td colspan="3">32-character string (data_string)</td></tr>
</table>

The job byte is structured as follows:

Bit	**Function**
0	Start function
1	reserved
2-7	Function number:

1: Reading a tool data record without inhibiting further access to it.

2: Reading a tool data record while inhibiting further access to it. This function must be called if tool data is to be saved in the data base after being edited. Until access to the inhibited data record is reenabled, the PLC user program is denied access also to any other data records stored in the data base.

3: Saving a tool data record in the data base. At the same time, this function reenables access to this function.

4: Reenabling access to a data record read previously with function no. 2.

5: Interchanging two data records in the data base, i.e. interchanging the contents of two data records saved in the data base.

6: Copying the contents of a data record (source record) to another data record (destination record).

7: Deleting the contents of a data record (source record).

8: Saving the tool table in an ASCII file. Together with the file name under which the tool table contents is to be saved, you may specify the whole path name, which must not exceed 50 characters (incl. end identifier **NUL** (00_H)). To ensure that the file can be read in also from the operator interface, it must be saved under **/database/tooltab1.dat**. For backup purposes, this file can be output also on a serial interface.

9: Reading in the contents of the tool table from an ASCII file. With this function you can read in an ASCII file created by operating interface input or with function **8**.

☞ **The function described above overwrites the contents of the current tool table.**

10: Deleting all data records of the tool table. This function deletes the entire table. Subsequently, an NC runup must be carried out which will cause a new tool table to be created and changes in the configuration file for the tool table to be activated.

This function is activated when bit 0, **start function**, is set in the job byte and remains set for at least one PLC cycle.

Bit 0 may be reset via the PLC sequential program with the **function active** message.

Otherwise, the **start function** signal is reset automatically when the function has been carried out.

Bit 4, **function active**, remains set in the status until the function has been carried out, at least, however, for the duration of one PLC cycle.

When the data has been returned, the function is completed and bit 4, **function active**, is reset in the status.

12.13.2 Configuration example

The data of the tool in location 5, sector 2, table 1 is to be read while any further access to this data is to be inhibited.

The area containing the job data is to begin at byte 8 in data module DM21.

The function is to be started with the rising edge on input **I21.0**.

Module **-B21WZDAT** could be called as follows:

```
AN      B       I21.0
A       B       M321.0
R       B       M321.0              ; resetting the edge-trigger marker
CM              -DM21WzgDb          ; call data module DM21
A       B       M321.7              ; evaluate answer ?
JPC             -checkRsp
A       B       I21.0               ; scan input 21.0
AN      B       M321.0              ; formation of positive edge
JPI             -end
S       B       M321.0              ; setting the edge-trigger marker
S       B       M321.7              ; setting the evaluate answer marker
S       B       D8.0                ; setting start function
                                    ; definition of parameters to be transferred
L       BY      K1,A                ; function number for reading a data record
SLL  BY         A,2                 ; shifting the function no. to the respective
                                    ; location in the job byte
O       BY      K1,A                ; start function
T       BY      A,D8
L       DW      K292,A              ; total length of data area = 292 bytes
T       DW      A,D12               ; (36 bytes job data + 256 bytes data record)
L       W       K36,A               ; offset of 36 bytes
T       W       A,D16,
L       W       K-1,A               ; no channel-specific function
T       W       A,D18
L       DW      K1,A                ; table no. 1
T       DW      A,D20
L       DW      K2,A                ; sector 2
T       DW      A,D24
L       DW      K5,A                ; location 5
T       DW      A,D28
L       DW      K-1,A               ; no destination table number
T       DW      A,D32
L       DW      K-1,A               ; no destination sector number
T       DW      A,D36
L       DW      K-1,A               ; no destination location number
T       DW      A,D40
CM              -B21WZDAT,2         ; call module -B21WZDAT
P0              K8                  ; the data area begins at data byte 8
P1              K21                 ; in data module DM21
JP              -end
                -checkRsp
L       DW      D8,A                ; job, status and error codes in reg. A
A       B       A.12                ; wait while function is still active
JPC             -end
A       B       M321.7
R       B       M321.7              ; resetting of evaluate answer
A       B       A.15                ; error flag set
JPI             -FktOk
                                    ; error handling
```

```
JP          -end
            -FktOk
                             ; processing of received data
            -end
EM
```

12.13.3 Status messages

If an error or status occurs after calling the **edit tool data** function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$: Parameter error caused by incorrect parameter input in the data area.

FFFE$_H$: Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.

FFFD$_H$: Data overflow, the reserved data area is used to capacity, not enough space to store all the data.

FFFC$_H$: Module **B01APSMN** has not been called.

FFFB$_H$: Incorrect offset: The offset is too large (> reserved area) or too small (overwriting of job data occurs).

8000$_H$: NC-internal error, cannot be influenced by user.

7FFF$_H$: PxRos error, cannot be influenced by user.

7FFE$_H$: NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

68 : The specified tool table is empty.

72 : The data record specified in the job data could not be found.

108 : The previous data record was inhibited and has not yet been reenabled.

115 : The data record was not inhibited before it was saved in the data base or reenabled.

1001: Invalid table no.

The following error codes can appear only with functions **8** or **9**:

1010: The transferred file name is invalid.

1011: The file could not be opened.

1012: The file could not be closed.

1013: Error when reading the file.

1014: Error when writing the file.

1016: The file data are inconsistent.

12.14 External tool compensation (-B22WZKOR) program module

Function

The **external tool compensation** program module allows input of compensation values of compensation groups 1 and 2 (1st and 2nd external tool compensation).

1st compensation group:

The PLC program transfers the values for one of a maximum of 8 paired compensation values of ”radius and length compensation” on an NC channel.

2nd compensation group (as of software version V4.4.1):

The PLC program transfers the values for up to 8 possible compensation value sets of ”radius compensation, up to 3 length compensation values and tool edge position” on an NC channel.

☞ **As of software version V4.4.1, several compensation inputs of the 1st or 2nd compensation group can be combined in one module call.**

The units of these values are to be defined in MACODA parameter 9020 00010 and must be scaled to 0.0001 for transfer.

Example: MACODA parameter 9020 00010 is set to 0 (unit = mm). Entries of compensation values must be scaled to 0.1μ.

The external tool compensations of the 1st and 2nd compensation groups are activated and deactivated with the following G functions:

Compensation group	Transfer parameter	Activation with	Deactivation with

1	D = radius compensation
	H = length compensation
		G145,
		G245..G845
			G146

2	R = radius compensation $L_{(2)1}$, $L_{(2)2}$, $L_{(2)3}$ = length compensation or shift parameters SL = tool edge position	G147, G247..G847	G148

Compensation values become active only in the next program block to be prepared.
For compensation values to become active immediately upon their input, NC block preparation must be suspended (e.g., by **WAIT** or **block transfer inhibit**) or NC blocks already prepared must be prepared again (by **cancel distance to go**).

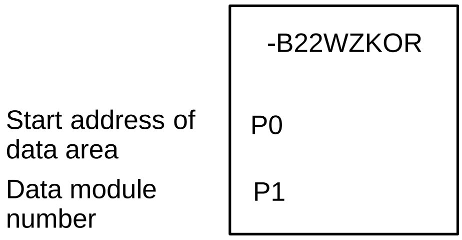
12.14.1 Module call

The module can be called unconditionally or conditionally.

If the module is called repeatedly, please note that a different data area must be parameterized for each call.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B22WZKOR**, the data area, the address of which is defined in parameter P0, has the following structure as of V4.4.1:

MSB LSB

Bit 31 . . . . 16 15 0

Error code (2 bytes)	Status (1 byte)	Job (1 byte)
Length of the total data area reserved (4 bytes)		
Channel number	Offset	
Type of compensation	Number of compensations	
1st compensation value (scaled to 0.0001)		
..		
..		
nth compensation value (scaled to 0.0001)		

The **job byte** is structured as follows:

Bit	**Function**
0	Start function
1-2	Tool compensation group Permissible values: 0 Input as previously (up to V4.4.0) of one radius compensation value and one length compensation value in DW3 and DW4. 1 Input in 1st compensation group; several compensations can be input with one call. 2 Input in 2nd compensation group; several compensations can be input with one call.
3	reserved
4-7	Compensation index Permissible values: 1–8

The function is started when bit 0 is set to logic 1 in the job byte. Compensation values are transferred to the NC immediately. The **function active** bit remains set until the NC has acknowledged the compensation input.

Because module **-B22WZKOR** does not return any data, no **offset** needs to be defined in double word DW2.

The NC channel on which the compensation values are to become active is specified by entering the respective **channel number** in double word DW2.

The structure in which compensation values are transferred depends on which compensation group is specified in the job byte:

Compensation group 0:

For existing PLC programs, the data area is transferred in its former structure (up to V4.1.0) by specifying compensation group 0. Thus, existing PLC programs remain operable without any changes required.

In this case, the compensation values are entered in double words **DW3** and **DW4**.
As was the case in the past, inputs can be made in compensation group 1 only.

MSB LSB

Bit 31 . . . . 16 15 0

<table>
<tr><td>Error code (2 bytes)</td><td>Status (1 byte)</td><td>Job (1 byte)</td></tr>
<tr><td colspan="3">Length of the total data area reserved (4 bytes)</td></tr>
<tr><td>Channel number</td><td colspan="2">Offset</td></tr>
<tr><td colspan="3">Radius compensation value (scaled to 0.0001)</td></tr>
<tr><td colspan="3">Length compensation value (scaled to 0.0001)</td></tr>
</table>

Compensation group 1 or 2:

In this case, the **number of compensations** is entered in the low-order word and the **type of compensation** in the high-order word of DW3.

Bit	Function
0-15	Number of compensations
16-31	Type of compensation Permissible values: 1 Compensation with 1 length compensation value 2 Compensation with 1 radius and 1 length compensation value 3 Compensation with 1 radius and 2 length compensation values and with tool edge position 4 Compensation with 1 radius and 3 length compensation values and with tool edge position

☞ **In compensation group 1, only compensation type 2 (1 radius and 1 length compensation value) may currently be applied.**

Compensation values, which must be entered with 0.0001 scaling, follow from DW4 onwards.

The number of compensations refers to the type of compensation and varies between 1, 2, 4 or 5 values.

Example: For 1 compensation of compensation type 4, a total of 5 compensation values must be entered, which include “1 radius and 3 length compensation values and the tool edge position”.

For compensations involving several length compensation values, inputs of the individual **length compensation** values must be made in the following **order**:

1. **L3**: Length compensation along the axis that is specified as the L3 length compensation axis in MACODA parameter 7050 00420, assignment compensation – logical axis.
2. **L1**: Length compensation along the axis that is specified as the L1 length compensation axis in MACODA parameter 7050 00420, assignment compensation – logical axis.
3. **L2**: Length compensation along the axis that is specified as the L2 length compensation axis in MACODA parameter 7050 00420, assignment compensation – logical axis.

12.14.2 Configuration example

A radius compensation of 0.1234 mm and a length compensation of 100 mm is to be input for the 2nd compensation (**G245**) on channel 1. The function is to be started with the rising edge on input **I22.0**.

The data area with the job parameters begins at byte 12 in data module DM22.

Module **-B22WZKOR** could be called as follows:

```
AN   B    I22.0
A    B    M322.0
R    B    M322.0         ; resetting the edge-trigger marker
CM        -DM22WZKOR     ; call data module DM22
A    B    M322.7         ; evaluate answer ?
JPC       -checkRsp
A    B    I22.0          ; scan input 22.0,
AN   B    M322.0         ; formation of positive edge
EMI
S    B    M322.0         ; setting the edge-trigger marker
S    B    M322.7         ; setting the evaluate answer marker
L    BY   K2,A           ; 2nd compensation (G245)
SLL  BY   A,4            ; compensation index begins at bit 4 in job
O    BY   K1,A           ; setting of start function
T    BY   A,D12          ; save job byte
L    DW   K20,A          ; definition of parameters to be transferred
T    DW   A,D16          ; total length of data area = 20 bytes,
L    W    K0,A           ; no offset required as no data
T    W    A,D20          ; is returned,
L    W    K1,A           ; compensation values on channel 1
T    W    A,D22
L    DW   K123456,A      ; radius compensation value = 12.3456 mm
T    DW   A,D24
L    DW   K1000000,A     ; length compensation value = 100 mm
T    DW   A,D28
CM        -B22WZKOR,2    ; call module -B22WZKOR
P0        K12            ; the data area begins at data byte 12
P1        K22            ; in data module DM22
EM
-checkRsp
L    DW   D12,A          ; job, status and error codes in reg. A
A    B    A.12           ; wait while function is still active
EMC
A    B    M322.7
R    B    M322.7         ; resetting of evaluate answer
A    B    A.15           ; error flag set
EMI
                         ; error handling
EM
```

12.14.3 Status messages

If an error or status occurs after calling the **external tool compensation** function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$:	Parameter error caused by incorrect parameter input in the data area.
FFFE$_H$:	Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.
FFFC$_H$:	Module **B01APSMN** has not been called.
8000$_H$:	NC-internal error, cannot be influenced by user.
7FFF$_H$:	PxRos error, cannot be influenced by user.
7FFE$_H$:	NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

1$_H$:	Invalid channel no.
13$_H$:	Invalid compensation index.
14$_H$:	Max. number of compensations exceeded (may be displayed with the compensation index).
15$_H$:	Invalid type of compensation.
16$_H$:	Invalid tool edge position.
17$_H$:	Invalid compensation group.

12.15 External zero shift (-B23XTNPV) program module

Function

The **external zero shift** program module allows inputs of axis zero shift values on an NC channel. The PLC program transfers these values for the specified number of axes and axis zero shift groups. Up to 3 axis zero shifts can be specified for each axis; they are assigned according to the sequence of axes, beginning with the 1st physical axis. The number of axis zero shift values as specified for each axis must be entered.

Use G functions **G160**, **G260** or **G360** to select one of the three axis zero shifts that can be made for each axis. To deactivate the external axis zero shift function, use **G167**.

A newly input axis zero shift value becomes active only with the **next program block** to be processed **in block preparation**. For axis zero shift values to become active immediately upon their input, block preparation must be suspended (e.g., by **WAIT** or **block transfer inhibit**) or program blocks already prepared must be prepared again (by **cancel distance to go**).

As of version V5.2.2, you may assign external axis zero shift values to axes also under their **physical axis numbers** (as opposed to their logical axis numbers). The selection as to whether the logical or the physical axis number is to be transferred is made by setting the appropriate function code in the job byte. If **physical axis numbers** are used, transfer of values is axis-related, with the axis index being followed by up to 3 axis zero shift values.

CAUTION
Axis zero shift values of axes not included in the message are set to 0 on the channel where the values are transferred!

Axis zero shift values are transferred with 0.0001 scaling in the unit as specified in MACODA parameter 9020 00010.

Example: MACODA parameter 9020 00010 is set to 0 (unit = mm). Entries of axis zero shift values must be scaled to 0.1µ for transfer.

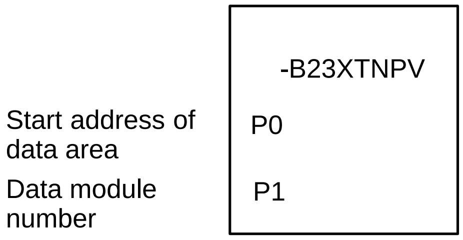
12.15.1 Module call

The module can be called unconditionally or conditionally.

If the module is called repeatedly, please note that a different data area must be parameterized for each call.

Module parameters

The module is called with the two parameters P0 and P1.



The start address (in bytes) of the data area where the function, status word and error message parameters to be transferred are to be stored must be entered in parameter P0.

If the data area is located in a data module, the data module number must be entered in parameter P1.

In cases where the data area is not defined in a data module but, e.g., in the marker field, constant K-1 must be entered in parameter P1.

In the case of program module **-B23XTNPV** with **logical axis numbers** used, the data area, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 . . . . 16 15 0

Error code (2 bytes)	Status (1 byte)	Job (1 byte)
Length of the total data area reserved (4 bytes)		
Channel number	Offset	
Number of axes		
Number of axis zero shift (ZS) values for each axis		
1st axis ZS value of the 1st axis (scaled to 0.0001)		
:		
1st axis ZS value of the nth axis (scaled to 0.0001)		
2nd axis ZS value of the 1st axis (scaled to 0.0001)		
:		
2nd axis ZS value of the nth axis (scaled to 0.0001)		
3rd axis ZS value of the 1st axis (scaled to 0.0001)		
:		
3rd axis ZS value of the nth axis (scaled to 0.0001)		

In the case of program module **-B23XTNPV** with **physical axis numbers** used, the data area, the address of which is defined in parameter P0, has the following structure:

MSB LSB

Bit 31 . . . . 16 15 0

Error code (2 bytes)	Status (1 byte)	Job (1 byte)
Length of the total data area reserved (4 bytes)		
Channel number	Offset to the 1st address of data returned.	
Number of axes		
Number of axis zero shift (ZS) values for each axis		
physical axis number		
1st axis ZS value (scaled to 0.0001)		
2nd axis ZS value (scaled to 0.0001)		
3rd axis ZS value (scaled to 0.0001)		
physical axis number		
1st axis ZS value ...		
:		
physical axis number		
1st axis ZS value (scaled to 0.0001)		
2nd axis ZS value (scaled to 0.0001)		
3rd axis ZS value (scaled to 0.0001)		

The job byte is structured as follows:

Bit	**Function**
0	Start function
1-7	reserved
4-7	Function number: 0: Axis ZS value input with logical axis numbers (for structure of data area, see above) 1: Axis ZS value input with physical axis numbers (for structure of data area, see above)

This function is activated when bit 0, **start function**, is set in the job byte and remains set for at least one PLC cycle.

Bit 0 may be reset via the PLC sequential program with the **function active** message. Otherwise, the **start function** signal is reset automatically when the function has been carried out.

Bit 4, **function active**, remains set in the status until the function has been carried out, at least, however, for the duration of one PLC cycle. When the data has been returned, the function is completed and bit 4, **function active**, is reset in the status.

Offset

Because the **-B23XTNPV** module does not return any data, no offset needs to be defined in double word DW2.

Channel number

The NC channel on which the axis zero shift values are to become active is specified by entering the respective channel number in double word DW2.

Number of axes

Contains the number of axes for which the external axis zero shifts are transferred.

Number of axis ZS values for each axis

Number of axis zero shift values entered for each axis.

Permissible values:

1: Values for the 1st axis zero shift (G160)
2: Values for the 1st and 2nd axis zero shift (G160,G260)
3: Values for the 1st, 2nd and 3rd axis zero shift (G160,G260,G360)

Axis zero shift values (logical axis number)

Contains the value by which the axis zero point of a single **logical axis** (with logical axis number) is to be shifted. The value must be entered with 0.1µ scaling.

Physical axis number

Index of the physical axis.

1st ... 3rd axis zero shift value (physical axis number)

Up to 3 axis zero shift values by which the axis zero point of a single **physical axis** (with physical axis number) is to be shifted. The values must be entered with 0.1µ scaling.

12.15.2 Configuration example

The following axis zero shifts are to be carried out on channel 1:

with **G160**:
1st axis, axis ZS by 10 mm,
2nd axis, axis ZS by –30 mm, and
3rd axis, axis ZS by 50 mm,

and with **G260**:
1st axis, axis ZS by 20 mm,
2nd axis, axis ZS by –60 mm, and
3rd axis, axis ZS by 100 mm.

The area containing the job parameters begins at byte 0 in data module DM23. The function is to be started with the rising edge on input I23.0.

Module **-B23XTNPV.PAH** could be called as follows:

```
AN     B      I23.0
A      B      M323.0
R      B      M323.0              ; resetting the edge-trigger marker
CM            -DM23xtNpv          ; call data module DM23
A      B      M323.7              ; evaluate answer ?
JPC           -checkRsp
A      B      I23.0               ; scan input 23.0,
AN     B      M323.0              ; formation of positive edge
EMI
S      B      M323.0              ; setting the edge-trigger marker
S      B      M323.7              ; setting the evaluate answer marker
L      BY     K1,A                ; data read-in
T      BY     A,D0                ; saving the job byte
L      DW     K44,A               ; total length of data area = 28 bytes
T      DW     A,D4
L      W      K0,A                ; offset irrelevant
T      W      A,D8
L      W      K1,A                ; channel no. 1
T      W      A,D10
L      DW     K3,A                ; axis ZS values for 1st, 2nd and 3rd axis
T      DW     A,D12
L      DW     K2,A                ; 2nd axis ZS values for each axis (G160,G260)
T      DW     A,D16
L      DW     K100000,A           ; with G160, 10mm for 1st axis
T      DW     A,D20
L      DW     K-300000,A          ; with G160, -30mm 2nd axis
T      DW     A,D24
L      DW     K500000,A           ; with G160, 50mm for 3rd axis
T      DW     A,D28
L      DW     K200000,A           ; with G260, 20mm for 1st axis
T      DW     A,D32
L      DW     K-600000,A          ; with G260, -60mm for 2nd axis
T      DW     A,D36
L      DW     K1000000,A          ; with G260, 100mm for 3rd axis
T      DW     A,D40
CM            -B23XTNPV,2         ; call module B23XTNPV.PAH
P0     K0                         ; the data area begins at data byte 0
P1     K23                        ; in data module DM23
       -checksp
L      DW     D0,A                ; job, status and error codes in reg. A
A      B      A.12                ; wait while function is still active
EMC
A      B      M323.7
R      B      M323.7              ; resetting of evaluate answer
A      B      A.15                ; error flag set
JPC           -error
                                  ; function completed correctly
EM
       -error
                                  ; error handling
EM
```

12.15.3 Status messages

If an error or status occurs after calling the **external zero shift** function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors and statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

FFFF$_H$:	Parameter error caused by incorrect parameter input in the data area.
FFFE$_H$:	Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.
FFFC$_H$:	Module **B01APSMN** has not been called.
8000$_H$:	NC-internal error, cannot be influenced by user.
7FFF$_H$:	PxRos error, cannot be influenced by user.
7FFE$_H$:	NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

1$_H$:	Invalid channel no.
10$_H$:	Number of axes is greater than the number of configured axes
11$_H$:	Number of axis ZS shifts is greater than the maximum permissible number

12.16 Communication with FIFO modules (-B55FIFOC) program module

Function

Communication with the CC10.3 module is provided via an 8-byte area in the form of a FIFO in the additional field. Therefore, also large data volumes can be transferred via this interface. The B55FIFOC module is designed to facilitate this communication for the PLC user.

From version 4.2.6 onwards, also the SPS_LIB.PAH module must be integrated if the B55FIFOC module is called from the PLC. The former module contains access functions to additional inputs and outputs used by the B55FIFOC module.

☞ **When the power is switched on or a program run has been stopped with the programming unit, the CC10.3 module is restarted automatically when the PLC program is started.**

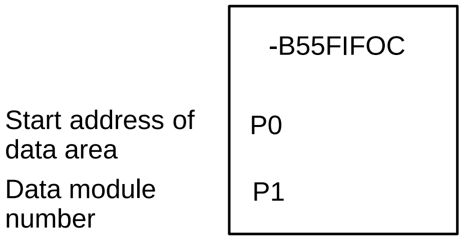
(Existing) restrictions of the CC10.3 module:

☞ **If the PLC is switched to Stop while the CC10 control is still running up to establish PLC communication (red LED goes out!) and the READY signal has not been set yet, the CC10 control can be restarted only if the 24V power supply is switched on again.**

12.16.1 Module call

Module parameters

The data area containing the job data is defined by parameters P0 and P1 when the module is called.



P0 contains the operand address and P1 the number of the data module. If the data area is located within a data module, the value transferred in P1 must be -1.

Subsequently, the module takes over communication with the CC10.3.

For program module **B55FIFOC.PAH**, the data area, whose address is defined in parameter P0, is structured as follows:

MSB LSB

Bit 31 . . . . 16 15 0

<table>
<tr><td>Error code (2 bytes)</td><td>Status (1 byte)</td><td>Job (1 byte)</td></tr>
<tr><td colspan="3">Length of the total data area reserved (4 bytes)</td></tr>
<tr><td>Channel number</td><td colspan="2">Offset</td></tr>
<tr><td>Centralized / decentralized</td><td colspan="2">CC10.3 / Rho3</td></tr>
<tr><td>Bus master FIFO no.</td><td colspan="2">Module address</td></tr>
<tr><td>Status/acknowledgement of CC10.3</td><td colspan="2">Busmaster FIFO depth</td></tr>
<tr><td>DM no., demand upstream data to CC10</td><td colspan="2">Bit width of the digital interface</td></tr>
<tr><td>Volume of demand upstream data to CC10</td><td colspan="2">Offset, demand upstream data to CC10</td></tr>
<tr><td>Offset, demand downstream data from CC10</td><td colspan="2">DM no., demand downstream data from CC10</td></tr>
<tr><td>Unit of measurement (return value)</td><td colspan="2">Volume of demand downstream data from CC10</td></tr>
<tr><td colspan="3">Digital interface (CC10.3 inputs)</td></tr>
<tr><td colspan="3">Digital interface (CC10.3 outputs)</td></tr>
</table>

Job/Status

Unlike the other APS modules, this module is active only while it is being called. Therefore, there is no need to set the ’**Start function**’ bit, and the ’**Job active**’ status bit is reset upon completion of the call. The error flag and the error code, if any, can then be evaluated immediately after calling the module.

Error code

If an error occurs, the module transfers the error code indicating the cause of the error (see below).

Size of data area

To prevent data from being overwritten accidentally, the user sets the maximum number of bytes here, earmarked for this function.

Offset

Offset defines the distance in bytes of the digital interface from the beginning of the data area used for the job transfer (job byte).

CC10.3/Rho3

Identifies whether a CC10.3 or a rho3 control unit is involved.

The following values are permitted:

0:	rho3
2:	CC10.3

Centralized/decentralized

Identifies whether the module is in centralized or decentralized operation.

☞ **At present, only centralized operation is permitted.**

Module address

The address as set on the module.

Busmaster FIFO no.

Centralized operation only.

Busmaster FIFO depth

Decentralized operation only.

Status/acknowledgement

Feedback whether the module is ready for communication.

The following values are permitted:

Bit 0:	CC10.3 communicating with PLC
Bit 10:	CC10.3 ready for communication

Bit width of digital interface

Packet size of the digital interface

The packet size of the digital interface is configured on the module and available both in transmitting and receiving directions.

DM no. of demand upstream data to CC10

Data module no. of demand data.

Offset, demand upstream data to CC10

Offset in bytes within the data module where the demand data is saved.

Volume of demand upstream data to CC10

Volume of demand data (in bytes).

If no demand data is to be transmitted, 0 must be entered here for the length.

DM no. of demand downstream data from the CC10

Data module no. of demand data.

Offset, demand downstream data from the CC10

Offset in bytes within the data module where the demand downstream data from the CC10 is saved.

Volume of demand downstream data from the CC10

Here, the module returns the volume of the demand downstream data from the CC10.3.

Digital interface (CC10.3 inputs)

Beginning of the digital CC10.3 input data.

Digital interface (CC10.3 outputs)

Beginning of the digital CC10.3 output data.

☞ **The latter must follow the input data contiguously!**

12.16.2 Status messages

If an error or status occurs after calling the **communication with FIFO modules** function which prevents proper execution of this function, **bit 7** is set in **status**.

Also, the error code represented in binary code indicates the kind of error or status that has occurred.

There are two categories of errors or statuses:

General errors and statuses occurring upon a function call that are caused by incorrect parameter settings or an internal NC error:

$\mathbf{FFFF_H}$: Parameter error caused by incorrect parameter input in the data area.

$\mathbf{FFFE_H}$: Operand overflow, the operand data exceeds the limit of the reserved data area, e.g. the highest marker to be addressed.

$\mathbf{FFFC_H}$: Module **B01APSMN** has not been called.

$\mathbf{FFFD_H}$: No memory for internal management.

$\mathbf{8000_H}$: NC-internal error, cannot be influenced by user. (Module NCS_LIB.PAH defective or does not match the module software version where the error occurred.)

$\mathbf{7FFF_H}$: PxRos error, cannot be influenced by user.

$\mathbf{7FFE_H}$: NCS error, cannot be influenced by user.

Function-specific errors and statuses occurring with the acknowledgement that the function has been executed:

1 : The CC10.3 control unit may be called only while in centralized operation.

2: The interface bit width transferred is larger than the one returned from the CC10.3.

3: The data module specified for the demand upstream data to the CC10.3 does not exist.

4: The data module for the demand upstream data to the CC10.3 contains less data than were to be transferred. The volume of the data transferred was reduced to the capacity of the data module.

5: The offset stated within the data module for demand upstream data to the CC10.3 is larger than the length of the data module.

6: The specified volume of the demand upstream data to the CC10.3 is bigger than the maximum permitted value of 128 bytes. The data volume transferred was limited to 128 bytes.

7 : The specified data module for the demand downstream data from the CC10.3 does not exist.

8: The specified data module for the demand downstream data from the CC10.3 is smaller than the data volume to be transferred from the CC10.3. The data volume was limited to the size of the data module.

9: The specified offset within the data module for the demand downstream data from the CC10.3 is bigger than the size of the data module.

☞ **For a description of the handshake upon demand data transfer/reading, please see the CC10.3 documentation.**

12.17 Conversion Integer –> ASCII (-B97ITOA) program module

Function

The **conversion Integer –> ASCII** module converts an integer into a character string ending on the string delimiter ASCII NUL (00hex).

12.17.1 Module call

Module parameters

The data area containing the job data is defined by parameters P0 and P1 when the module is called.

	-B97ITOA
4-byte integer value	P0
Offset	P1
Data module number	P2

P1 contains the operand address or, resp., the offset if the address is within the data module; P2 contains the number of the data module. If the address is not in a data module, the value to be transferred in P2 must be -1.

☞ **For the conversion to function properly, please make sure that there is enough space for all the characters (max. 11 characters).**

Example 1:

Marker 20 contains the integer to be converted into a character string.

This character string is to be saved from marker 200 onwards:

```
CM     -B97ITOA,3                  ;
P0     M20                         ; integer value
P1     M200                        ; destination address of character string
P2     K-1                         ; not in DM
```

Example 2:

Marker 20 contains the integer to be converted into a character string.

This character string is to be saved from data word 40 onwards in data module 2:

```
CM     -B97ITOA,3                  ;
P0     M20                         ; integer value
P1     K40                         ; destination address of character string from
                                   ; D40
P2     K2                          ; in DM2 onwards
```

12.18 Read system date/time (-B98DATTM) program module

Function

The **read system date/time** module returns the time, the date and the day of the week.

All this data is returned in the form of byte values.

12.18.1 Module call

Module parameters

The data area containing the job data is defined by parameters P0 through P6 when the module is called.

Meaning of the parameter values:

	-B98DATTM
Seconds	P0
Minutes	P1
Hours	P2
Day	P3
Month	P4
Year	P5
Day of the week	P6

Parameter ranges:

Year	0 – 99
Days of the week	0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday

13 Machine operating panel

☞ **For machine operating panels using KNS, see ”PLC development environment” manual.**

13.1 Machine operating panel with PROFIBUS-DP

The machine operating panel with PROFIBUS–DP can be used for the PNC-P and for the PNC-R.

The logic statuses of all keys, switches and potentiometers as well the data of a connected handwheel are directly mapped in the I/O area of the PROFIBUS–DP which is configurable via WinDP.

CAUTION
Errors and warnings relating to the operating panel bus, if, e.g. a machine operating panel connection is interrupted, will not automatically result in opening the Ready2 contact.

It therefore has to be ensured in the PLC sequential program that functions activated via the machine operating panel do not lead to hazards to persons or machines in case of an error.

Since all input data of the machine operating panels are set to zero in case of error, safety is ensured if either normally closed contacts (n.c.c.) or normally open contacts (n.o.c.) are used depending on the function type.

13.1.1 Signal transmission

The data are located on the inputs and outputs configured via WinDP. The offsets start at zero.

Example:

If the operating panel is positioned on I200/O200, S1 is assigned to input I203.6.

Assignment:

S1	n.o.c.	I203.6		H1	O201.0
S2	n.o.c.	I203.7		H2	O201.1
S3	n.o.c.	I204.5		H3	O201.2
S4	n.o.c.	I204.6		H4	O201.3
S5	n.o.c.	I204.7		H5	O201.4
S6	n.o.c.	I207.5		H6	O201.5
S7	n.o.c.	I207.6		H7	O201.6
S8	n.o.c.	I207.7		H8	O201.7
S9	n.o.c.	I206.6		H9	O200.0
S10	n.o.c.	I206.7		H10	O200.1
S11	n.o.c.	I205.0		H11	O200.2
S12	n.o.c.	I205.1		H12	O200.3
S13	n.o.c.	I205.2		H13	O200.4
S14	n.o.c.	I205.3		H14	O200.5
S15	n.o.c.	I205.4		H15	O200.6
S16	n.o.c.	I205.5		H16	O200.7
S17	n.o.c.	I205.6		H17	O203.0
S18	n.o.c.	I205.7		H18	O203.1
S19	n.o.c.	I206.4		H19	O203.2
S20	n.o.c.	I206.5		H20	O203.3
S21	n.o.c.	I201.0	n.c.c. I200.0	H21	O203.4
S22	n.o.c.	I201.1	n.c.c. I200.1	H22	O203.5
S23	n.o.c.	I201.2	n.c.c. I200.2	H23	O203.6
S24	n.o.c.	I201.3	n.c.c. I200.3	H24	O203.7
S25	n.o.c.	I201.4	n.c.c. I200.4	H25	O202.0
S26	n.o.c.	I201.5	n.c.c. I200.5	H26	O202.1
S27	n.o.c.	I201.6	n.c.c. I200.6	H27	O202.2
S28	n.o.c.	I201.7	n.c.c. I200.7	H28	O202.3
S29	n.o.c.	I202.0	n.c.c. I203.0	H29	O202.4
S30	n.o.c.	I202.1	n.c.c. I203.1	H30	O202.5
S31	n.o.c.	I203.2	n.c.c. I202.2	H31	O202.6
S32	n.o.c.	I203.3	n.c.c. I202.3	H32	O202.7
S33	n.o.c.	I203.4	n.c.c. I202.4		
S34	n.o.c.	I203.5	n.c.c. I202.5		
SS1	Rotary switch	I206.0 ... I206.3			
SS2	Rotary switch	I204.0 ... I204.4			
SS3	Rotary switch	I207.0 ... I207.4			
ST5	Pin2 .. Pin6	I208.7 ... I208.3			
ST3	Pin2	I208.0			
	Pin3	I208.2			
	Pin6	I209.1			
	Pin7	I209.3			
	Pin8	I209.5			
	Pin9	I209.7			
	Pin15	I208.1			
	Pin18	I209.0			
	Pin19	I209.2			
	Pin20	I209.4			
	Pin21	I209.6			

Machine operating panel with PROFIBUS-DP

The addresses of the I/Os are relative to the start of the data area

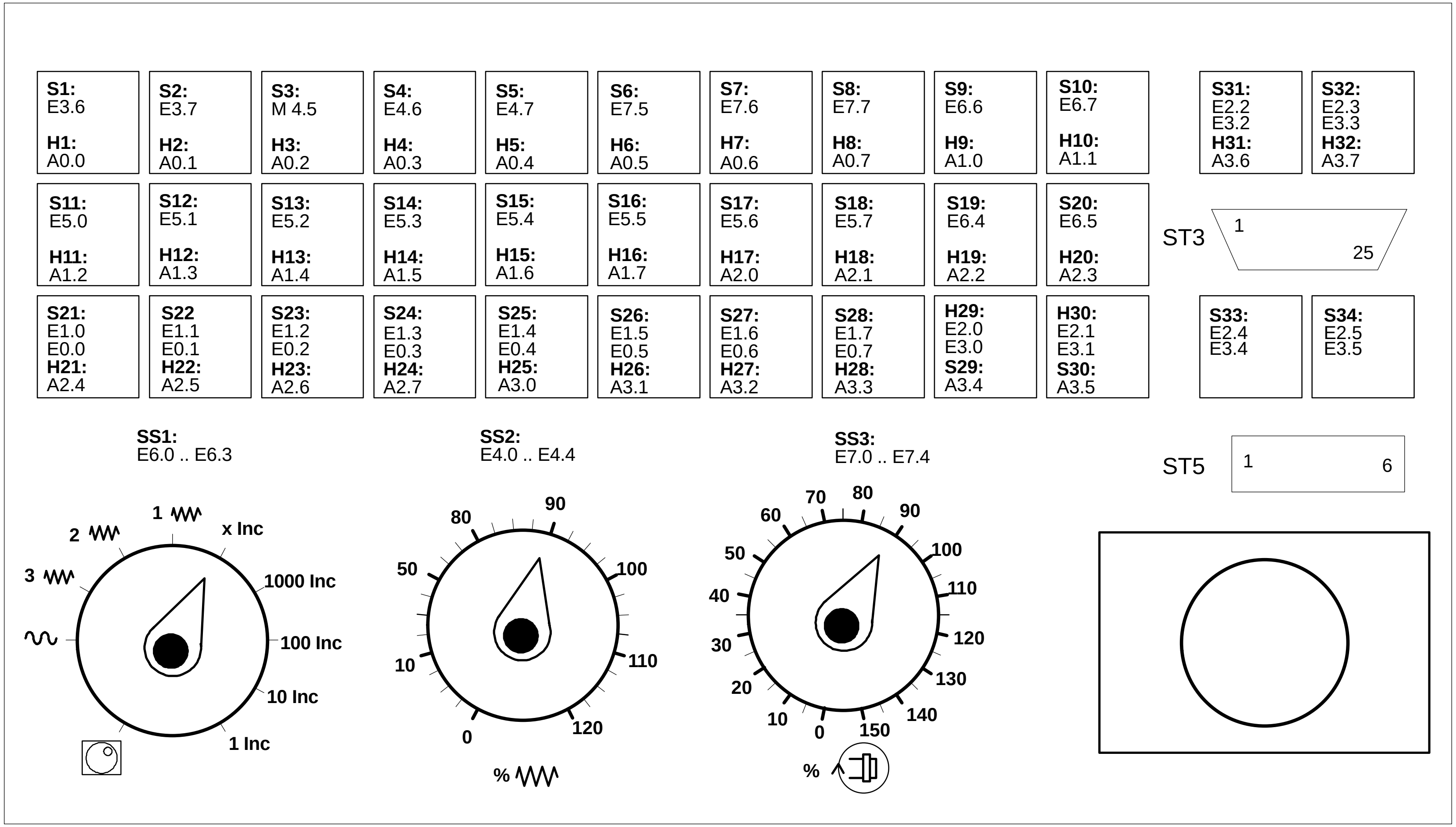
Assignment ST3:

Pin 2	E8.0
Pin 3	E8.2
Pin 6	E9.1
Pin 7	E9.3
Pin 8	E9.5
Pin 9	E9.7
Pin15	E8.1
Pin18	E9.0
Pin19	E9.2
Pin20	E9.4
Pin21	E9.6

E = input
A = output

Assignment ST5:

Pin 2	E8.7
Pin 3	E8.6
Pin 4	E8.5
Pin 5	E8.4
Pin 6	E8.3



S1: E3.6 **H1:** A0.0	**S2:** E3.7 **H2:** A0.1	**S3:** M 4.5 **H3:** A0.2	**S4:** E4.6 **H4:** A0.3	**S5:** E4.7 **H5:** A0.4	**S6:** E7.5 **H6:** A0.5	**S7:** E7.6 **H7:** A0.6	**S8:** E7.7 **H8:** A0.7	**S9:** E6.6 **H9:** A1.0	**S10:** E6.7 **H10:** A1.1	**S31:** E2.2 E3.2 **H31:** A3.6	**S32:** E2.3 E3.3 **H32:** A3.7
S11: E5.0 **H11:** A1.2	**S12:** E5.1 **H12:** A1.3	**S13:** E5.2 **H13:** A1.4	**S14:** E5.3 **H14:** A1.5	**S15:** E5.4 **H15:** A1.6	**S16:** E5.5 **H16:** A1.7	**S17:** E5.6 **H17:** A2.0	**S18:** E5.7 **H18:** A2.1	**S19:** E6.4 **H19:** A2.2	**S20:** E6.5 **H20:** A2.3		
S21: E1.0 E0.0 **H21:** A2.4	**S22** E1.1 E0.1 **H22:** A2.5	**S23:** E1.2 E0.2 **H23:** A2.6	**S24:** E1.3 E0.3 **H24:** A2.7	**S25:** E1.4 E0.4 **H25:** A3.0	**S26:** E1.5 E0.5 **H26:** A3.1	**S27:** E1.6 E0.6 **H27:** A3.2	**S28:** E1.7 E0.7 **H28:** A3.3	**H29:** E2.0 E3.0 **S29:** A3.4	**H30:** E2.1 E3.1 **S30:** A3.5	**S33:** E2.4 E3.4	**S34:** E2.5 E3.5

SS1:
E6.0 .. E6.3

SS2:
E4.0 .. E4.4

SS3:
E7.0 .. E7.4

Connecting the handwheel

The handwheel is connected via D-sub plug connector ST3 on the **machine operating panel (MOP).**

Apart from the digital signals of the incremental measurement encoder, switching signals (inputs) can also be transferred to the PLC.

Switching signals are transmitted through plug connector ST3:

Pin 2	I 8.0
Pin 3	I 8.2
Pin 6	I 9.1
Pin 7	I 9.3
Pin 8	I 9.5
Pin 9	I 9.7
Pin15	I 8.1
Pin18	I 9.0
Pin19	I 9.2
Pin20	I 9.4
Pin21	I 9.6

13.2 Machine operating panel with CAN bus

The machine operating panel with CAN bus can only be used for the PNC-R.

The logic statuses of all keys, switches and potentiometers as well the data of a connected handwheel are transferred via CAN bus. These signals are mapped on the iPCL/ICL700 marker range and can thus be linked by the PLC sequential program.

The necessary machine operating panels are defined by appropriate entries in the OM2 organization module (default assignment).
Also, in the marker range of organization module OM2, a start address for signal mapping can be defined for every machine operating panel.

CAUTION
Errors and warnings relating to the operating panel bus, if, e.g. a machine operating panel connection is interrupted, will not automatically result in opening the Ready2 contact.

It therefore has to be ensured in the PLC sequential program that functions activated via the machine operating panel do not lead to hazards to persons or machines in case of an error.

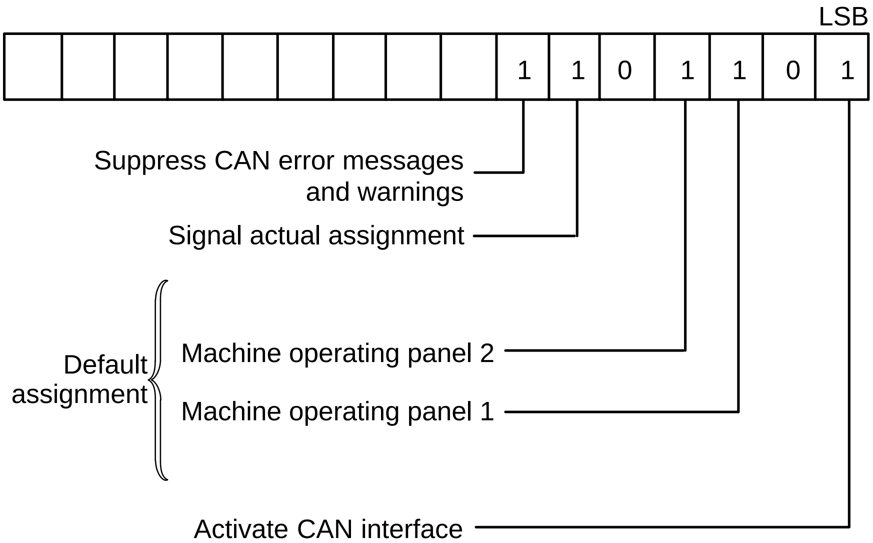
Since all input data of the machine operating panels are set to zero in case of error, safety is ensured if either normally closed contacts (n.c.c.) or normally open contacts (n.o.c.) are used depending on the function type.

By checking the actual assignment, which is also set to zero in case of error, an error can be detected by the PLC sequential program and the Ready2 contact can be opened, if required.

13.2.1 Activating the machine operating panel

The definitions in organization module OM2 are made in the following data words:

Data word 97


LSB
1
1
0
1
1
0
1
Suppress CAN error messages and warnings
Signal actual assignment
Default assignment
Machine operating panel 2
Machine operating panel 1
Activate CAN interface

The operating panel interface is activated by setting data bit 0. The default assignment is set by data bits 2 and 3 as follows:

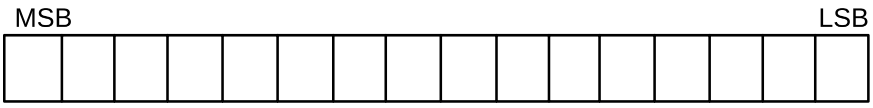
The assignment is automatically recognized when the operating panel bus is initialized. If not all available components are detected, initialization is continued with the existing components after a time-out.

To activate MOP1 and MOP2, e.g., data bits 2 and 3 must be set in addition to data bit 0.
For example, if 2 machine operating panels (MOP1 and MOP2) are connected, but data bit 3 is not set, only the input data of MOP1 will be copied into the marker range.

Data bit 5 serves to define whether or not the actual assignment is to be mapped in the marker range. This information on the components detected during initialization is made available to the PLC sequential program in the marker range only if data bit 5 has been set.

Data bit 6 serves to suppress all error messages and warnings relating to the operating panel bus. Error messages or warnings are output, e.g., if not all of the components specified in the default assignment are detected, or if the data link to one of the machine operating panels is interrupted (connector unplugged). If data bit 6 has been set, the system will not generate any error messages or warnings.

Data word 99



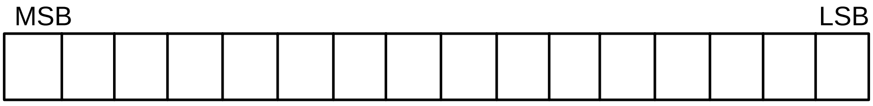
The start address of the marker area where the **MOP1** signals are to be mapped is defined in **data word 99**. The PLC sequential program has access to this map.

The signals of the **MOP1** are altogether mapped on 20 consecutive bytes. The input signals (keys, rotary switches, plug connectors) are mapped on the first 16 bytes; the output signals (display lights) are mapped on the following 4 bytes.

The start address may lie anywhere in the marker area. However, when defining the start address, please make sure that the selected marker area is not already assigned to another function.

Possible entry:	iPCL:	$K0_D$ to $K8172_D$ (marker byte 0 to marker byte 8172)
	ICL 700:	$K0_D$ to $K6124_D$ (marker byte 0 to marker byte 6124)

Data word 100



The start address of the marker area where the **MOP2** signals are to be mapped is defined in **data word 100.** The PLC sequential program has access to this map.

The signals of the **MOP2** are altogether mapped on 20 consecutive bytes. The input signals (keys, rotary switches, plug connectors) are mapped on the first 16 bytes; the output signals (display lights) are mapped on the following 4 bytes.

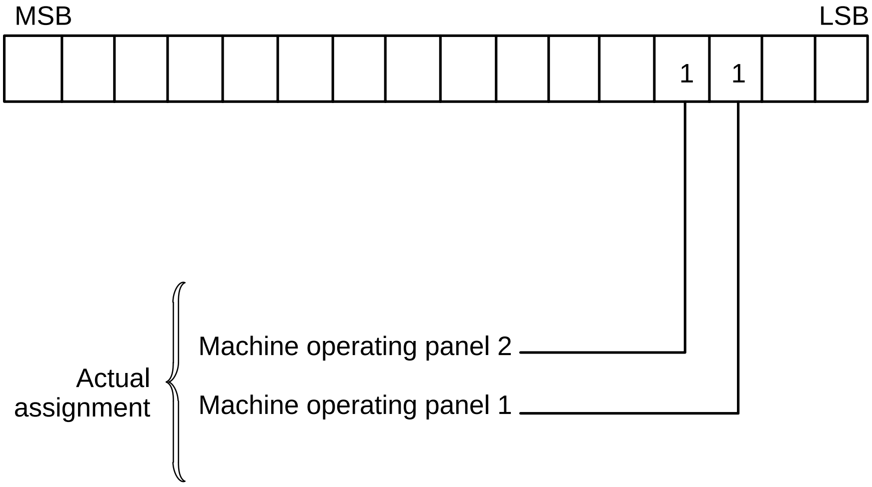
The start address may lie anywhere in the marker area. However, when defining the start address, please make sure that the selected marker area is not already assigned to another function.

Possible entry:	iPCL:	$K0_D$ to $K8172_D$ (marker byte 0 to marker byte 8172)
	ICL 700:	$K0_D$ to $K6124_D$ (marker byte 0 to marker byte 6124)

Data word 101

The start address of the marker area where the **actual assignment** is to be stored is defined in **data word 101**. The actual assignment is copied to 2 consecutive bytes and coded as follows:

Actual assignment coding:



The PLC sequential program has access to this map.

The start address may lie anywhere in the marker area. However, when defining the start address, please make sure that the selected marker area is not already assigned to another function.

Possible entry:	$K0_D$ to $K6124_D$ (corresponds to marker byte 0 to marker byte 6124)

13.2.2 Signal transmission

The switches or displays are mapped on the corresponding markers according to the definition of data word 99 (100) in the organization module OM2.

Example:

If marker M6100 is entered as the start address in data word 99 for machine operating panel MOP1, this means that the 16 bytes of switch input signals as well as the 4 bytes of display output signals are stored on marker bytes M6100 .. M6119.

Assignment:

Marker assignment for the machine operating panel:

S1	n.o.c.	M6103.6			H1	M6117.0
S2	n.o.c.	M6103.7			H2	M6117.1
S3	n.o.c.	M6104.5			H3	M6117.2
S4	n.o.c.	M6104.6			H4	M6117.3
S5	n.o.c.	M6104.7			H5	M6117.4
S6	n.o.c.	M6107.5			H6	M6117.5
S7	n.o.c.	M6107.6			H7	M6117.6
S8	n.o.c.	M6107.7			H8	M6117.7
S9	n.o.c.	M6106.6			H9	M6116.0
S10	n.o.c.	M6106.7			H10	M6116.1
S11	n.o.c.	M6105.0			H11	M6116.2
S12	n.o.c.	M6105.1			H12	M6116.3
S13	n.o.c.	M6105.2			H13	M6116.4
S14	n.o.c.	M6105.3			H14	M6116.5
S15	n.o.c.	M6105.4			H15	M6116.6
S16	n.o.c.	M6105.5			H16	M6116.7
S17	n.o.c.	M6105.6			H17	M6119.0
S18	n.o.c.	M6105.7			H18	M6119.1
S19	n.o.c.	M6106.4			H19	M6119.2
S20	n.o.c.	M6106.5			H20	M6119.3
S21	n.o.c.	M6101.0	n.c.c.	M6100.0	H21	M6119.4
S22	n.o.c.	M6101.1	n.c.c.	M6100.1	H22	M6119.5
S23	n.o.c.	M6101.2	n.c.c.	M6100.2	H23	M6119.6
S24	n.o.c.	M6101.3	n.c.c.	M6100.3	H24	M6119.7
S25	n.o.c.	M6101.4	n.c.c.	M6100.4	H25	M6118.0
S26	n.o.c.	M6101.5	n.c.c.	M6100.5	H26	M6118.1
S27	n.o.c.	M6101.6	n.c.c.	M6100.6	H27	M6118.2
S28	n.o.c.	M6101.7	n.c.c.	M6100.7	H28	M6118.3
S29	n.o.c.	M6102.0	n.c.c.	M6103.0	H29	M6118.4
S30	n.o.c.	M6102.1	n.c.c.	M6103.1	H30	M6118.5
S31	n.o.c.	M6103.2	n.c.c.	M6102.2	H31	M6118.6
S32	n.o.c.	M6103.3	n.c.c.	M6102.3	H32	M6118.7
S33	n.o.c.	M6103.4	n.c.c.	M6102.4		
S34	n.o.c.	M6103.5	n.c.c.	M6102.5		
SS1	Rotary switch	M6106.0 ... M6106.3				
SS2	Rotary switch	M6104.0 ... M6104.4				
SS3	Rotary switch	M6107.0 ... M6107.4				
ST5	Pin2 .. Pin6	M6108.7 ... M6108.3				
ST3	Pin2	M6108.0	Pin3	M6108.2	Pin6	M6109.1
	Pin7	M6109.3	Pin8	M6109.5	Pin9	M6109.7
	Pin15	M6108.1	Pin18	M6109.0	Pin19	M6109.2
	Pin20	M6109.4	Pin21	M6109.6		

Machine operating panel with CAN connection (SNCI only)

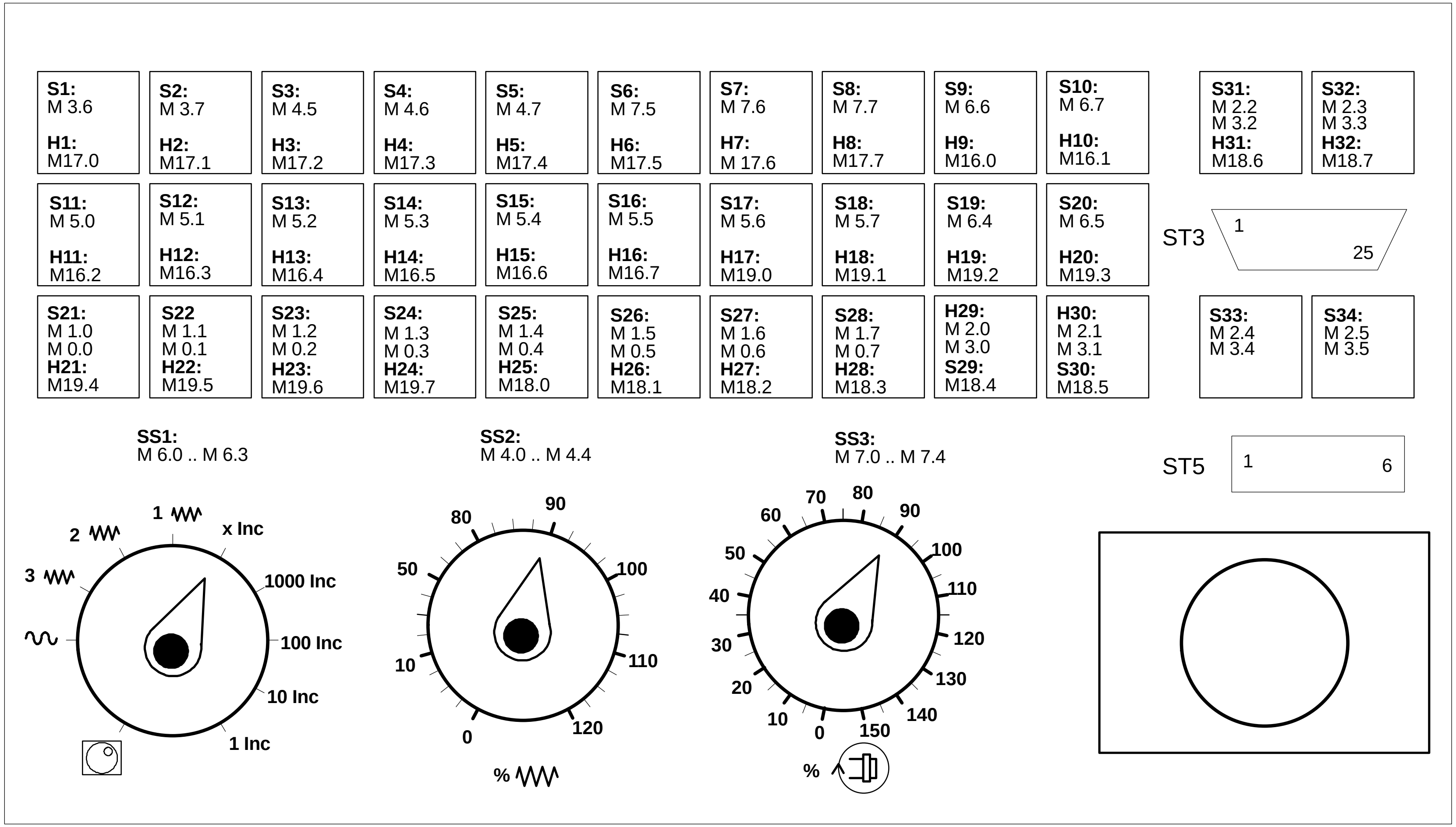
The addresses of the markers are relative to the start of the data area

Assignment ST3:

Pin 2	M 8.0
Pin 3	M 8.2
Pin 6	M 9.1
Pin 7	M 9.3
Pin 8	M 9.5
Pin 9	M 9.7
Pin15	M 8.1
Pin18	M 9.0
Pin19	M 9.2
Pin20	M 9.4
Pin21	M 9.6

Assignment ST5:

Pin 2	M 8.7
Pin 3	M 8.6
Pin 4	M 8.5
Pin 5	M 8.4
Pin 6	M 8.3



S1: M 3.6 **H1:** M17.0	**S2:** M 3.7 **H2:** M17.1	**S3:** M 4.5 **H3:** M17.2	**S4:** M 4.6 **H4:** M17.3	**S5:** M 4.7 **H5:** M17.4	**S6:** M 7.5 **H6:** M17.5	**S7:** M 7.6 **H7:** M 17.6	**S8:** M 7.7 **H8:** M17.7	**S9:** M 6.6 **H9:** M16.0	**S10:** M 6.7 **H10:** M16.1	**S31:** M 2.2 M 3.2 **H31:** M18.6	**S32:** M 2.3 M 3.3 **H32:** M18.7
S11: M 5.0 **H11:** M16.2	**S12:** M 5.1 **H12:** M16.3	**S13:** M 5.2 **H13:** M16.4	**S14:** M 5.3 **H14:** M16.5	**S15:** M 5.4 **H15:** M16.6	**S16:** M 5.5 **H16:** M16.7	**S17:** M 5.6 **H17:** M19.0	**S18:** M 5.7 **H18:** M19.1	**S19:** M 6.4 **H19:** M19.2	**S20:** M 6.5 **H20:** M19.3		
S21: M 1.0 M 0.0 **H21:** M19.4	**S22** M 1.1 M 0.1 **H22:** M19.5	**S23:** M 1.2 M 0.2 **H23:** M19.6	**S24:** M 1.3 M 0.3 **H24:** M19.7	**S25:** M 1.4 M 0.4 **H25:** M18.0	**S26:** M 1.5 M 0.5 **H26:** M18.1	**S27:** M 1.6 M 0.6 **H27:** M18.2	**S28:** M 1.7 M 0.7 **H28:** M18.3	**H29:** M 2.0 M 3.0 **S29:** M18.4	**H30:** M 2.1 M 3.1 **S30:** M18.5	**S33:** M 2.4 M 3.4	**S34:** M 2.5 M 3.5

Connecting the handwheel

The handwheel is connected via D-sub plug connector ST3 on the **machine operating panel (MOP).**

Apart from the digital signals of the incremental measurement encoder, switching signals (inputs) can also be transferred to the PLC.

Switching signals are transmitted through plug connector ST3:

Pin 2	M 8.0
Pin 3	M 8.2
Pin 6	M 9.1
Pin 7	M 9.3
Pin 8	M 9.5
Pin 9	M 9.7
Pin15	M 8.1
Pin18	M 9.0
Pin19	M 9.2
Pin20	M 9.4
Pin21	M 9.6

Notes:

A Appendix

A.1 Overview of interface signals

A.1.A General interface signals, PLC –> NC

NC-I	Signal designation	NC-I	Signal designation
0.0	System control reset	1.0	Stroke inhibit
0.1	Edit inhibit	1.1	Stroke reservation
0.2	res.	1.2	Stroke ON
0.3	res.	1.3	res.
0.4	res.	1.4	res.
0.5	res.	1.5	res.
0.6	res.	1.6	res.
0.7	res.	1.7	res.
2.0	res.	3.0	res.
2.1	res.	3.1	res.
2.2	res.	3.2	res.
2.3	res.	3.3	res.
2.4	res.	3.4	res.
2.5	res.	3.5	res.
2.6	res.	3.6	res.
2.7	res.	3.7	res.

A.1.B General interface signals, NC –> PLC

NC-O	Signal designation	NC-O	Signal designation
4.0	res.	5.0	Stroke intended
4.1	res.	5.1	Stroke is not running
4.2	res.	5.2	res.
4.3	res.	5.3	res.
4.4	res.	5.4	res.
4.5	res.	5.5	res.
4.6	res.	5.6	res.
4.7	res.	5.7	res.
6.0	res.	6.0	res.
6.1	res.	6.1	res.
6.2	res.	6.2	res.
6.3	res.	6.3	res.
6.4	res.	6.4	res.
6.5	res.	6.5	res.
6.6	res.	6.6	res.
6.7	res.	6.7	res.

A.1.C Channel-related interface signals, PLC –> NC

NC-I	Signal designation	NC-I	Signal designation
0.0	Conditional jump	1.0	Cycle start
0.1	res.	1.1	Control reset
0.2	res.	1.2	Override 100%
0.3	res.	1.3	res.
0.4	res.	1.4	Automatic restart
0.5	res.	1.5	res.
0.6	Workpiece coordinates manual +	1.6	Optional stop
0.7	Workpiece coordinates manual –	1.7	Limit rapid travel
2.0	Mode selection bit 0	3.0	Mode selection by PLC
2.1	Mode selection bit 1	3.1	Return to contour
2.2	Mode selection bit 2	3.2	G131 Tool rotation release
2.3	Mode selection bit 3	3.3	Cancel distance to go
2.4	res.	3.4	Block skip 1
2.5	res.	3.5	res.
2.6	Online correction enable	3.6	res.
2.7	Online correction direction of rotation	3.7	res.
4.0	Override bit 0 (Override value)	5.0	Override bit 8 (with direct input in 0.01%)
4.1	Override bit 1 (Override value)	5.1	Override bit 9 (with direct input in 0.01%)
4.2	Override bit 2 (Override value)	5.2	Override bit 10 (with direct input in 0.01%)
4.3	Override bit 3 (Override value)	5.3	Override bit 11 (with direct input in 0.01%)
4.4	Override bit 4 (Override value)	5.4	Override bit 12 (with direct input in 0.01%)
4.5	Override bit 5 (with direct input in 0.01%)	5.5	Override bit 13 (with direct input in 0.01%)
4.6	Override bit 6 (with direct input in 0.01%)	5.6	Override bit 14 (with direct input in 0.01%)
4.7	Override bit 7 (with direct input in 0.01%)	5.7	Override bit 15 (with direct input in 0.01%)
6.0	Customer input 1	7.0	Automatic reselection OFF
6.1	Customer input 2	7.1	Switch to next block
6.2	Customer input 3	7.2	Coordinate coupling completed
6.3	Customer input 4	7.3	res.
6.4	Customer input 5	7.4	Rapid retraction
6.5	Customer input 6	7.5	Feed inhibit
6.6	Customer input 7	7.6	Feed hold
6.7	Customer input 8	7.7	Block transfer inhibit
8.0	Feed inhibit general 1	9.0	Feed inhibit general 9
8.1	Feed inhibit general 2	9.1	Feed inhibit general 10
8.2	Feed inhibit general 3	9.2	Feed inhibit general 11
8.3	Feed inhibit general 4	9.3	Feed inhibit general 12
8.4	Feed inhibit general 5	9.4	Feed inhibit general 13
8.5	Feed inhibit general 6	9.5	Feed inhibit general 14
8.6	Feed inhibit general 7	9.6	Feed inhibit general 15
8.7	Feed inhibit general 8	9.7	Feed inhibit general 16
10.0	Feed hold 1	11.0	Feed hold 9
10.1	Feed hold 2	11.1	Feed hold 10
10.2	Feed hold 3	11.2	Feed hold 11
10.3	Feed hold 4	11.3	Feed hold 12
10.4	Feed hold 5	11.4	Feed hold 13
10.5	Feed hold 6	11.5	Feed hold 14
10.6	Feed hold 7	11.6	Feed hold 15
10.7	Feed hold 8	11.7	Feed hold 16

NC-I	Signal designation	NC-I	Signal designation
12.0	Block transfer inhibit 1	13.0	Block transfer inhibit 9
12.1	Block transfer inhibit 2	13.1	Block transfer inhibit 10
12.2	Block transfer inhibit 3	13.2	Block transfer inhibit 11
12.3	Block transfer inhibit 4	13.3	Block transfer inhibit 12
12.4	Block transfer inhibit 5	13.4	Block transfer inhibit 13
12.5	Block transfer inhibit 6	13.5	Block transfer inhibit 14
12.6	Block transfer inhibit 7	13.6	Block transfer inhibit 15
12.7	Block transfer inhibit 8	13.7	Block transfer inhibit 16

A.1.D Channel-related interface signals, NC –> PLC

NC-O	Signal designation	NC-O	Signal designation
14.0	Program running	15.0	Active mode bit 0
14.1	Feed hold active	15.1	Active mode bit 1
14.2	Control reset executed	15.2	Active mode bit 2
14.3	Block transfer inhibit active	15.3	Active mode bit 3
14.4	res.	15.4	res.
14.5	res.	15.5	res.
14.6	res.	15.6	res.
14.7	res.	15.7	res.
16.0	NC ready	17.0	Activate block skip 1
16.1	Program stop with M0, M1	17.1	res.
16.2	Program end with M2, M30	17.2	res.
16.3	Optional stop activated	17.3	Channel status bit 0
16.4	Ready for re-entry	17.4	Channel status bit 1
16.5	Re-entry active	17.5	Channel status bit 2
16.6	Remove finished	17.6	Channel status bit 3
16.7	res.	17.7	Channel status bit 4
18.0	G131 Tool rotation	19.0	External tool correction active bit 0
18.1	res.	19.1	External tool correction active bit 1
18.2	res.	19.2	External tool correction active bit 2
18.3	Override 0%	19.3	External tool correction active bit 3
18.4	Override 100%	19.4	External zero offset active bit 0
18.5	res.	19.5	External zero offset active bit 1
18.6	res.	19.6	res.
18.7	Inpos range 2 active	19.7	res.
20.0	Rapid traverse active	21.0	G189 active
20.1	G70 active	21.1	G90 active
20.2	G32 active	21.2	G91 active
20.3	G33 active	21.3	G92 active
20.4	Thread cycle active	21.4	G95 active
20.5	G41 active	21.5	Const. cutting velocity active
20.6	G42 active	21.6	Coordinate coupling completed
20.7	G63 active	21.7	res.
22.0	Customer output 1	23.0	General tool compensation active bit 0
22.1	Customer output 2	23.1	General tool compensation active bit 1
22.2	Customer output 3	23.2	General tool compensation active bit 2
22.3	Customer output 4	23.3	General tool compensation active bit 3
22.4	Customer output 5	23.4	res.
22.5	Customer output 6	23.5	res.
22.6	Customer output 7	23.6	res.
22.7	Customer output 8	23.7	res.
24.0	CPL customer output 1	25.0	CPL customer output 9
24.1	CPL customer output 2	25.1	CPL customer output 10
24.2	CPL customer output 3	25.2	CPL customer output 11
24.3	CPL customer output 4	25.3	CPL customer output 12
24.4	CPL customer output 5	25.4	CPL customer output 13
24.5	CPL customer output 6	25.5	CPL customer output 14
24.6	CPL customer output 7	25.6	CPL customer output 15
24.7	CPL customer output 8	25.7	CPL customer output 16

A.1.E Axis-related interface signals, PLC –> NC

NC-I	Signal designation	NC-I	Signal designation
0.0	Manual +	1.0	Manual feed/incremental bit 0
0.1	Manual –	1.1	Manual feed/incremental bit 1
0.2	Axis override 100%	1.2	Manual feed/incremental bit 2
0.3	Inch incr. step	1.3	Manual feed/incremental bit 3
0.4	Diameter incr. Step	1.4	Travel limit range bit 0
0.5	Handwheel direction of rotation	1.5	Travel limit range bit 1
0.6	res.	1.6	Mode selection bit 0
0.7	res.	1.7	Mode selection bit 1
2.0	Handwheel assignment bit 0	3.0	Axis is discharged
2.1	Handwheel assignment bit 1	3.1	Control reset (auxiliary axis)
2.2	res.	3.2	Suppress software limit switches
2.3	res.	3.3	Cancel fixed stop
2.4	res.	3.4	res.
2.5	Traverse to next grid position (Hirth axis)	3.5	res.
2.6	res.	3.6	res.
2.7	res.	3.7	Torque reduction
4.0	Override bit 0	5.0	res.
4.1	Override bit 1	5.1	res.
4.2	Override bit 2	5.2	res.
4.3	Override bit 3	5.3	res.
4.4	Override bit 4	5.4	res.
4.5	Coupling lag: NC error OFF	5.5	res.
4.6	Standstill torque: NC error OFF	5.6	res.
4.7	Gantry slave to master position	5.7	res.
6.0	Customer input 1	7.0	Hold command position
6.1	Customer input 2	7.1	res.
6.2	Customer input 3	7.2	res.
6.3	Customer input 4	7.3	res.
6.4	Customer input 5	7.4	res.
6.5	Customer input 6	7.5	Drive inhibit, general
6.6	Customer input 7	7.6	Drive OFF, general
6.7	Customer input 8	7.7	Feed inhibit, general
8.0	Drive inhibit 1	9.0	Drive inhibit 9
8.1	Drive inhibit 2	9.1	Drive inhibit 10
8.2	Drive inhibit 3	9.2	Drive inhibit 11
8.3	Drive inhibit 4	9.3	Drive inhibit 12
8.4	Drive inhibit 5	9.4	Drive inhibit 13
8.5	Drive inhibit 6	9.5	Drive inhibit 14
8.6	Drive inhibit 7	9.6	Drive inhibit 15
8.7	Drive inhibit 8	9.7	Drive inhibit 16
10.0	Drive off 1	11.0	Drive off 9
10.1	Drive off 2	11.1	Drive off 10
10.2	Drive off 3	11.2	Drive off 11
10.3	Drive off 4	11.3	Drive off 12
10.4	Drive off 5	11.4	Drive off 13
10.5	Drive off 6	11.5	Drive off 14
10.6	Drive off 7	11.6	Drive off 15
10.7	Drive off 8	11.7	Drive off 16

NC-I	Signal designation	NC-I	Signal designation
12.0	Feed inhibit 1	13.0	Feed inhibit 9
12.1	Feed inhibit 2	13.1	Feed inhibit 10
12.2	Feed inhibit 3	13.2	Feed inhibit 11
12.3	Feed inhibit 4	13.3	Feed inhibit 12
12.4	Feed inhibit 5	13.4	Feed inhibit 13
12.5	Feed inhibit 6	13.5	Feed inhibit 14
12.6	Feed inhibit 7	13.6	Feed inhibit 15
12.7	Feed inhibit 8	13.7	Feed inhibit 16

A.1.F Axis-related interface signals, NC –> PLC

NC-O	Signal designation	NC-O	Signal designation
14.0	Sercos system ready	15.0	Axis inhibited (test)
14.1	Drive ready	15.1	Axis override 0 %
14.2	Drive under control	15.2	Axis override 100 %
14.3	Travel command	15.3	Reset executed (auxiliary axis)
14.4	Traversing direction –	15.4	res.
14.5	Axis running	15.5	res.
14.6	Axis in position	15.6	Reference point was reached
14.7	res.	15.7	Reference point is known
16.0	Index of the master axis bit 0	17.0	Move to fixed stop active
16.1	Index of the master axis bit 1	17.1	Fixed stop reached
16.2	Index of the master axis bit 2	17.2	Axis near endpoint
16.3	Index of the master axis bit 3	17.3	Error diagnostics class 1
16.4	Index of the master axis bit 4	17.4	Class 2 diagnostics changed
16.5	Axis in grid position (Hirth axis)	17.5	Class 3 diagnostics changed
16.6	Coupling lag exceeded	17.6	Torque limit
16.7	Standstill torque exceeded	17.7	Drive-controlled interpolation
18.0	Axis position 1	19.0	Channel number bit 0
18.1	Axis position 2	19.1	Channel number bit 1
18.2	Axis position 3	19.2	Channel number bit 2
18.3	Axis position 4	19.3	Channel number bit 3
18.4	Axis position 5	19.4	res.
18.5	Axis position 6	19.5	res.
18.6	Axis position 7	19.6	res.
18.7	Axis position 8	19.7	res.
20.0	Customer output 1	21.0	SERCOS drive status bit 0
20.1	Customer output 2	21.1	SERCOS drive status bit 1
20.2	Customer output 3	21.2	SERCOS drive status bit 2
20.3	Customer output 4	21.3	SERCOS drive status bit 3
20.4	Customer output 5	21.4	SERCOS drive status bit 4
20.5	Customer output 6	21.5	SERCOS drive status bit 5
20.6	Customer output 7	21.6	SERCOS drive status bit 6
20.7	Customer output 8	21.7	SERCOS drive status bit 7

NC-O	Signal designation	NC-O	Signal designation
22.0	res.	23.0	res.
22.1	res.	23.1	res.
22.2	res.	23.2	res.
22.3	res.	23.3	res.
22.4	res.	23.4	res.
22.5	res.	23.5	res.
22.6	res.	23.6	res.
22.7	res.	23.7	res.
24.0	res.	24.0	res.
24.1	res.	25.1	res.
24.2	res.	25.2	res.
24.3	res.	25.3	res.
24.4	res.	25.4	res.
24.5	res.	25.5	res.
24.6	res.	25.6	res.
24.7	res.	25.7	res.
26.0	res.	27.0	res.
26.1	res.	27.1	res.
26.2	res.	27.2	res.
26.3	res.	27.3	res.
26.4	res.	27.4	res.
26.5	res.	27.5	res.
26.6	res.	27.6	res.
26.7	res.	27.7	res.

A.1.G Spindle-related interface signals, PLC –> NC

NC-I	Signal designation	NC-I	Signal designation
0.0	Spindle M3 manual	1.0	Spindle speed jog bit 0
0.1	Spindle M4 manual	1.1	Spindle speed jog bit 1
0.2	Spindle M5 manual	1.2	Spindle speed jog bit 2
0.3	res.	1.3	res.
0.4	Spindle jog M3	1.4	res.
0.5	Spindle jog M4	1.5	res.
0.6	res.	1.6	C axis ON
0.7	res.	1.7	C axis OFF
2.0	Override bit 0	3.0	res.
2.1	Override bit 1	3.1	res.
2.2	Override bit 2	3.2	Spindle orientation manual
2.3	Override bit 3	3.3	Spindle override 100%
2.4	Override bit 4	3.4	Reset (spindle)
2.5	res.	3.5	res.
2.6	res.	3.6	res.
2.7	res.	3.7	res.
4.0	res.	5.0	Gear range 1 acknowledge
4.1	res.	5.1	Gear range 2 acknowledge
4.2	res.	5.2	Gear range 3 acknowledge
4.3	res.	5.3	Gear range 4 acknowledge
4.4	res.	5.4	res.
4.5	res.	5.5	res.
4.6	res.	5.6	res.
4.7	res.	5.7	Gear idle motion acknowledge
6.0	Customer input 1	7.0	res.
6.1	Customer input 2	7.1	res.
6.2	Customer input 3	7.2	res.
6.3	Customer input 4	7.3	res.
6.4	Customer input 5	7.4	res.
6.5	Customer input 6	7.5	Drive inhibit, general
6.6	Customer input 7	7.6	Drive OFF, general
6.7	Customer input 8	7.7	Spindle inhibit, general
8.0	Drive inhibit 1	9.0	Drive inhibit 9
8.1	Drive inhibit 2	9.1	Drive inhibit 10
8.2	Drive inhibit 3	9.2	Drive inhibit 11
8.3	Drive inhibit 4	9.3	Drive inhibit 12
8.4	Drive inhibit 5	9.4	Drive inhibit 13
8.5	Drive inhibit 6	9.5	Drive inhibit 14
8.6	Drive inhibit 7	9.6	Drive inhibit 15
8.7	Drive inhibit 8	9.7	Drive inhibit 16
10.0	Drive OFF 1	11.0	Drive OFF 9
10.1	Drive OFF 2	11.1	Drive OFF 10
10.2	Drive OFF 3	11.2	Drive OFF 11
10.3	Drive OFF 4	11.3	Drive OFF 12
10.4	Drive OFF 5	11.4	Drive OFF 13
10.5	Drive OFF 6	11.5	Drive OFF 14
10.6	Drive OFF 7	11.6	Drive OFF 15
10.7	Drive OFF 8	11.7	Drive OFF 16

NC-I	Signal designation	NC-I	Signal designation
12.0	Spindle inhibit 1	13.0	Spindle inhibit 9
12.1	Spindle inhibit 2	13.1	Spindle inhibit 10
12.2	Spindle inhibit 3	13.2	Spindle inhibit 11
12.3	Spindle inhibit 4	13.3	Spindle inhibit 12
12.4	Spindle inhibit 5	13.4	Spindle inhibit 13
12.5	Spindle inhibit 6	13.5	Spindle inhibit 14
12.6	Spindle inhibit 7	13.6	Spindle inhibit 15
12.7	Spindle inhibit 8	13.7	Spindle inhibit 16

A.1.H Spindle-related interface signals, NC –> PLC

NC-O	Signal designation	NC-O	Signal designation
14.0	SERCOS system ready	15.0	Spindle orientation active
14.1	Drive ready	15.1	Spindle is oriented
14.2	Drive under control	15.2	Progr. spindle speed reached
14.3	Spindle command	15.3	Spindle speed 100 %
14.4	Direction of rotation M4	15.4	Idling speed reached
14.5	Gear range switching active	15.5	Spindle stopped
14.6	Spindle in position	15.6	res.
14.7	res.	15.7	res.
16.0	Output gear range 1	17.0	C axis is active
16.1	Output gear range 2	17.1	Spindle speed limited
16.2	Output gear range 3	17.2	Spindle override 0%
16.3	Output gear range 4	17.3	Error diagn. class 1
16.4	res.	17.4	Class 2 diagnostics changed
16.5	res.	17.5	Class 3 diagnostics changed
16.6	res.	17.6	Position control active
16.7	Output gear in idle motion	17.7	C axis switching active
18.0	Index of coupling bit 0	19.0	res.
18.1	Index of coupling bit 1	19.1	res.
18.2	Index of coupling bit 2	19.2	res.
18.3	Spindle is master	19.3	res.
18.4	Synchronous 1	19.4	Control reset acknowledge (Spindle)
18.5	Synchronous 2	19.5	res.
18.6	Coupling error	19.6	res.
18.7	res.	19.7	res.
20.0	Customer output 1	21.0	SERCOS drive status bit 0
20.1	Customer output 2	21.1	SERCOS drive status bit 1
20.2	Customer output 3	21.2	SERCOS drive status bit 2
20.3	Customer output 4	21.3	SERCOS drive status bit 3
20.4	Customer output 5	21.4	SERCOS drive status bit 4
20.5	Customer output 6	21.5	SERCOS drive status bit 5
20.6	Customer output 7	21.6	SERCOS drive status bit 6
20.7	Customer output 8	21.7	SERCOS drive status bit 7

NC-O	Signal designation	NC-O	Signal designation
22.0	res.	23.0	res.
22.1	res.	23.1	res.
22.2	res.	23.2	res.
22.3	res.	23.3	res.
22.4	res.	23.4	res.
22.5	res.	23.5	res.
22.6	res.	23.6	res.
22.7	res.	23.7	res.
24.0	res.	25.0	res.
24.1	res.	25.1	res.
24.2	res.	25.2	res.
24.3	res.	25.3	res.
24.4	res.	25.4	res.
24.5	res.	25.5	res.
24.6	res.	25.6	res.
24.7	res.	25.7	res.
26.0	res.	27.0	res.
26.1	res.	27.1	res.
26.2	res.	27.2	res.
26.3	res.	27.3	res.
26.4	res.	27.4	res.
26.5	res.	27.5	res.
26.6	res.	27.6	res.
26.7	res.	27.7	res.

A.2 Overview of version dependency of the APS modules for ICL 700

The program modules included in the APS are not available for all software versions of the PNC-R with ICL700. The column ”**new in version**” indicates the PNC-R software version in which the module can be first used.
In the PNC-R with iPCL and in PNC-P control units, all program modules are available at any time.

For more information on the contents of the changes, see the readme file and the files in the ”.../APS/Profi” directory on the ”**SWOSA3 V7.x.x.x**” installation CD for the PNC-R under the heading ”APS function modules for ICL700”.

	APS module designation	**Description**	**New as of version**	**Functionality modified as of software version**									
b	B01APSMN.PAH	server module (bit interface)	**2.1.2**	2.2.6			3.2.0	3.3.0	4.1.0 4.1.7			5.1.18	
b	B02AXPOS.PAH	read actual axis values	**2.1.2**										
b	B03PANAB.PAH	external program selection / deselection	**2.2.0**		2.3.4 2.3.8								
b	B04SATZV.PAH	NC block input	**2.0.0**									5.1.18	
b	B05TKOMP.PAH	temperature comp.	**2.2.2**									5.1.18	
b	B06LGANA.PAH	logic analyzer	**2.0.0**			2.4.1							
b	B07MPARA.PAH	read MACODA parameters	**2.0.0**										
b	B10SEROP.PAH	open/close serial interface	2.3.0		2.3.0 2.3.11								
b	B11SERIO.PAH	data read-in/read-out	**2.3.0**		2.3.0								
b	B14HANDW.PAH	Transmission of handwheel data with PROFIBUS-DP	**7.3.0**										
b	B15IDTNR.PAH	read SERCOS ident. number	3.2.0										5.2.2
b	B20WZLST.PAH	edit tool list	2.1.2							4.2.6			
b	B21WZDAT.PAH	edit tool data	2.1.2						4.1.7				
b	B22WZKOR.PAH	external tool compensation	2.1.2	2.2.5								5.1.18	
b	B23XTNPV.PAH	external zero shift	3.1.0										5.5.2
b	B24FLTAB.PAH	**not available any more as of V 5.1.18!**	4.3.4								4.3.7	5.1.18	
	B50R7REQ.PAH	call FMS services	**2.1.2**	2.2.6	2.3.0 2.3.4	2.4.1							
	B51R7CON.PAH	read and acknowledge FMS services	**2.1.2**	2.2.6									
	B52R7IND.PAH	FMS services	2.0.0										
b	B55FIFOC.PAH	communication with FIFO modules (CC10/Rho)	4.2.6							4.2.6			
b	B97ITOA.PAH	integer → ASCII conversion	4.3.7								4.3.7		
b	B98DATTM.PAH	read system date/time	**4.2.6**							4.2.6			

b = APS functions described in this manual

A.3 Transferring an existing ICL project into an iPCL project

★ Creating a new project with the **iPCL** control type with transfer of all modules from the ICL project:

- Delete any existing APS modules of the ICL.
- Create a library directory with the APS modules of the iPCL.
- Replace the OM2 by the OM2iPCL module and adapt the entries in the OM2 in the project directory and in the symbol table.

After the changes, the iPCL project can be created fault–free.
It is operable with the iPCL if the PLC program contains neither direct accesses II, IO nor extended inputs or extended outputs.

Additional differences between ICL and iPCL

APS modules no longer required in an iPCL project:

- The module B01APSMN does not have to be called in the PLC program anymore.
- The modules NCSLIBW, ICLLIB and B06LGANA do not have to be linked to the program anymore either.

Although these modules exist for reasons of compatibility, they do not contain any logic.
These modules can be removed from the symbol file.

No additional inputs and outputs:

- The iPCL has no interrupt inputs and outputs.

Remanence behavior when switching off the control unit:

For a remanent start, the function in DW2 of the OM2 has to be activated and the area limits in DW7 – DW10 (DW33 additionally in case of iPCL) have to be defined.

- **ICL:** In the ICL, all operands and data modules are located in the static RAM of the module and are therefore automatically remanent (according to setting in OM2).
- **iPCL**: In contrast to the ICL, remanent data modules have to be marked as remanent; the number of remanent data modules is limited to 128.
  - PNC-P:
    The data are saved when the control unit or the PC is shut down, and the remanent data are stored on the hard disk in accordance with the setting in OM2 or according to the symbol file in case of data modules and are loaded automatically when a restart takes place.
  - SNCI:
    Since switching off the SNCI module does not allow for a controlled shutdown, an automatic saving of the remanent data is not possible here. The remanent data therefore have to be saved by the PLC user program.

In OM2, it is possible to activate a cyclic saving of the remanent data with the consequence that all remanence areas defined in the DW3 are saved after each PLC cycle. Considering that large quantities of remanence data noticeably increase the cycle time, this should only be made use of for small remanence data quantities.
As an alternative, it is possible to use special commands to store specific areas in the static RAM when this area has been changed (see ”iPCL System Description and Programming Manual”, Chapter 7.3 Save/load remanence areas).

A.4 Index

A
Activating messages, MSD, 11–3
APS, 2–4
Auxiliary functions
  BCD–coded channel–independent, 9–3
  bit–coded, 9–1
Axis–related interface signals, 6–1
  Overview of input signals NC–I, 6–1
  Overview of output signals NC–O, 6–3
  Signal description NC–I, 6–5
  Signal description NC–O, 6–17
  Start addresses in MACODA, 6–1

C
Channel–related interface signals, 5–1
  Overview of input signals NC–I, 5–1
  Overview of output signals NC–O, 5–3
  Signal description NC–I, 5–4
  Signal description NC–O, 5–15
  Start addresses in MACODA, 5–1
Communication, PLCNC via APS, 3–1
Communication structure, 2–1
Communication with FIFO modules, Program module (–B55FIFOC), 12–72
Conversion Integer –> ASCII, Program module (–B97ITOA), 12–77

D
Data read–in/read–out, Program module (–B11SERIO), 12–35
Digital and analog I/Os, 10–4
Documentation, 1–7
Double word DW0, Program module parameter, 12–2
Double word DW1, Program module parameter, 12–4
Double word DW2, Program module parameter, 12–5
Double word DW3, Program module parameter, 12–5
Double word DWx, Program module parameter, 12–5
DP master interface
  for ICL700, 10–1
  for iPCL, 10–1
DP master status, 10–3

E
Edit tool data, Program module (B21WZDAT), 12–55
Edit tool list, Program module (–B20WZLST), 12–49
EMC Directive, 1–1
EMERGENCY–STOP devices, 1–5
Error code, 12–4
ESD
  Electrostatic discharge, 1–6
  grounding, 1–6
  workplace, 1–6
ESD–sensitive components, 1–6
Ethernet coupling, 2–1
External program selection/deselection, Program module (–B03PANAB), 12–10
External tool compensation, Program module (–B22WZKOR), 12–60
External zero shift, Program module (–B23XTNPV), 12–66

G
General interface signals, 4–1
  Overview of input signals NC–I, 4–1
  Overview of output signals NC–O, 4–1
  Signal description NC–I, 4–2
  Signal description NC–O, 4–6
  Start address in MACODA, 4–1
Grounding bracelet, 1–6

H
Handwheel
  Connection with CAN, 13–11
  Connection with PROFIBUS–DP, 13–4
Handwheel data, Program module (–B14HANDW), 12–40
High–level language modules, 2–5

I
Intended use, 1–1
Interfaces
  APS (application software), 2–4
  NC interface (NCS), 2–1
iPCL, Compatibility with ICL 700, 2–4

L
Logic analyzer, Program module (–B06LGANA), 12–23
Low–Voltage Directive, 1–1

M
Machine error and status display, 11–1
Machine operating panel
  Activating, 13–6
  Handwheel connection, 13–4, 13–11
  with CAN bus, 13–5
  with PROFIBUS–DP, 13–1
Modules sensitive to electrostatic discharge. *See* ESD–sensitive components
MSD file, 11–1

N
NC block input, Program module (–B04SATZV), 12–15
NC interface (NCS), 2–1
NFS file management, 2–1

O
Open/close serial interface, Program module (–B10SEROP), 12–28
Overview, integrated PLC, 2–1

P
PROFIBUS–DP, 10–1
  Commissioning for the PNC–R and PNC–P (iPCL), 10–2
  Commissioning for the PNC–R and PNC–P (KNS), 10–1

Program modules, 3–1
    Definition, 12–1
    iPCL and ICL700, 12–1

Q

Qualified personnel, 1–2

R

Read actual axis values, Program module (–B02AXPOS), 12–6
Read MACODA parameters, Program module (–B07MPARA), 12–24
Read system date/time, Program module (–B98DATTM), 12–78
Read/Input SERCOS ident. number, Program module (–B15IDTNR), 12–43
Release, 1–7

S

Safety instructions, 1–4
Safety markings, 1–3
Signal description NC–I
    Axis–related interface signals, 6–5
    Channel–related interface signals, 5–4
    General interface signals, 4–2
    Spindle–related interface signals, 7–5
Signal description NC–O
    Axis–related interface signals, 6–17
    Channel–related interface signals, 5–15
    General interface signals, 4–6
    Spindle–related interface signals, 7–15
Signal transmission
    Machine operating panel with CAN bus, 13–9
    Machine operating panel with PROFIBUS–DP, 13–2
Spare parts, 1–6
Spindle speed, external, 8–1
Spindle–related interface signals, 7–1
    Overview of input signals NC–I, 7–1
    Overview of output signals NC–O, 7–3
    Signal description NC–I, 7–5
    Signal description NC–O, 7–15
    Start addresses in MACODA, 7–1
Start address
    Axis–related interface, 6–1
    Channel–related interface, 5–1
    General interface, 4–1
    Spindle–related interface, 7–1
System architecture, 2–2

T

Temperature compensation, Program module (–B05TKOMP), 12–19
Test activities, 1–5
Trademarks, 1–8

Notes:

Rexroth
Bosch Group

Bosch Rexroth AG
Electric Drives and Controls
Postfach 11 62
64701 Erbach
Berliner Straße 25
64711 Erbach
Deutschland
Tel.: +49 (0) 60 62/78-0
Fax: +49 (0) 60 62/78-4 28
www.boschrexroth.com

Australia

Bosch Rexroth Pty. Ltd.
3 Valediction Road
Kings Park NSW 2148
Phone:+61 (0) 2 98 31 77 88
Fax: +61 (0) 2 98 31 55 53

USA

Bosch Rexroth Corporation
5150 Prairie Stone Parkway
Hoffmann Estates, Illinois 60192
Phone:+1 (0) 847 6 45-36 00
Fax: +1 (0) 847 6 45-08 04

United Kingdom

Bosch Rexroth Ltd.
Broadway Lane, South Cerney
Cirencester GL7 5UH
Phone:+44 (0) 1285-86 30 00
Fax: +44 (0) 1285-86 30 03

Canada

Bosch Rexroth Canada Corp.
490 Prince Charles Drive South
Welland, Ontario L3B 5X7
Phone:+1 (0) 905 7 35-05 10
Fax: +1 (0) 905 7 35-56 46

1070 073 741-112 (02.10) GB · HB NC · BRC/ESM11 · Printed in Germany