SIEMENS

SINUMERIK 840D/840Di/810D

Tool management

Description of Functions

Valid for

Control
SINUMERIK 840D powerline/840DE powerline
SINUMERIK 840Di powerline/840DiE powerline
SINUMERIK 810D powerline/810DE powerline

Software	
NCU system software for 840D/840DE	7.3
NCU system software for 840Di/840DiE	2.3
NCU system software for 810D/810DE	7.3

Version 09.05

SINUMERIK® Documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the “Remarks” column.

Status code in the “Remarks” column:

A . . . . . New documentation.
B . . . . . Unrevised reprint with new Order No.
C . . . . . Revised edition with new status.
If the technical subject matter shown on the page has changed compared to the previous edition status, this is indicated by the changed edition status in the header of the respective page.

Edition	Order No.	Remarks
04.95	6FC5 297–2AC60–0BP0	**A**
09.95	6FC5 297–3AC60–0BP0	**C**
03.96	6FC5 297–3AC60–0BP1	**C**
08.97	6FC5 297–4AC60–0BP0	**C**
12.97	6FC5 297–4AC60–0BP1	**C**
12.98	6FC5 297–5AC60–0BP0	**C**
08.99	6FC5 297–5AC60–0BP1	**C**
07.00	6FC5 297–5AC60–0BP2	**C**
05.01	6FC5 297–6AC60–0BP0	**C**
11.02	6FC5 297–6AC60–0BP1	**C**
09.05	6FC5 297–7AC60–0BP0	**C**

Trademarks
SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIROTEC®, SINUMERIK® and SIMODRIVE® are registered trademarks of Siemens AG. The other designations in this publication may also be trademarks, the use of which by third parties may constitute copyright violation.

Further information is available in the Internet under:
http://www.siemens.com/motioncontrol

This publication was produced with Interleaf V 7

Other functions not described in this documentation may be executable in the control. However, no claim can be made regarding the availability of these functions when the equipment is first supplied or in the event of servicing.

We have checked that the contents of this document correspond to the hardware and software described. Nevertheless, differences might exist and we cannot, therefore, guarantee that they are completely identical. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are also welcome.

Subject to change without prior notice.

Order No. 6FC5 297–7AC60–0BP0
Printed in Germany

Siemens Aktiengesellschaft

Preface

SINUMERIK® Documentation

The SINUMERIK documentation is organized in three parts:

- General Documentation
- User Documentation
- Manufacturer/Service documentation

More detailed information about other SINUMERIK 840D/840Di/810D brochures, and brochures for all SINUMERIK controllers (e.g. universal interface, measuring cycles, etc.) can be obtained from your local Siemens representative.

An overview of publications, which is updated monthly and also provides information about the language versions available, can be found on the Internet at:
http://www.siemens.com/motioncontrol
Follow menu items – “Support” –> “Technical Documentation” –>
“Overview of Documentation”.

The Internet version of DOConCD (DOConWEB) is available at:
http://www.automation.siemens.com/doconweb

Target audience

This document is designed for machine tool manufacturers. The documentation provides a detailed description of the functions necessary to operate the SINUMERIK 840D/810Di/810D and SIMODRIVE 611D controls.

Standard version

This documentation only describes the functionality of the standard version. Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

Hotline

If you have any questions, please get in touch with our hotline:

A&D Technical Support
Phone: +49 (0) 180 / 5050 – 222
Fax: +49 (0) 180 / 5050 – 223
E-mail: mailto:adsupport@siemens.com
Internet: http://www.siemens.de/automation/support-request

If you have any queries (suggestions, corrections) in relation to this documentation, please fax or e-mail us:

Fax: +49 (0) 9131 / 98 – 63315
E-mail: mailtomotioncontrol.docu@siemens.com

Fax form: Refer to the reply form at the end of the document.

Internet address

http://www.siemens.com/motioncontrol

Structure of the manual

The Description of Functions is structured as follows:

- General table of contents
- Descriptions of functions, installation and start-up, programming, data backup, data and alarms, PLC sample programs
- Appendix with list of abbreviations, terms and references
- Index

Notice

The page numbers provide the following information:
Part of Description of Functions / Book / Section – Page

If you require information about a function, you will find the function and the code under which it is classified in the inside cover title of the manual.

If you need information about a certain term, please go to the section headed Index in the Appendix and look for the term concerned. The Description of Functions code, the chapter number and the number of the page on which you can find the information you need are listed in this section.

Within each of the Description of Functions in Chapters 4 and 5 you will find definitions on effect, data format, input limits etc. for the various signals and data definitions.
These definitions are explained in the “Technical comments” section below.

SW version

The software versions indicated in the documentation relate to the SINUMERIK 840D controller. The software version valid in parallel for the SINUMERIK 810D controller (if the function has been enabled, see /OI/, Catalog NC 60) is not indicated specifically.

Symbols

!

Important

This symbol is always displayed in this document to draw your attention to an important item of information.

Ordering data option

In this documentation you will find the symbol shown on the left with a reference to an ordering data option. The function described is executable only if the control contains the designated option.

Machine manufacturer

This pictorial symbol appears in this document to indicate that the machine manufacturer can control or modify the function described. See machine manufacturer's specifications.

Safety Information

This Manual contains information which you should carefully observe to ensure your own personal safety and prevention of material damage. Notes relating to your personal safety are highlighted in the manual by means of a warning triangle, no warning triangle appears in conjunction with notes that relate to property damage. The warnings are shown below in decreasing order of danger.

Danger

Indicates that death or severe personal injury **will** result if proper precautions are not taken.

Warning

Indicates that death or severe personal injury **may** result if proper precautions are not taken.

Caution

With a warning triangle indicates that minor personal injury can result if proper precautions are not taken.

Caution

Without a warning triangle means that material damage can occur if appropriate precautions are not taken.

Notice

Indicates that an undesirable event or state may arise if the relevant notes are not observed.

If several hazards of different degrees occur, the hazard with the highest degree must always be given preference. If a warning note with a warning triangle warns of personal injury, the same warning note can also contain a warning of material damage.

Qualified Personnel

Start-up and operation of the device/equipment/system in question must only be performed using this documentation. Commissioning and operation of a device/ system may only be performed by **qualified personnel**. Qualified personnel as referred to in the safety guidelines in this documentation are those who are authorized to start up, earth and label units, systems and circuits in accordance with the relevant safety standards.

Intended use

Please note the following:

Warning

The unit may be used only for the applications described in the catalog or the technical description, and only in combination with the equipment, components and devices of other manufacturers where recommended or permitted by Siemens. To ensure trouble-free and safe operation of the product, it must be transported, stored and installed as intended and maintained and operated with care.

Technical information

Notations

The following notation and abbreviations are used in this documentation:

- PLC interface signals –> IS “Signal name” (signal data)
  Example:
  - IS “MMC-CPU1 ready” (DB10, DBX108.2) i.e. the signal is stored in data block 10, data byte 108, bit 2.
  - IS “feed/spindle override” (DB31-48, DBB0) i.e. the signals are stored for each axis/spindle in data blocks 31 to 48, data block byte 0.
- Machine data –> MD: MD_NAME (German name)
- Setting data –> SD: SD_NAME (German name)
- The symbol “≙” means “corresponds to”.

Order codes

Chapters Machine Data and Signal Description provide an explanation of the data and signals which are important for the respective function. This information, which is provided in table format, includes a number of terms and abbreviations, which are explained here.

Values in the table

The machine data presented in the Descriptions of Functions always represent the values for an NCU572.2.

The values for a different NCU (e.g. NCU570, NCU571, NCU573) are contained in the List Manual.

References: /LIS/, “Lists”

Default value

The machine data/setting data is preset to this value during startup. If default values for the channels differ, this is indicated by “/”.

Value range

Specifies the input limits. If no value range is specified, the data type determines the input limits and the field is marked “***”.

Changes

Changes made to machine data, setting data, etc. do not take immediate effect in the control. The conditions for such changes to take effect are always indicated. The possible options are listed in order of priority below:

- POWER ON (po)
  - “RESET” key on front panel of NCU module, or disconnection/reconnection of power supply
- NEW_CONF (cf)
  - “New configuration” function of the PLC interface
  - “RESET” button on the control unit
- RESET (re) “RESET” key on control unit
- Immediately (im) after entry of the value

Protection level

Protection levels 0 to 7 have been used. The lock for protection levels 0 to 3 (4 to 7) can be canceled by entering the correct password (setting the correct keyswitch position). The operator only has access to information protected by one particular level and the levels below it. The machine data is assigned different protection levels by default.

Only the write protection level appears in the table. However, there is a fixed assignment between write and read levels:

Write protection level	Read protection level
0	0
1	1
2	4

References: /BA/ Operator’s Guide MMC
/FB/ A2, Various Interface Signals

Unit

The unit refers to the default setting of machine data SCALING_FACTOR_USER_DEF_MASK and SCALING_FACTOR_USER_DEF. If there is no physical unit set in the MD, the field is marked with “–”.

Data type

The following data types are used in the control system:

- DOUBLE
  Real or integer values (decimal values or integers), input limits from
  $+/-4,19*10^{-307}$ to $+/-1,67*10^{308}$
- DWORD
  Integers
  Input limits from $-2,147*10^{9}$ to $+2,147*10^{9}$
- BOOLEAN
  Possible input values: true or false/0 or 1
- BYTE
  Integers from –128 to +127
- STRING
  consisting of max. 16 ASCII characters (uppercase letters, digits and underscore)

Data management

The explanations of the PLC interface in the individual Descriptions of Functions assume a theoretical maximum number of components:

- 4 mode groups (associated signals stored in DB11)
- 8 channels (associated signals stored in DB21–30)
- 31 axes (associated signals stored in DB31–61)

For details of the actual number of components which can be implemented with each software version, please refer to

References: /FB/ K1, BAG, Channel, Program control

■

Notes

Table of contents

Introduction 1

Tool management (TOOLMAN) ensures that at any given time, the correct tool is in the correct location and that the data assigned to the tool is up to date. This function is used on machine tools with circular, chain or box magazines. As well as speeding up tool changes, it avoids scrap by monitoring tool service life and machine downtimes by using spare tools.

Tool-management functions

When dealing with tools, there are 4 types of function:

- **TMBF** **T**ool **M**anagement **B**ase **F**unctions
  Default in NCK
  (TMBF = Tool Management Base Functions)
- **TMFD** **T**ool **M**anagment **F**lat **D** numbers
  (TMFD = Tool Management Flat D Numbers)
- **TMMO** **T**ool **M**anagement **Mo**nitoring function
  (TMMO = Tool Management Tool Monitoring)
- **TMMG** **T**ool **M**anagement **M**a**g**azines
  (TMMG = Tool Management Magazines)

Included in the basic version of SINUMERIK 840D/840Di/810D are:

- TMBF or:
- TMBF + TMFD

Available with the tool management option are:

- TMBF + TMMO + TMMG

The function is capable of managing up to 30 real magazines with a total of 600 magazine locations and 600 tools, and up to 12 edges per tool (max. 1500 tool edges). The maximum number of edges per tool depends on the software version (12 edges in SW version 5.1 and later) and machine data settings.

With HMI-Advanced, the most user-friendly configuration, the full range of tool management functions is available. But even with an OP 030 or HMI Embedded, the main functions can be utilized on a task-related basis.

New structure

The range of functions to be executed by the tool management system has been extended ever further over time. A new structure will be selected in future based on the following categories:

TMBF
Basic functions of tool management (available even when tool management is not active)
TMMO
Tool monitoring
TMMG
Tool magazine management (only available when tool management is active)
TMFD
Tool Management with Flat D numbers (only without active tool management)

Main tool management functions (standard)	HMI Advanced	HMI Embedded	OP 030
System diagrams in the standard software	X	X	X
Options for configuring screenforms and softkeys	X		
Easy start-up via system displays	X		
Editing of tool data	X	X	X
Magazine and tool list	X	X	X
Vacant position search and positioning	X	X	X
Loading and unloading of tools	X	X	X
Easy search for empty locations using softkeys	X		
More than one magazine is possible	30	30	30
More than one loading and unloading point per magazine	X		
Tool cabinet and tool catalog	X		
“Relative” D numbers with user-customizable numbering	X		
Adapter data	X		
Local offsets	X		
Loading and unloading via code carrier system	X		
Data backup via RS-232 (V.24) interface	X	X	
Data backup on hard disk	X		

1.1 General overview

This document describes the scope of the tool management functions. Tool management functions are included in MMC, NCK and PLC. The appropriate functions are shown in the function structure (see Section 2.1). The tool management is divided into several sub-areas, which were outlined in the introduction.

Basic functions

The TMBF area of the tool management contains the basic functions. These basic functions are generally available, even in systems without an active TM system.
Basic functions include, for example, creating and deleting tools, entering offsets and tool changes. On the basic function level, a specific number (max. 12) of tool edges (D numbers) is assigned to each T number (tool identification).
Alternatively, the function TMFD or “Flat D numbers” (freely selectable D number independently of the T number) can be activated in systems without active tool management. You can select any number of tool edges per tool; the number of edges per tool is not limited to 12. With “Flat D Numbers”, the user is responsible for management and assignment of T numbers to D numbers.

Miscellaneous functions

The miscellaneous functions of tool management are magazine management, tool and location search and monitoring the tool life, workpiece numbers or wear values. These miscellaneous functions are only available when tool management (option) is active.
Magazine management functions must be implemented by the machine manufacturer in systems without active tool management. These will generally be executed via the PLC.

Magazine management

“Magazine management” refers to the administration of magazine locations. These locations might be empty, loaded with tools or assigned to oversized tools in adjacent locations.
Empty locations can be “loaded” with other tools. The tool management function provides the machine manufacturer with a function for optimized management of tools/magazine locations.
Magazine management provides extended functions such as load, unload or position tools and includes searches for tools, magazine locations and search strategies for replacement tools. After expiry of the monitoring time activated in the tool monitoring function, tools are disabled and no longer used. If tool monitoring detects an existing identical tool (duplo tool) which is not disabled, this tool is then automatically used for any further machining.

In the simplest case scenario, all that needs to be configured when tool management is activated are magazines, loading magazines, spindles, grippers, etc. Furthermore, the interfaces (DB 71 to DB 73) must be processed in the PLC (see Section 2.3).
Task-related tool motions (e.g. position chain, swivel gripper) are derived from the interface processing. On completion of tool motions, the positions and task status are acknowledged via basic program blocks (FC 7 and FC 8). If necessary, create another cycle (or ASUB) for the NC program where tool changing is programmed with the required travel motions. An identifier is programmed for the tool change or tool preselection when the TM system is active. A duplo number is also available to support unique identification of replacement tools. Tool identifier and duplo numbers are always mapped on an internally assigned T number. This internally assigned T number is used for addressing the variables described in the following.

OPI variables

Additional functions are available by using OPI variables (see Section 2.2, PI commands) from the PLC or HMI (see Section 5.12.5). The NC program (e.g. cycle, ASUB) provides corresponding language commands to achieve optimum adaptation of the tool management to the machine environment. You can obtain a clear overview from the data structures that form the basis of the tool management. They are represented in the form of NC data blocks.

1.2 Key data

Operator panels

The following operator panels can be used for tool management (TOOLMAN):

- HMI Embedded
  Two interfaces are available:
  – Standard
  – Shopfloor-oriented (Shopmill)
- HMI Advanced
- OP 030 e.g. for loading magazines

The HMI Embedded and HMI Advanced operator interface functionality differs in the following areas:

- Start-up
- Data backup on hard disk
- Operation from configurable screens

The following are not implemented in HMI Embedded:

- Configuring of user softkeys for empty location search
- Start-up via system displays
- More than one loading and unloading point per magazine
- Tool cabinet, tool catalog
- “Relative” D numbers with user-customizable numbering
- Adapter data
- Location offsets
- Loading and unloading via code carrier system

Data

Data storage and management is carried out in the NC and HMI Advanced. All data can be read and written manually, via the NC program or by data transfer.

Operation

Operation is performed via system screens. These include screens for start-up (HMI Advanced only) and screens for tool management operation (magazine lists, tool lists, loading/unloading).

Programming in the NC part program

The tool management function makes it possible to call a tool in the part program using a name (identifier), e.g. T = "end mill 120mm".

Tool call is still possible via the T No. tool number. The T No. is then the name of the tool (e.g. T=12345678).
A tool is uniquely defined by its name and duplo number. Furthermore each tool can be unambiguously identified by its "internal" T number. The internal T number is as a rule assigned by NCK and is not used for programming a tool change in the main program.
The T call is the instruction to change the tools for the turret type of magazine.
In the case of a chain or box-type magazine, the T call is the instruction to prepare for the tool change. The M06 function loads the prepared tool into the spindle.

in software version 6 and higher

The following characters are permitted for the identifier:

[_] [a...z] [A...Z] [0...9] ; [+ – . ,]

Identifiers are case-sensitive, i.e. differentiate between uppercase and lowercase characters.

Notice

M06 is the CNC code generally used for tool change.

PLC

There are data blocks (DB71–73) for receiving tool management commands and function blocks (FC7, 8) for acknowledging the tool management commands.
Another block, FC22, is used as a direction selection for magazines.
Tool management data can also be read and written via FB2 and 3. Complex tool-management services can be initiated by FB4.

Magazine types

Circular, chain magazines and box magazines can be managed. Other magazine types, e.g. pick-up magazine, are mapped onto these.
Real magazines can be defined as a circular, chain or box-type. Loading points or loading stations shall be used as the magazine type for loading and unloading.
Type designation "magazine buffer" covers all other locations in which tools can be placed (spindle, gripper,...).

Location coding

Tools are supported both by fixed location coding and variable location coding.

Location type

The location type defines the type and shape of the location. By assigning location types to magazine locations it is possible to subdivide a magazine into areas. This means that different types of special tools, e.g. “especially_large”, “especially_heavy” can be assigned to specific locations.

The location types can be placed in ascending order or hierarchy. This order determines that a tool that is supposed to be inserted in a “small” location type can also be placed in a “larger” location type if no “smaller” location type is vacant.

Monitoring

In tool management, it is possible to select either workpiece counts or tool life monitoring (with reference to the cutting edges). Tool wear monitoring is also available with SW 5 and higher. Spare tools (duplo tools) are differentiated by means of a duplo number.

Search strategy

Customizable search strategies are available for tool change. Various strategies are possible for tool search and to search for empty location of the “old tool”.
You can still set a search strategy for loading tools.

Excerpt from TM basic data

Term	Data/Range
Magazine configurations per channel	1
Total number of magazines	max. 32
Total number of magazine locations	max. 600
Total number of tools	max. 600
Programming the tools in the NC program using an identifier (name) with 32 alphanumeric characters	e.g. T =“Angle head cutter_32”
Duplo no.	1 –32000
Total number of cutting edges	max. 1500
Location type definition	Yes
Consider adjacent location in half locations	2dimensional
Location coding	fixed or variable

Term	Data/Range
Strategy for tool search	can be set (programmed) via system variables
Strategy for location search	can be set (programmed) via system variables.
M06 command for tool change	M code, settable via MD, channel-specific
Tool change with M06 code or T command	settable via MD, channel-specific
Wear monitoring	for every cutting edge
Wear monitoring according to tool life	resolution msec
Wear monitoring according to number of workpieces	Counters
Access to tool mangement data via NC program	System variables
Automatic decoding stop until tool is selected.	Yes
T=Location No.	settable via MD

Option

Tool Management is an OPTION.

2 Overview

2.1 Function structure of tool management

HMI

- Tool data display, input/output
- Magazine data display, input/output
- Compensation data, input/output
- Tools and material management
  - Master data
  - Particular tool data
  - Code carrier
- Loading/unloading dialog

NCK

- Tool data management
  - Status
  - Monitoring
  - Corrections
- Magazine data management
  - Magazines
  - Magazine locations
- Tool management
  - Search for tool
  - Finding an empty location
  - Change tool
  - Load, unload

PLC

- Magazine control
- Gripper control
- Spindle control
- Safety interlocks
- Execute tool change
- Calculation of position, if necessary
- Special change strategy, if required

2.2 HMI/PLC – NCK data structure (OPI)

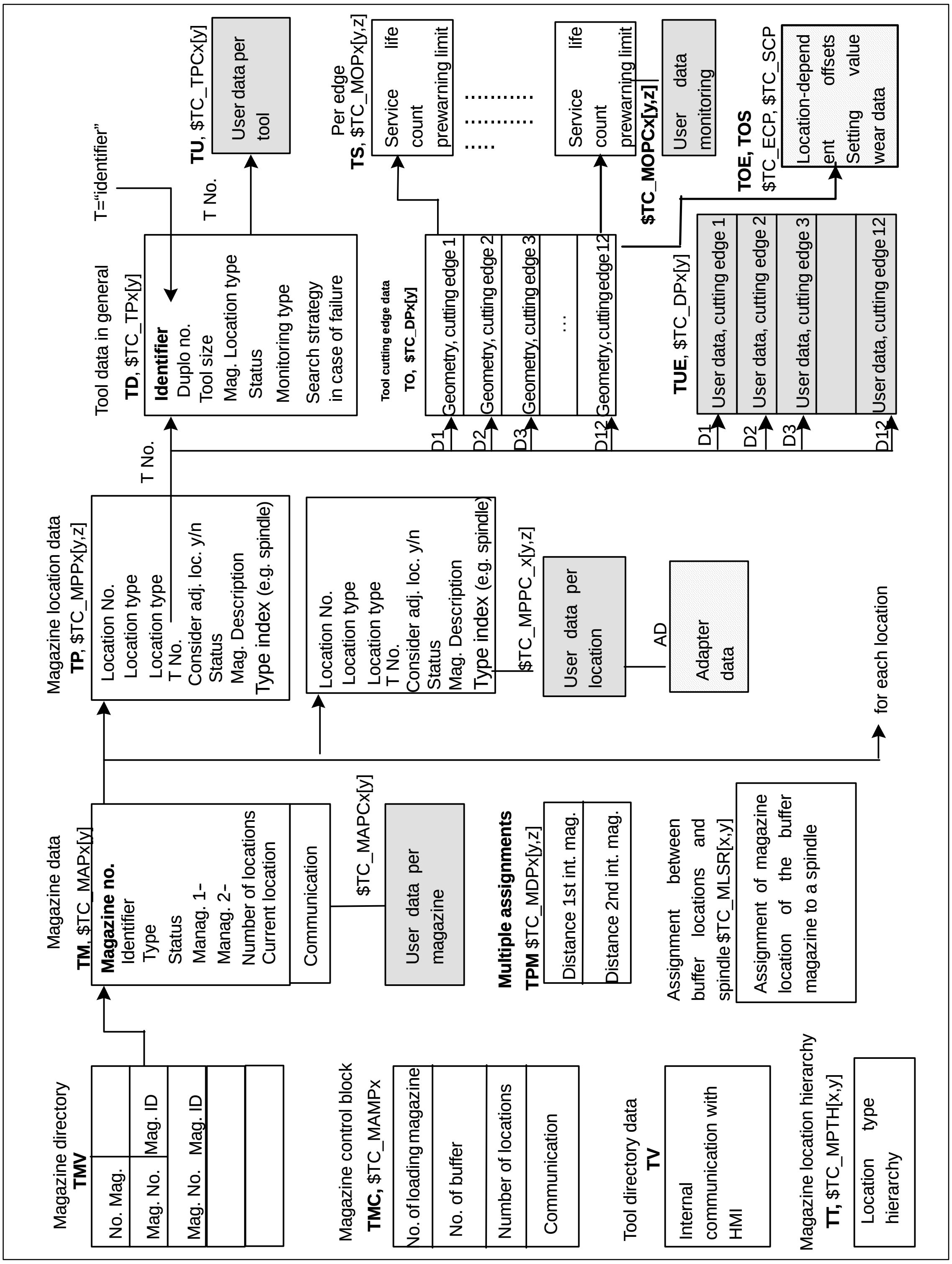


Bild 2-1 Structure of magazine data and tool data

Unchecked boxes mark the previous data of the tool management. Checked boxes show the user data.
New data blocks are displayed as checked and grayed boxes.

TOA area

A TOA area constitutes and independent group in tool management. There is no link existing to other TOA areas.
Up to 10 independent TOA areas may be created depending on the number of channels available. Several channels can be assigned to one TOA area but one channel cannot be assigned to more than one TOA area. A subset of magazines, buffer locations and loading magazines can be assigned to one TOA area.

2.3 PLC – NCK interfaces

Overview

The heart of the SINUMERIK 840D tool management system is located on the NCK. The PLC merely contains the interfaces for the machine-specific part (see Fig. 2-2).

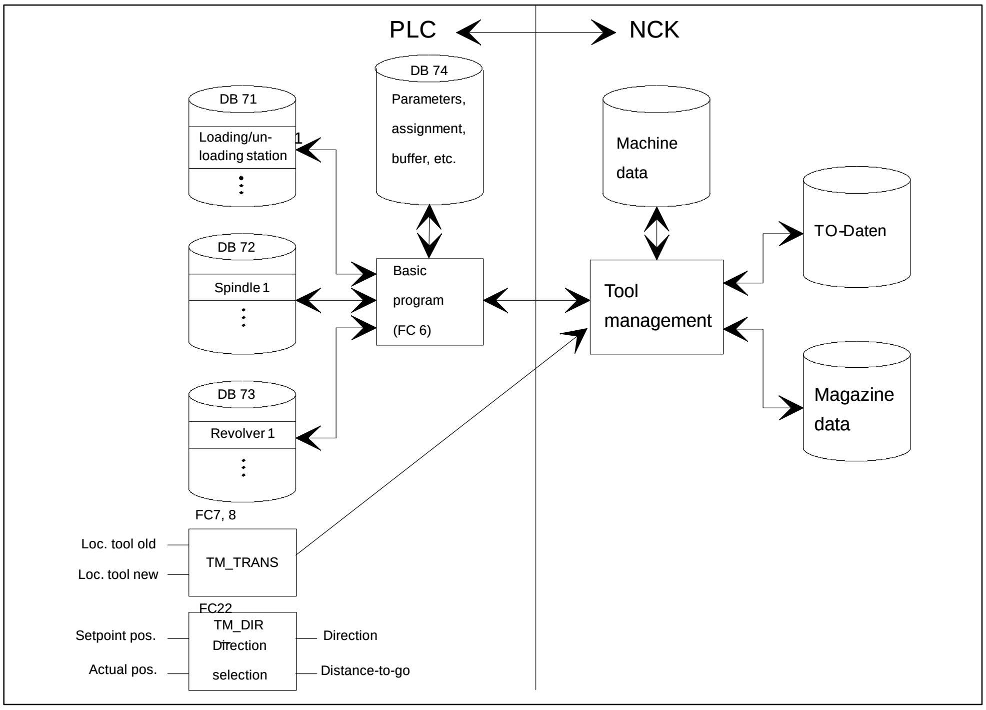


Bild 2-2 Data structure and PLC – NCK interface

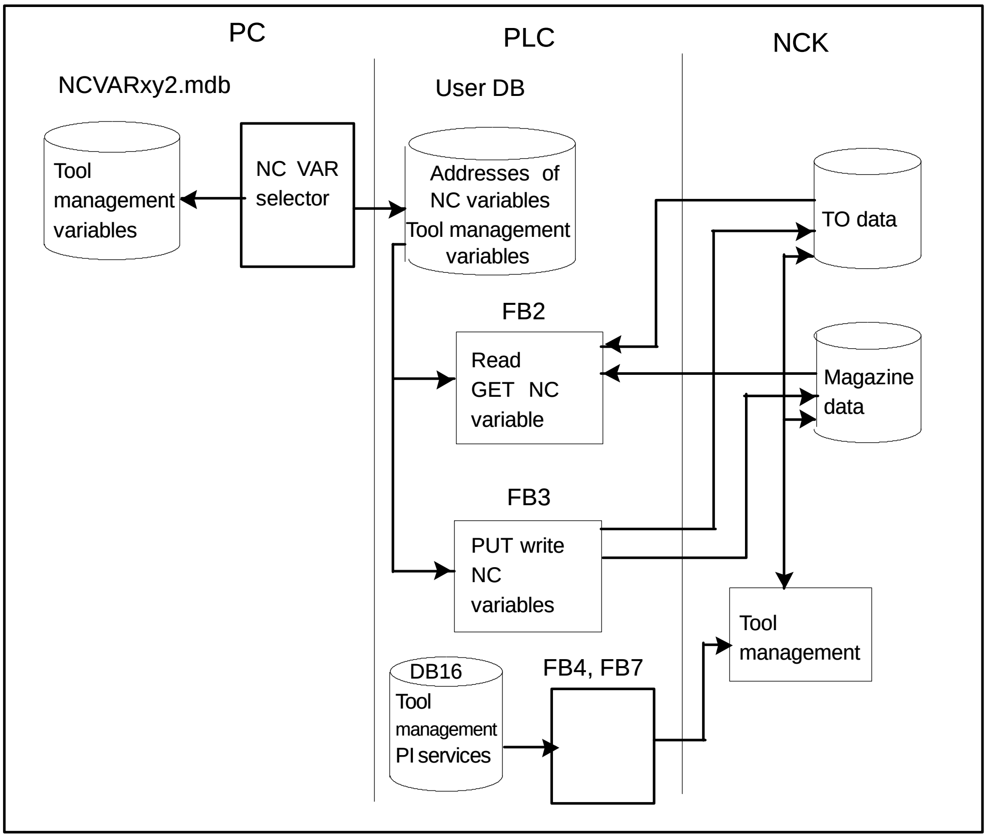


Bild 2-3 Extended interface for tool management between PLC and NCK

2.4 Magazine configuration

Magazine configuration

In one configuring process, it is possible to create a magazine configuration which includes one or several real magazines (actual magazine for storing tools, NCK is capable of managing several real magazines). All the magazines of one configuration can be operated simultaneously in one channel. Several magazine configurations can be defined but only one configuration can be active in one channel at one time.

Magazine and tool data are stored in the NC in the so-called TO area. The TO area can in turn be sub-divided by machine data into individual TO units. It must further be defined by machine data, which channel works or which channels work on which TO units. Only one magazine configuration can be active at any one time per TO units. If several channels are assigned to TO units, then the magazine configuration applies for all assigned channels.

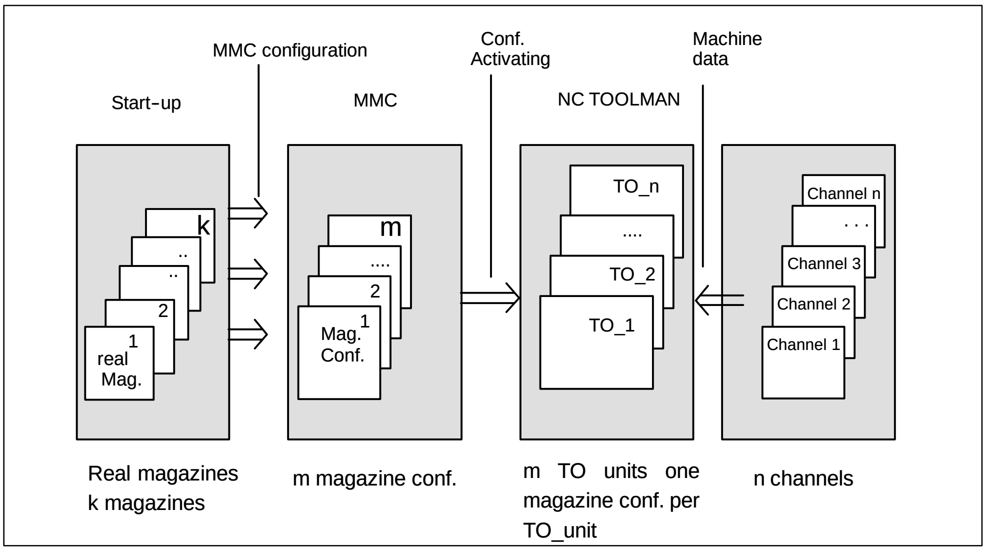


Bild 2-4 Assignment of magazines to channels

For more information, see Section 3.1 and 4.4.1.

2.5 Magazine list

The magazine list is a location-oriented map of the tool magazine, gripper and spindle. Tool management only works with the tools from the magazine list.

Additional tools without a magazine assignment can also be selected for tool changes. The tool must be inserted in the machine manually and removed again manually after machining (manual tool). The same applies to the tool list. For manual tools, see Section 3.2.11.

HMI

The structure of the magazine list (i.e. which data are to be displayed) is defined by the machine manufacturer via the PARAMTM.INI file. Up to three user-definable displays (screen forms) are available for dividing up the various types of data, e.g. offsets, wear, general data. These displays can be called up via their own softkeys. In the example below: *Maglist 1, Maglist 2, Maglist 3*. The tools that are located in the selected magazine are listed in the magazine list.

The magazine list configured under *Maglist 1* is the tool management main screen. All operations can be selected from this display. One magazine list is available for each channel.

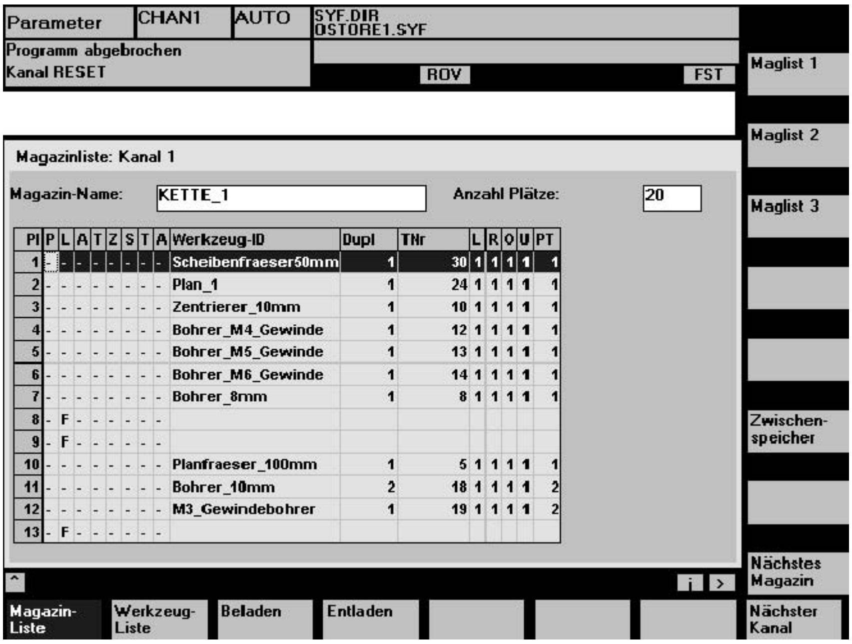


Bild 2-5 Example of a magazine list

2.6 Tool list

The tool list contains all the tools known to the NC. These are the tools in the magazine and tools which have been unloaded but whose data are to be retained.

The tool management function works with loaded tools from the magazine list.

HMI

The structure of the tool list is defined by the user. The data can be displayed in up to three user-definable screens. In the example below: *Tool list 1, Tool list 2, Tool list 3*.

In the tool list, all tools of the TO area are listed sorted according to the internal T no., i.e. even those tools that are not assigned to any magazine location. The display underneath the softkey "*Tool list 1*" is shown as a main screen.

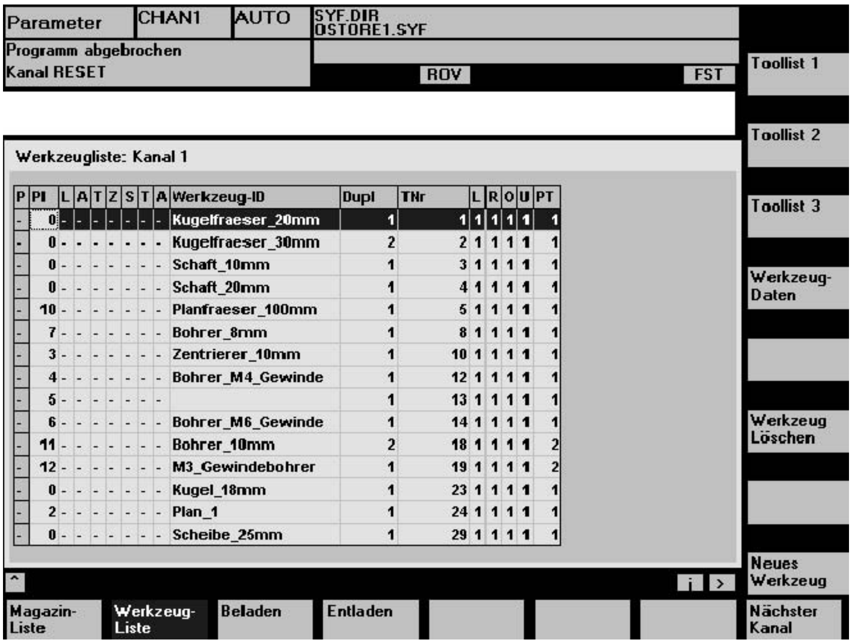


Bild 2-6 Example of a tool list

Modification of tool designations and duplo numbers in the lists

Renaming tools

The operator can change the tool name and the duplo number directly in the following screens: magazine, tool, working offset list and tool details. It is not possible to change the tool type directly in the working offset list.

Modifying tool identifier and duplo number

This function is defined with MD 9240: USER_CLASS_WRITE_TOA_NAME can be set to determine whether the user can modify the tool identifier and duplo number in lists or not. The default value is always 0. With this value, no changes can be made by the operator in the lists.

Changing tool type

MD 9241: USER_CLASS_WRITE_TOA_TYPE is set depending on whether the operator can change the tool type directly in the tool list, the magazine list and the tool details screen. The default value for the MD is 0. With this setting, the operator cannot make direct changes in the lists.

Notice

It is not possible to change the tool type directly in the working offset list. Changing the tool type of a cutting edge automatically changes the tool type of all cutting edges of the same tool. The cutting edges of a tool are not listed one after the other in the work correction list since these are sorted according to user-assigned D numbers.

Tool type function

If the operator changes the tool type of a cutting edge, the tool type of the other cutting edges of the same tool is also changed.

The following data are set to 0:

- Tool user data
- Tool compensation parameters of all cutting edges
  (the cutting-edge adapter data are not changed if the tool is at a magazine location and the function “Magazine location adapter data” is active at the NC.)
- Cutting edge user data of all cutting edges
- Cutting edge monitoring data of all cutting edges
- Location-dependent compensation parameters of all cutting edges (wear values and setup values)

Configuration of the modification procedure

Before the changes are made to the tool type, the operator is asked to confirm the changes. This prompt can be suppressed, depending on the current access rights, by setting the following data in paramtm.ini:

```
[ACCESSLEVEL]
ChangeToolTypeWithoutConfirmation=-1  ;value range -1 to 7
```

Confirmation is always requested with the default “–1”. Entering an access level (values 1–7) specifies the lowest access level at which the prompt is to appear.

Confirmation is requested with keyswitch “0” (access level 7). The prompt is suppressed with keyswitch “1” and higher (access level 6 and lower).

Example:

```
[ACCESSLEVEL]
ChangeToolTypeWithoutConfirmation=6  ;value range -1 to 7
```

New magazine list with multiple lines

Cutting edges in magazine list

Several lines are available for each tool in the magazine list. The edges for each tool are included in every magazine display.

2.7 Tool cabinet (HMI Advanced only)

Tool cabinet

The data of the tools employed can be stored in the tool cabinet. This data is called particular tool data. The data corrected when the tool was in operation in the NC can be stored in the tool cabinet while the tools are unloaded. The user can retrieve this data again when loading the tool. The user must, however, know the duplo number for the tool.

A complete set of tool data is kept in the cabinet for each individual tool in the control, called a selected tool. A selected tool is identified in the catalog by its technology, its tool type, its unique name for the tool type and its unique duplo no. (> 0) with regard to tool type and name. Each selected tool there has a different duplo no. even when technology, type and tool name are the same.

You can store or enter tool data for selected tools in the tool cabinet.

Notice

If the plant is powered down, softkey EXIT must be used. If it is not (e.g. power failure), the database can be corrupted. To avoid this happening, an uninterruptible power supply should be used.

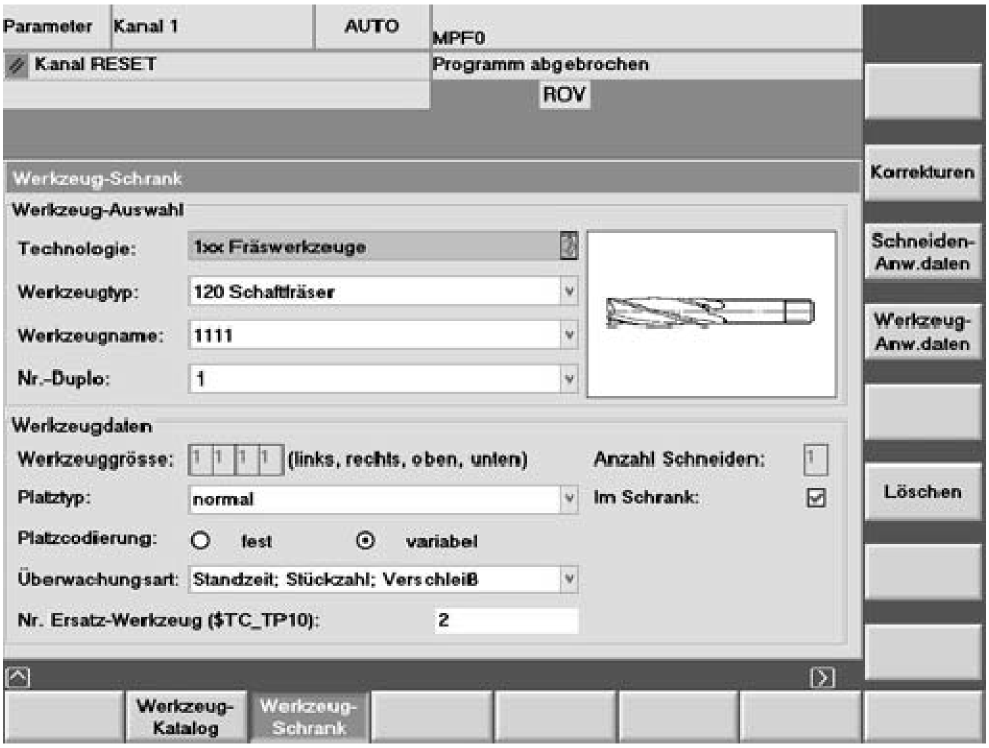


Bild 2-7 Example of tool cabinet

2.8 Tool catalog (HMI Advanced only)

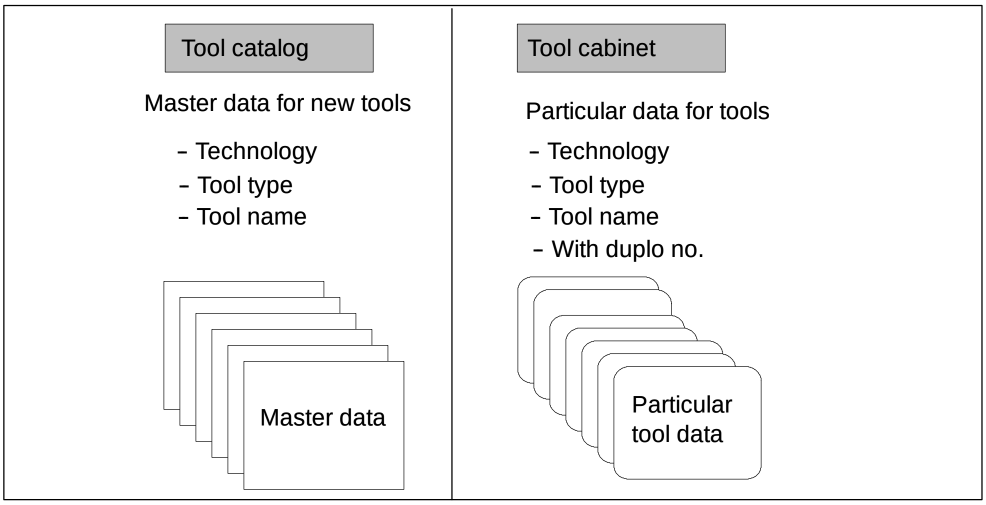


Bild 2-8 Differences between tool catalog and cabinet

Tool catalog

The tool catalog is empty when supplied. Tool data must be entered before a new tool can be loaded via the catalog. To this end, technology and tool type are selected and a tool name specified. After which the tool and cutting-edge data are entered.

As a result of this process, so-called “master data” have been set up for the tools.

When loading a new tool the user can call these master data. It is not possible to store the data of tools already used. There thus exists for each tool exactly one master data record of a certain technology, a certain type and a certain tool name. The tool master data are sorted in the tool catalog. The generally applicable tool data as well as the nominal and process information for the tools are listed in the tool master data.

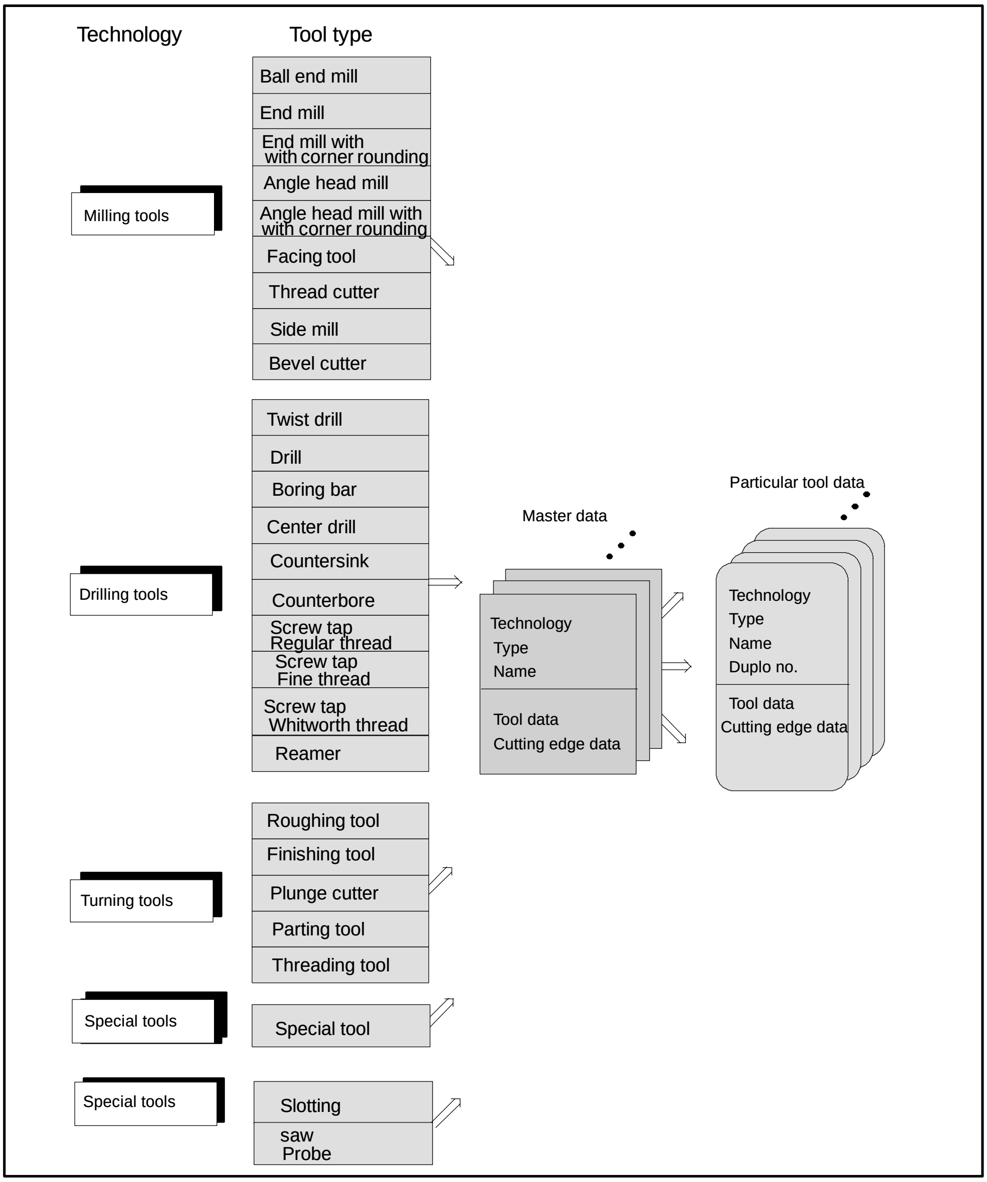


Bild 2-9 Structure of the tool catalog with master and operating data

The full list of tool types is contained in the Programming Guides.

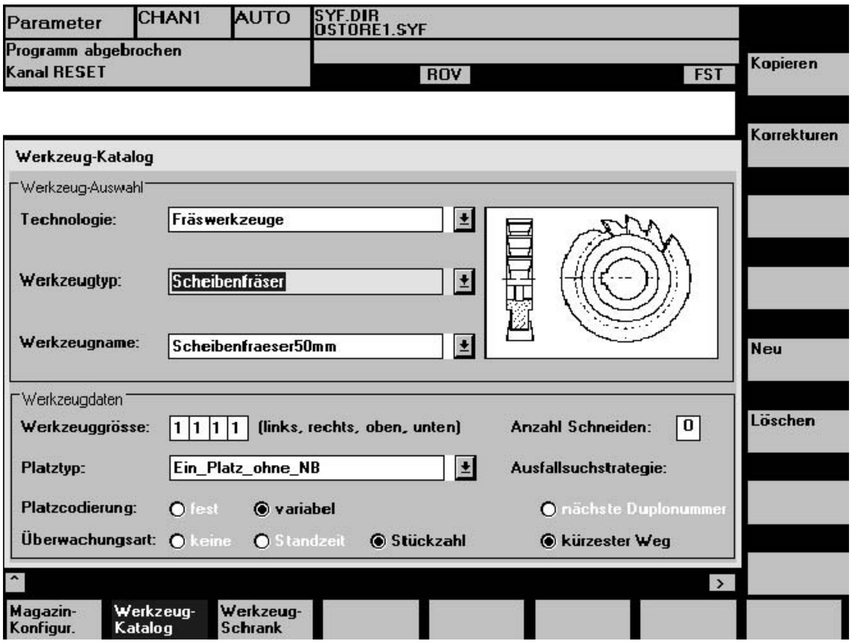


Bild 2-10 Example of tool catalog

Notice

Master data can only be read out of the control and transferred to another in their entirety.

Tool data:

The following data can be entered for every tool:

- Tool size e.g. 2222 (left, right, bottom, top)
- Location type
- Location coding (fixed, variable)
- Tool monitoring (none, tool life, workpiece count, wear)
- Number of cutting edges (display of defined edges only)
- Spare-tool search strategy (next duplo no., shortest path amongst others)

Cutting edge data:

- Cutting edge position
- Number of cutting edges (display only)
- Cutting edge number (display only)
- Offset parameters (geometry, wear, base)

- Monitoring data (set value, prewarning limit)

Notice

The database of the particular tool data can only be read out of the control and transferred to another control in its entirety.

2.9 Access protection, protection levels

The access to programs, data and functions is protected via 8 hierarchical levels according to customer requirements. These are divided into

- 4 password levels for Siemens, machine manufacturer and end user
- 4 Keyswitch positions for end user

Protection level	Locked by:	Users
0	Password	Siemens
1	Password	Machine manufacturer: Development
2	Password	Machine manufacturer: Startup engineer
3	Password	End user: Servicing
4	Keylock switch position 3	End user:Programmers, machine-setters
5	Keylock switch position 2	End user: Skilled operator without programming knowledge
6	Keylock switch position 1	End user: Trained operator without programming knowledge
7	Keylock switch position 0	End user: Semi-skilled operator

For further information, see Section 4.3.1.

HMI Advanced:
The access protection is defined in file c:\user\paramtm.ini. It must be entered after vocabulary word [ACCESSLEVEL].

HMI Embedded:
Protection is set via display machine data.

Examples of functions that can be disabled:

- Load
- Unload
- Magazine list, tool list display

- Tool cabinet, tool catalog
- Loading the magazine configuration

2.10 Openess in HMI

OA / OEM package

The OEM / OA package for HMI Advanced can be used to expand operating masks and tool-management functionalities.

HMI programming package / Open Architecture. Please refer to the most recent NC 60 Ordering Catalog for the current status.

OPI variables and PI services are available to expand the functionality. The special functions are linked by means of the OEM softkeys.

The OPI variables are described in OPI_GR.HLP/OPI_UK.HLP in the directory MM2\HLP.
The Help file OPI_GR.HLP/OPI_UK.HLP is shipped as part of the OEM package.

For more information, please refer to Section 5.12.5 in this description.

Description of functions 3

In this chapter, reference is made to variables, alarms and machine data. A detailed description of these features can be found in the following chapters:

Chapter 5: Programming
Chapter 8: Machine data
Chapter 9: Signal description, PLC interfaces
Chapter 10: Alarms

3.1 Magazines

The position of a tool is shown by a magazine identifier and a location identifier. Magazines have an identifier and a number, magazine locations only a number. In a real magazine (chain, turret, etc.), the position of the tool is identified by the magazine number and the location within the magazine assigned during start-up.

3.1.1 Buffer

Buffers are located in the second internal magazine. The buffer includes the spindle, toolholder, gripper, loader and transfer location. The buffers are located at magazine number 9998. Each buffer element is assigned a unique location. Any location numbers may be assigned. It is recommended that all spindles and toolholder be numbered in ascending order starting at number 1. The assignment to real magazines or of spindles/toolholders to other buffers is made during start-up ($TC_MDP2, $TC_MLSR).

Example: Assigning the locations in the buffer magazine

No.	Name	Type	Index	Assignment to spindles	Distances to magazine
1	Spindle_1	Spindle	1		0
2	Gripper_1	Gripper	1		0
3	Gripper_2	Gripper	2		0
4	Loader_1	Loader	1		0
5	Loader_2	Loader	2		0
6	Transfer_1	Transfer location	1		0

3.1.2 Loading magazine

The loading magazine is the 1st internal magazine and is assigned magazine number 9999. The loading magazine contains the loading points.
A distinction is made between

- loading points and
- loading stations

Loading points are provided for loading and unloading tools. The allocation of locations is fixed, all other locations can be assigned freely. In the case of fixed assignment, location 1 in the loading magazine is used.

Location 1 is reserved for loading/unloading to all spindles/toolholders. All positioning tasks for relocation actions to any locations (not loading points) are still handled via the 1st location. The tasks stated, which refer to a particular magazine location, are output at the interface of the loading point. The loading points are assigned to magazines during start-up ($TC_MDP1). A loading point is an open entry to the magazine where a tool can be **manually** put into and taken directly from the magazine.

A loading station is viewed as an "external magazine location" which a gripper, for example, can access to transfer a tool to the magazine during loading.

3.1.3 Box-type and chain magazines

The setting in MD 22550 (TOOL_CHANGE_MODE) must always be 1 for these types of magazine.

Chain and box-type magazines do not as a rule have any additional buffer available for transportation between magazine and spindle. These additional buffers can contain tools temporarily.

Commands are distributed in the PLC by FC 6. In this case, DB 72 acts as the user interface. There is a separate interface area for each spindle in the interface. A new command from NCK is only then entered in the interface one the previous command has been acknowledged with status values less than 100 (nowadays 1 ..7) by the FC 8.

1. The programming function T = identifier or T = location is implemented in the PLC in data block DB 72. Bit “Prepare tool” is activated in the associated interface.
2. Programming function M06 is also implemented in DB 72. In this instance, bit “Change tool” is set in the activated interface. The bit “Prepare tool” from an earlier T command is not reset here. If the bit “Prepare tool” shall no longer be set for the M06, then it is the task of the user program to reset this bit as part of acknowledging the last T command.
3. Programming functions T and M06 in the same block set the “Prepare tool” and “Change tool” bits simultaneously in the activated DB 72 interface.

Exceptional cases which are imaged in the PLC identically to “3.” above are as follows:

- Initiation of a tool change after block search (last accumulated tool change for the active tool)
- Trigger tool change for Init. blocks

NOTICE !
In these exceptional cases, the subroutine (macro, cycle) in which M06 is normally programmed is not executed.

Examples for machine tools with chain and box-type magazines

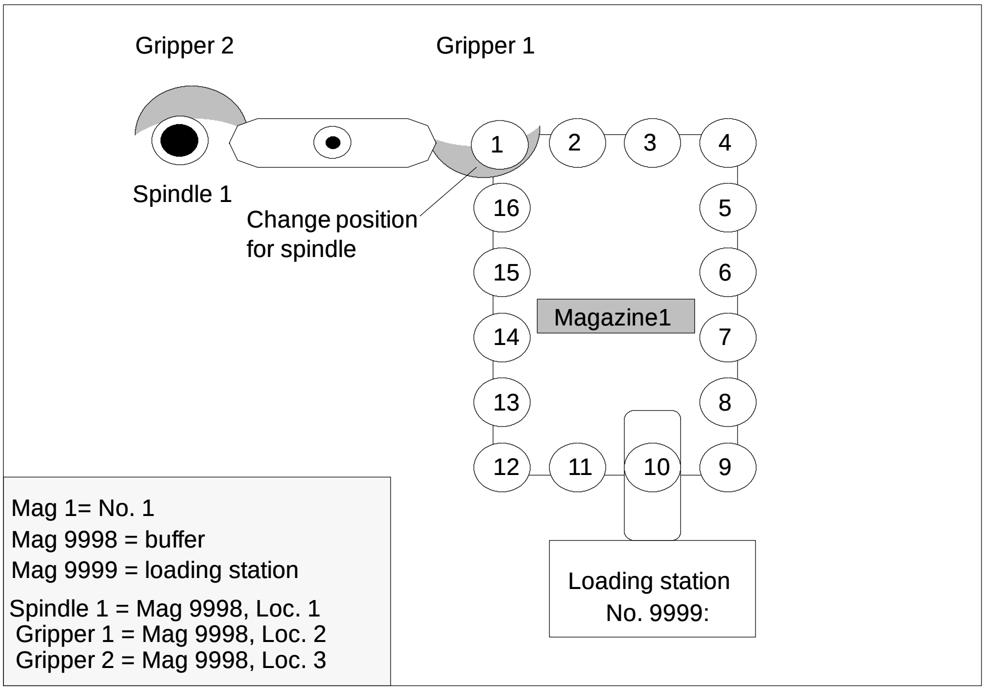


Bild 3-1 Machine tool with chain magazine

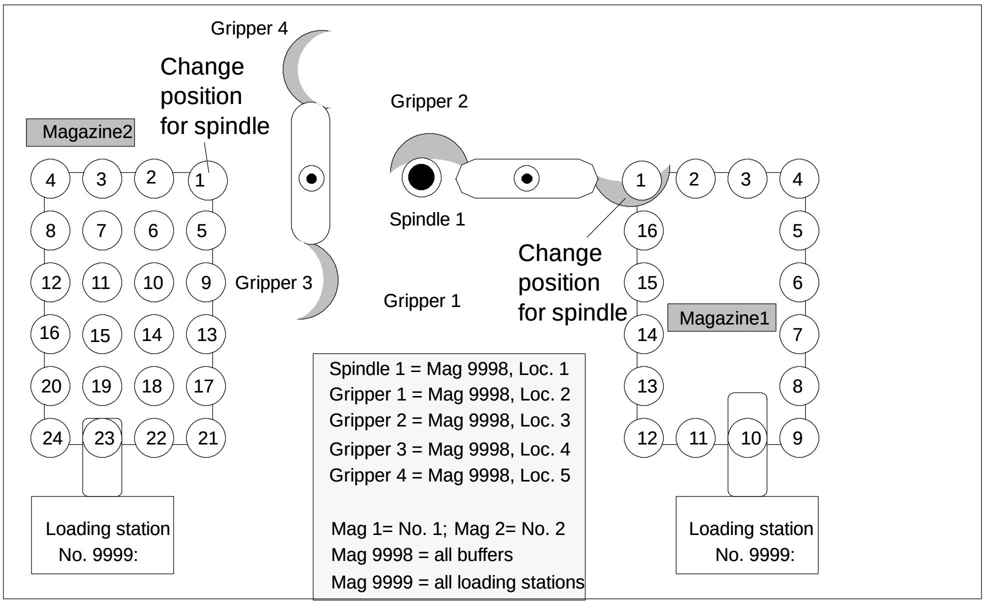


Bild 3-2 Machine tool with chain and box-type magazine

The magazine zero point is defined by $TC_MDP2 (with value assignment of 0). The change position (spindle) is normally assigned on this basis.

3.1.4 Circular magazine

The setting in MD 22550 (TOOL_CHANGE_MODE) is normally 0.

Circular magazines do not have any additional buffer with which tools can be transported from the magazine to the spindle. The tools on circular magazines are not physically transported to the spindle, but are moved into a defined position through rotation of the turret so that machining can take place with one particular tool. The tool is transported to the spindle or holder only in the software.

If TOOL_CHANGE_MODE is set to 1 for a turret, then the description above for chain and box magazines applies too.

The description below applies when TOOL_CHANGE_MODE = 0.

Programming command T = identifier initiates the tool change. T = location can be programmed as an alternative. When T = location, no tool need actually be stored in the location.

The commands is distributed in the PLC by FC 6. In this case, DB 73 is the user interface. There is a separate interface area for each turret. The turret numbers are assigned successively in ascending sequence according to magazine numbers during start-up. The permissible magazine range is 1 ... max. number of real magazines. A new command from NCK is then entered in the interface once the previous command has been acknowledged by FC 7 (alternatively by FC 8 as well).

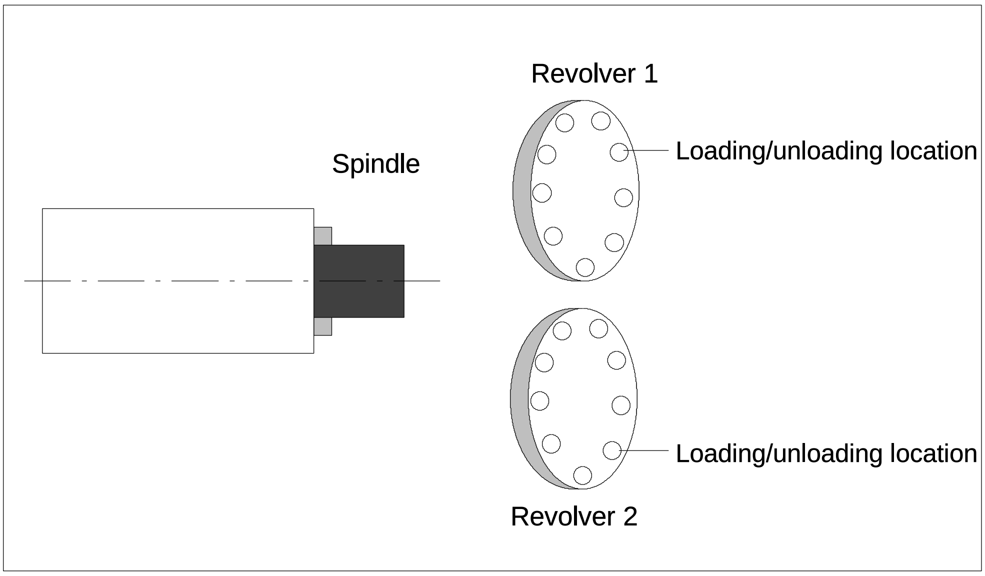


Bild 3-3 Double-slide turning machine with direct loading/unloading point in the turret

3.1.5 Other magazine types

In practice, there are other types of magazines in addition to the ones listed above. These are e.g. disk-type, washer, pick-up, rack, cage magazines (and many more). Such types must be mapped to the three types of magazines supported by the tool management.

3.1.6 Wear group

Locations in a magazine are linked to form an area referred to as the “wear group”. In this way, location groups of a magazine can be activated for particular machining operations.

A wear group number is assigned to each of these locations and the magazine is thereby divided into several different areas. Only tools from one of the areas are then used for a specific machining operation.

The wear group number for each magazine location is defined via system variable **$TC_MPP5[m,p]** (m: Magazine number, p: Location number).

Values in the range of – 32000 ... + 32000 can be assigned.

Values >0:
The specified number is assigned to the location (e.g. **$TC_MPP5[1,3] = 2** assigns the third location of magazine 1 to wear group number 2).

Value = 0:
The location is not assigned to a wear group, as a result the magazine locations are not generally included in tool searches.
If the parameter is set to 0, the data will be fully compatible with magazine data generated in older NCK software versions.

Values <0:
The wear group whose number corresponds to the absolute value of this number is disabled (e.g. **$TC_MPP5[1,3] = – 2** disables wear group number 2 of the magazine with number 1).
This applies even if there is only one disabled location in the wear group.

Notice

Wear groups are only available for real magazines. The definitions for $TC_MPP5 do not affect the status of tools.

Activate wear group

System variable $TC_MAP9 defines which wear group (magazine area) is active. To change the active wear group, the corresponding number is set in this system variable, thereby defining which wear group will be used to start the machining operation.

The default setting is 0. Therefore this is compatible with magazine data of earlier NCK software versions.

The wear group can also set active by a tool change or by the user via NC commands/OPI.

Disable wear group

If there is no longer any spare tool at the location of an active wear group, then the system switches to the next wear group and the former wear group is disabled.

Machining is continued by activating the next group and searching for a suitable replacement tool.

The wear group is also disabled if one of the locations has been disabled via system variable $TC_MPP5 (negative value).

Activate (internally)

Bit 0 of system variable **$TC_MAMP3** can be set to determine how internal activation of a wear group will affect the status of the associated tools.

Value 0:
The tool status is not changed (preset).

Value 1:
When activated, one tool from each tool group included is set to “active”. Tools already set earlier as active are not reset.

Disable (internally)

Bit 1 of system variable $TC_MAMP3 can be set to determine how internal deactivation of a wear group will affect the status of the associated tools.

Value 0:
The tool status is not changed (preset).

Value 1:
When a wear group is disabled all active tools are reset.

Notice

For information about tool searches in wear groups, see Section 3.4.5.

3.1.7 Background magazine

Background magazines are not directly supported by the tool management. However, functions for background magazines can be activated by setting the system variable selectively. System variable $TC_7 MAMP2 – bit 7 can be used to set whether the tool search begins in the magazine last used for tool replacement (bit 7 = 0) or whether the search is carried out in the order defined by “Spindle to magazine” (bit 7 = 1).
This system variable is allocated during magazine configuration (via start-up at the HMI) and saved as an INI file; $TC_MAMP2 – bit 7 is always preset to 0. It is for these reasons that the value for $TC_MAMP2 must either be changed in the INI file (before loading the magazine configuration) or be overwritten by the part program:
$TC_MAMP2=385 (bits 0, 7 and 8 set).

The assignment of “spindle to magazine” is set via system variable $TC_MDP2[n,m]; the order corresponds to the order in which this variable is written. This is pre-assigned as well by the magazine configuration:

Example for 4 magazines and one spindle:

$TC_MDP2[1,1]=0
$TC_MDP2[2,1]=0
$TC_MDP2[3,1]=0
$TC_MDP2[4,1]=0
... this assigns the first buffer (spindle) to magazines 1 to 4; a tool search would therefore start in magazine 1, followed by magazine 2, etc. up to magazine 4.

You can modify this search order by setting this system variable as follows:

1. Delete assignment:
   $TC_MDP2[1,0]=0
   $TC_MDP2[2,0]=0
   $TC_MDP2[3,0]=0
   $TC_MDP2[4,0]=0
2. Re-assign in different order:
   $TC_MDP2[2,1]=0
   $TC_MDP2[3,1]=0
   $TC_MDP2[4,1]=0
   $TC_MDP2[1,1]=0
   ... resulting in the search order Magazine 2, 3, 4, 1

The trigger criterion for changing the order of assignment can be the information in the change cycle that the new tool was found in another magazine. This can be read in the program $A_TOOLMN[t], whereby “t” is the internal T number of the tool. The new tool is obtained via GETSELT. You must remember the previous foreground magazine.

3.1.8 Consider adjacent location

Consider adjacent location is used for oversized tools. When searching for empty locations (loading, tool change) the bits 4... 11 are evaluated in magazine location parameter $TC_MPP4 (half location occupied/reserved). As this function requires additional memory space, the default setting is 0.

To active the function, set the following parameters:
$MN_TOOL_MANAGEMENT_MASK bit 3=1
$MC_TOOL_MANAGEMENT_MASK bit 3=1
In addition, for every magazine location that is to be considered for the adjacent location, parameter $TC_MPP3=1 must be set.

Two new functions are available with active consider adjacent location (from SW 7.2).

Overlap disabled magazine locations

This function is activated by setting the magazine location parameter $TC_MPP4 bit 13=1. If a location is disabled, it can now be "overlapped" by an oversized tool. This means the consider adjacent location function ignores the disabled state of a magazine location.

Example:

Chain magazine, location 12 is disabled (e.g. tool reception is defective).
An oversized tool (size 2/2/1/1) is loaded or is positioned in the spindle.
Previously it could only be inserted in location 10 or 14.
Now it can also be inserted in locations 11 or 13.

The following default settings are available:
As soon as a location is disabled, "Overlapping active" is automatically set (or reset as soon as the location is no longer disabled).
This setting is made in machine data $MN_TOOL_DEFAULT_DATA_MASK bit 4=1.

Overlapping magazine edge locations

This function is activated by setting the magazine description parameter $TC_MAP4 bits 8 to 11.

The following definition applies:
(Definition: smallest magazine location No. is top left, largest magazine location No. is bottom right).

Bit 8 **left** edge location may be covered
Bit 9 **right** edge location may be covered
Bit 10 edge location **top** may be covered
Bit 11 edge location **bottom** may be covered

The default setting for these bits is 0.

Example:

Flat magazine
Due to the mechanical conditions, oversized tools can cover the edge at the top and on the right.
You need to set:
$TC_MAP4[Magazine No] bit 9=1
$TC_MAP4[Magazine No] bit 10=1

The function can also be used with chain magazines and turrets. The sequence of evaluation is bit 10 and bit 11 (top, bottom).

3.2 Tool change box-type, chain, circular magazines

As a rule, programming the tool change differs for box-type and chain magazines from programming for circular magazines.

The tool change differences for these different magazine types are set for each channel via machine data MC_TOOL_CHANGE_MODE.

3.2.1 Prepare a tool change

Different methods of tool change can be programmed as a function of machine data (MD 22550) $MC_TOOL_CHANGE_MODE:

$MC_TOOL_CHANGE_MODE=0

```
T="Tool identifier"        ; Tool preparation and tool change with an NC
                           ; command (= within an NC block)
                           ;
                           ; NCK sends a command to the PLC
```

If an error is detected during tool preparation, then machining is stopped when the block T = identifier is read-in.

After correction and NC Start, the block with T = identifier is interpreted again and program processing is continued.

$MC_TOOL_CHANGE_MODE=1

- Within an NC block

```
T="Tool identifier" M06    ; Tool preparation and tool change. This pro-
                           ; gramming line results in a command to the
                           ; PLC
```

Programming tool preparation and tool change in one block (**T**= "Tool identifier" **M06**), corresponds to setting TOOL_CHANGE_MODE = 0.

- distributed over two NC blocks

```
T="Tool identifier"        ; Tool preparation
                           ; NCK sends a command to the PLC
M06                        ; Tool change (the number of the M code is
                           ; settable),
                           ; NCK sends a command to the PLC
```

Tool preparation and tool change are typically programmed in different blocks. Two commands are transferred to the PLC.

An alarm is triggered if an error occurs in T= “tool identifier”. If the MD TOOL_CHANGE_ERROR_MODE (MD 22562) is set accordingly, the alarm is delayed until the associated tool change command M06 is interpreted in the program run. Only then is the alarm output. The operator can make corrections in this block.

Notice

A D compensation is activated by a tool change. If the D command is not programmed in the block containing the tool change command, the tool compensation set in MD 20270: CUTTING_EDGE_DEFAULT is activated. If the value of the variable is –1 or >0 (selection of a specific compensation), the alarm 17181:
“D number for the tool does not exist in the NCK” may occur.

If the value is 0 (compensation deselected) or –2 (old compensation retained), there is no problem when determining the compensation.

Empty spindle

Program commands T0 and M06 remove the tool from the spindle and return it to the magazine. The spindle is then empty.

Possible problems in programming T/M06

$MC_TOOL_CHANGE_MODE=0; tool change with T address
The part program is executed through to the block T = “identifier”. The following problems can occur and are handled in the manner described:

- The tool data record is in the NCK but not assigned to a magazine location. The tool must be reloaded mechanically, if necessary, e.g. directly onto the spindle. The assignment of the tool to the magazine location/the spindle takes place e.g. with the function “Overstore”; $TC_MPP6[m,p] = T no., or by the HMI operation “Load (onto spindle)”.
- The tool data record is not in the NCK:
  Set up data block in the NCK, e.g. by HMI operation.
- Programming error in part program:
  Correct discrepant NC block in the part program.
- Alarm 22067:
  The desired tool change is not possible. The specified tool group does not contain a “ready to use” replacement tool which could be loaded. The tool monitoring function may have set all potentially suitable tools to the “disabled” status.

The Start pushbutton is operated once the operation has been completed. The NC block T “tool identifier” is interpreted again and program processing is continued provided operator intervention was correct. If not, the alarm will be generated again.

3.2.2 General tool change sequence

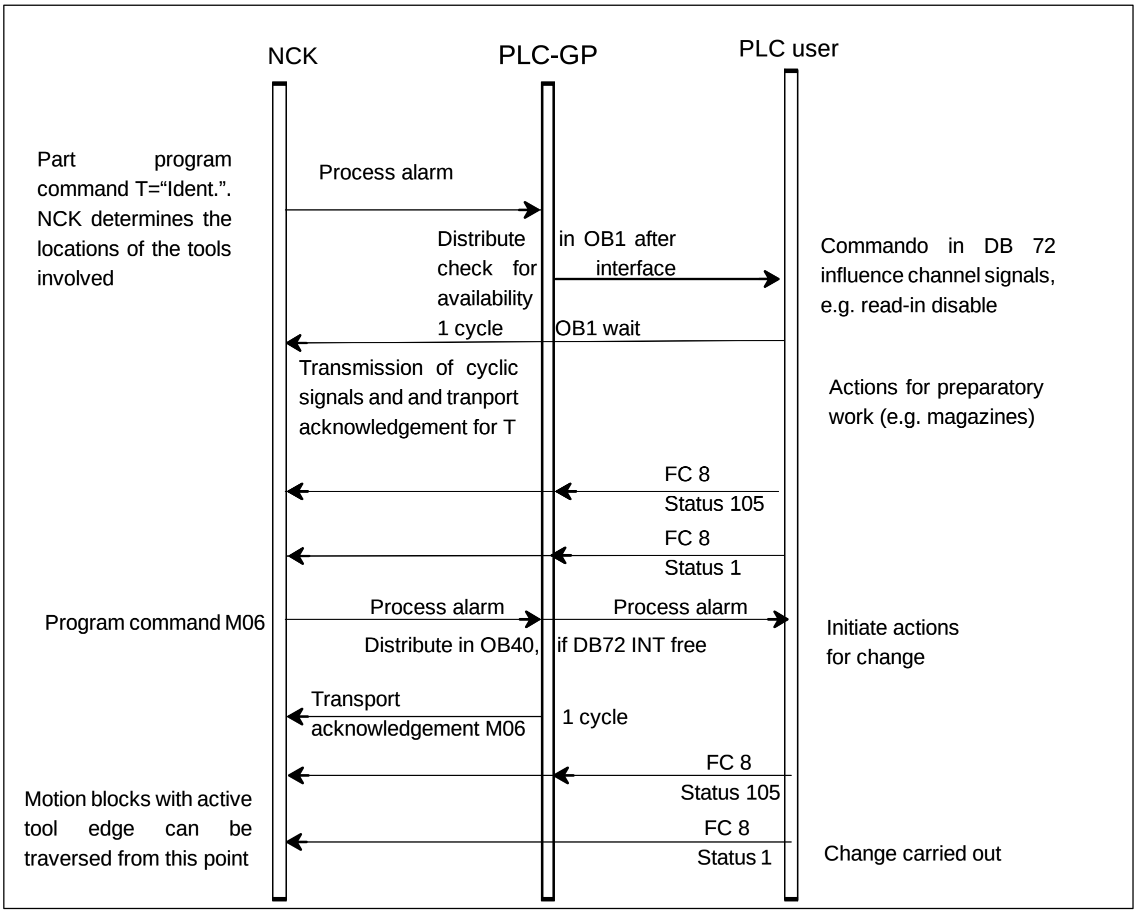


Bild 3-4 Preparing and changing a tool

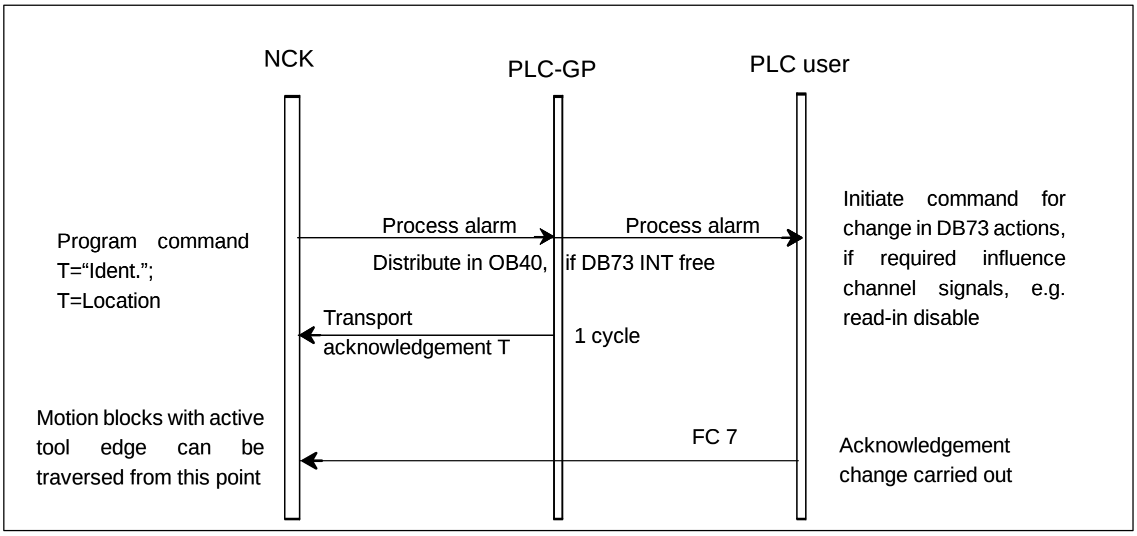


Bild 3-5 Tool changing with T command

The prompt for changing the tool is issued by the part program via T command or M command.

1. The tool-management function in NCK searches in accordance with the tool searching strategy and the requirements of the T call for a tool suitable for use (preparation) and, at the same time, searches for an empty location for the tool to be exchanged.
2. The calculated data are make available in DB 72/73. The user program must react by making a new tool available.
3. If machine data MD 22550: TOOL_CHANGE_MODE is set to 1, the PLC executes the tool change with the "M06 command" in the part program and signals the completion of the change operation.
   If the machine data is set to 0, the tool data is changed and the desired compensation become active when T or D are programmed. The PLC has the option of applying its own tool change strategy. It can choose its own empty location for storing the old tool.

Example

If, for example, in a tool change with a dual gripper, the old spindle tool is to be replaced in the magazine as "quickly" as possible, the PLC must check whether the location is suitable to accommodate the old spindle tool in terms of type and adjacent locations. Tool management is then to be informed of the change operation by the PLC (FC 8 block).

The new empty location search strategy "Replace new tool for old" is also available. Tool management thereby checks whether its is possible to replace the old tool with the new tool at the location of the new tool (1:1 replacement).

Notice

The tool change in an NCK-internal operation that is executed as an interaction with the PLC. The HMI only has the task of displaying data and facilitating data input.

Spindle and toolholder

Tool management can also be used for machines that have no spindle (e.g. punch presses or turrets). In this case the term “spindle” is replaced by “toolholder”; define the setting in MD 20124 TOOL_MANAGEMENT_TOOLHOLDER. If the MD setting is >0, the spindle numbers $TC_MPP5 are interpreted as toolholder numbers.

Fixed location coding

If fixed location coding is selected for a tool, the tool will always be returned to the same location when it is replaced.

Variable location coding

Tools defined with variable location coding can be returned to any location for the appropriate tool size and location type in the magazine.

Automatic tool return to real magazine

1. An automatic tool return is initiated by the TM only if the tool is transported via several stations (status 105) after a T preparation command from the PLC and the T preparation command is finally acknowledged positively with status 1. The return of a preselected tool from the buffer can be suppressed by setting MD 20310: TOOL_MANAGEMENT_MASK, Bit 15 = 1.
2. If a tool change is interrupted because the control is switched off but the tool is already located in a buffer location (gripper), the next tool change must either return the tool in the buffer to the spindle or to the real magazine.
3. If several tools are located in the buffer the spindle tool is considered first. If there is no tool on the spindle, the order for return is in accordance with system variable $TC_MLSR.

Example for the time sequence of a tool change

The following example shows a typical cut-to-cut sequence of operations for a tool change with a tool changer and a fixed absolute tool change point on a milling machine.

Machining program

```
N970   G0   X=   Y=   Z=   LF   ;Retract from the contour
N980   T1   LF                  ;Tool preselection
N990   W_WECHSEL   LF           ;Subroutine call without parameters
N1000  G90  G0  X=  Y=  Z=  M3  S1000  LF  ;Machining resumed
```

Subroutine for tool change

```
PROC W_WECHSEL LF
N10    SPOSA=   S0   LF                    ;Spindle positioning
N20    G75  FP=2  X1=0  Y1=0  Z1=0;        ;Approach tool change point
N30    M06   LF                            ;Change tool
N40    M17   LF
```

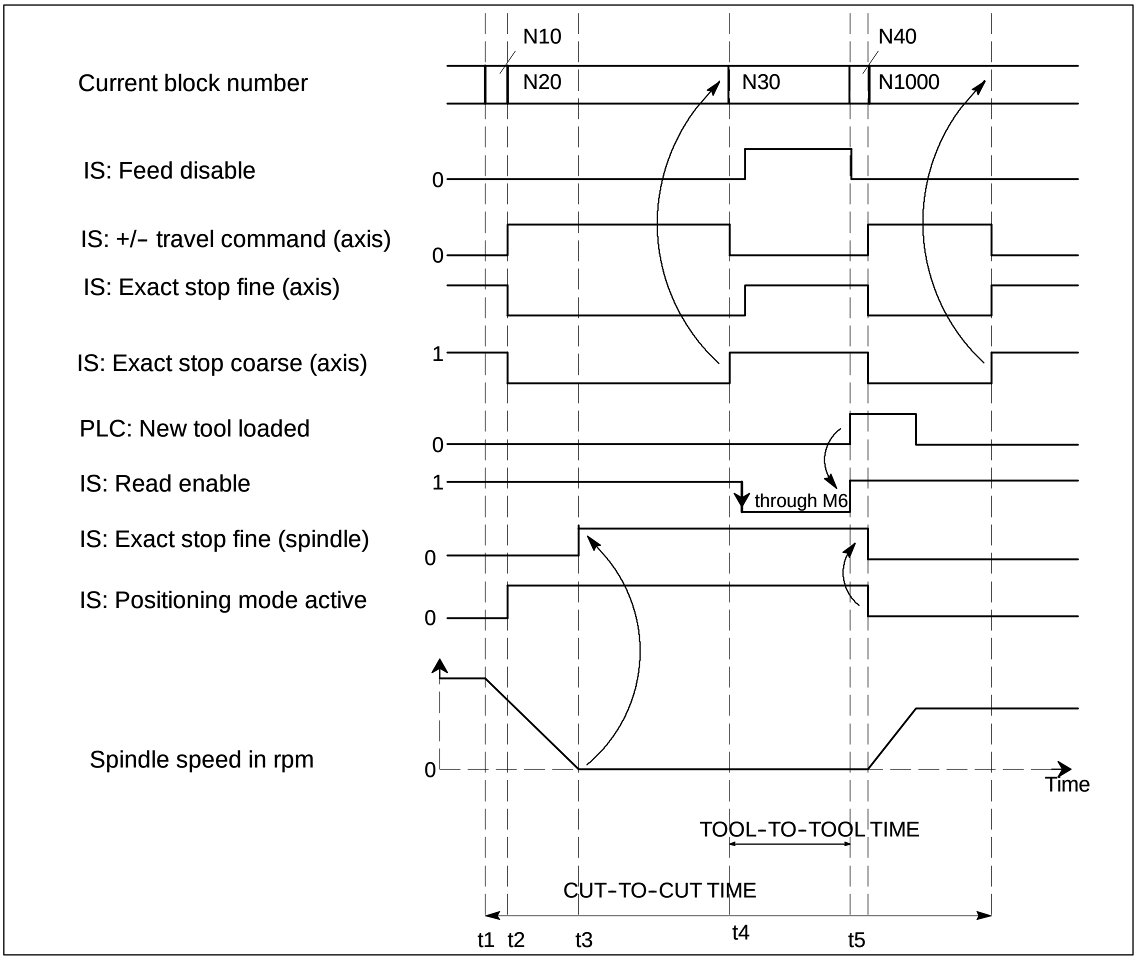


Bild 3-6 Chronological sequence of tool change

t1	Axes at standstill Spindle rotates Start of tool change cycle in N10.
t2	Traverse axis with G75 in N20 to tool change point
t3	Spindle reaches programmed position from block N10
t4	Axes reach exact stop coarse from N20; N30 now starts: M06 removes the previous tool from the spindle and loads and clamps the new tool.
t5	Tool changer swivels back to original position.

Then, in N1000 of the calling main program,

- new tool offset can be selected,
- the axes can be returned to the contour, or
- the spindle can be accelerated.

3.2.3 Select a tool and the cutting edge

Notice

T number and the M function are no longer transferred to the PLC as an auxiliary function if TOOLMAN is activated.
Numbers are valid tool names as well, e.g. “3” instead of T = “3” can be programmed more simply as T3.
There must be a tool with the T number as the identifier available when working with the T number.

Example:
If you want to call a tool using T3, the tool must have the name “3”. A tool CANNOT be called with the internal T number managed by NCK only.

Select/deselect tool compensation on Reset

The following machine data can be used to control the behavior on RESET:

- MD 20310: TOOL_MANAGEMENT_MASK bit 14
- MD 20122: TOOL_RESET_NAME
- MD 20110: RESET_MODE_MASK
- MD 20130: CUTTING_EDGE_RESET_VALUE
- MD 20132: SUMCORR_RESET_VALUE

You can determine whether to:

- deselect the active tool
- keep the active tool selected
- of a particular tool is selected (according to MD 20122 TOOL_RESET_NAME)

If, in terms of its data, a new tool is selected that is not yet on the master spindle or the master toolholder (or main spindle, main toolholder), then a tool change is executed a tool change for a reset or the end of the program. With this type of tool change (in a similar manner to block searches), the PLC is not capable of influencing the selection of the tool.

Select a tool at start of program

With machine data

- MD 20310: TOOL_MANAGEMENT_MASK bit 14
- MD 20122: TOOL_RESET_NAME
- **MD 20112:** START_MODE_MASK
- MD 20130: CUTTING_EDGE_RESET_VALUE

can be set to define whether

- the tool on the main spindle or the main toolholder is selected again or remains selected
- or select a specific tool (as defined in MD 20122: TOOL_RESET_NAME).

If a new tool is selected which in the data is not yet specified as being on the spindle, a tool change is performed when the program is started. In this type of tool change the PLC cannot influence the selection of the tool, just as for block search.

Tool rejection by the PLC

On a block search, selection on reset or program start, the tool is selected during preprocessing. In this case the PLC is not allowed to reject the tool.

Notice

If bit 4 of machine data MD 20310: TOOL_MANAGEMENT_MASK set, then PLC usually has the possibility to again request preparation for a tool change, yet this time with different parameters, i.e. to reject the tool.

Communication between PLC and tool management

The communication between PLC and NCK during a tool change is via the VDI interface. Tool change is triggered by the tool management in the NCK. The TM outputs commands to the PLC which acknowledges them either positively or negatively depending on the situation (see also Section 2.3).

Command acknowledgement

When acknowledging a command from the NCK, the PLC can change the parameters of this command in the acknowledgement data.

The following sequence is implemented to allow acknowledged commands from the PLC to be assigned in the NCK:

- The T number in the command is used to determine the tool in the NCK.
- The data of the current tool location are obtained from the tool.
- This current tool location is checked against the address specified in the command.
- If the data does not correspond, the data is corrected in the acknowledged command and in the original command residing in the NCK.
- Acknowledgement of the command in the NCK is continued.

Notice

If the tool to be changed is transported from the magazine to the toolholder in multiple individual steps, the PLC acknowledgement number 105 applies.

With the PI command _N_TMMVTL a tool in status “being changed” cannot be moved.

The following applies for loading, unloading, reloading and positioning: The PLC must not change the target positions specified by the NCK for the NewTool as they have to be identical with those in the NCK.

Example 1

The data printed in bold font (can be changed) in the PLC acknowledgement and NCK indicate that the NewTool is not longer present at the From location specified in the command (NewTool: from M: 00002 P: 00001). It was, for example, moved to gripper location 9998/3 after output of the command “Prepare tool change” by an asynchronous PLC motion command. Only then is the tool preparation command acknowledged by the PLC. NCK checks the tool data and compares them to the data in the command (underlined) and corrects the command data in the NCK after internal command assignment, thus allowing subsequent acknowledgements to proceed with the valid data.

T00001 N:N10 CMD:00002

NewTool:
from M: <u>00002</u> P: <u>00001</u> to M: **09998** P: **00003** TNo: 00001 Spindle : 00001

OldTool:
from M: **00000** P: **00000** to M: **00000** P: **00000**

After automatic correction of the command parameter “NewTool: from” in NCK to NewTool: from M: 09998 P: 00003 to M: 09998 P: 00003
the original command is acknowledged in the NCK with this correct data.

Example 2

1. NCK tool change command to PLC
2. An asynchronous tool motion command triggered by one of the PLC commands 8 or 9 (tool was transported), i.e. direct data manipulation plus associated mechanical tool movement.
3. Acknowledgement of the tool change command

Tool preparation command NCK –> PLC (bring TNo.=1 from gripper= 9998/4 to spindle 9998/1) is calculated and output to the PLC:

T00001 N:N10 CMD:00002
NewTool:
from M: 09998 P: 00004 to M: 09998 P: 00001 TNo: 00001 spindle 00001
OldTool:
from M: 09998 P: 00002 to M: 00003 P: 00004

Asynchronous tool motion command 9 from PLC (bring TNo.=1 from magazine location 9998/4 to 9998/3):

T00002 N: ACK: 00009 un: 00001
NewTool:
from M: 09998 P: 00004 to M: 09998 P: 00003
OldTool:
from M: 00000 P: 00000 to M: 00000 P: 00000

The tool changes the location data in the NCK as well as the mechanical location. Tool TNo.=1 is now in location 9998/3.

PLC acknowledges the tool preparation command as follows:

T00003 N: ACK: 00002 un: 00001
NewTool:
from M: 09998 P: 00004 to M: 09998 P: 00004
OldTool:
from M: 09998 P: 00002 to M: 09998 P: 00002

As the tool with TNo.=1 is actually located in position 9998/3, the NCK first assigns the command then corrects the acknowledgement data within the NCK:

T00003 N: ACK: 00002 un: 00001
NewTool:
from M: 09998 P: 00003 to M: 09998 P: 00001
On the machine this corresponds to the real tool transport from the gripper (9998/3) to the spindle (9998/1).

Example 3

If the command for tool change is removed by the asynchronous tool motion command “unload this tool that was just loaded for change”, alarms 6405 and 6442 are displayed when the change command from the NCK is acknowledged.

Selection of the tool offset

Once the tool has been changed the following options are available for selecting the tool compensation:

1. The compensation number is programmed in the same block as the command for the tool change.
2. It is specified by the setting in MD 20270: CUTTING_EDGE_DEFAULT

= 0	The compensation is deselected (= D0).
> 0	Number of the compensation, which is selected in accordance with M06
= –1	The compensation number of the old tool remains valid and is selected for the new tool after M06.
= –2	The last selected compensation remains valid until a D number is programmed.

Notice

Detailed information on cutting and compensation numbers can be found in /FB1/ W1 – tool compensation.

3.2.4 Predecoding (preprocessing) and block execution (main run)

Sequence

The cutting edge geometry cannot be calculated until the tool management knows the tool that is actually to be used. Only the identifier is stated in the part program for tool change. Generally, the tool with the status “active” is then used. If this is then disabled, then one of the other spare tools is used – the Spare tool. The precoding delays selection of the new compensations until it is clear which tool is to be used. Only then can precalculation of the blocks be restarted.

Tool change must have been completed before the path can be traversed with the tool compensation of the new tool.

The block is split if the preprocessing run detects that a new edge of a new tool has been selected for the first time and tool preparation has been initiated, but not yet completed.

The following synchronization points exist between predecoding and block execution:

Example:

Programmed NC block:

```
N1D1 M06 Txx X100 Y100
```

Sequential blocks:

```
N1 Txx M06 end of block
N2 D1 X100 Y100
```

1.	Interpreter detects an compensation selection (D number)
2.	It determines that a tool change was previously programmed which has not yet resulted in selection of a tool.
3.	Interpreter carries out “block splitting”.
4.	Output of block N1: Block 1 receives a request from the execution blocks to output their collective blocks, and also if programmed, M06, T numbers, ...
5.	Output of block N2: Block 2 receives the rest, in particular all travel information and any D numbers if programmed.
6.	Tool management stops execution of the block during preprocessing until it is clear which tool is to be used.
7.	After receiving the tool preparation acknowledgement, execution of block 2 is continued, or first the new T number is entered in the block and is used to calcualte the contour again.

Tool change at the main spindle or master toolholder

The main run waits in synchronism with tool change block for transport acknowledgement.

1. Main run waits in synchronism with tool change block for end of acknowledgement (if bit 5 or bit 6 of MD 20310: TOOL_MANAGEMENT_MASK is set) or
2. After a tool change in the main run, the NCK automatically performs synchronization with the end of the tool change in the first block in which an edge of the new tool is selected.

Notice

The transport acknowledgement is an internal acknowledgement of an NCK command. It indicates to the NCK that the output command was accepted. When a new command is output to the PLC, the NCK waits for the acknowledgement of the previous command.

Tool already in spindle

If the programmed tool is already in the spindle, by default no command is sent to the PLC (The response can be defined by the MD setting.)

Tool change at the secondary spindle or secondary toolholder

1. Main run does not wait. There is no synchronization.
2. Main run waits in synchronism with tool change block for transport acknowledgement
3. Main run waits in synchronism with tool change block for end of acknowledgement.

Tool change preparation in a main spindle

1. Tool management decides during the main run which tool is to be used (the active tool or a replacement tool). Until then, the preprocessor waits at the point in the program at which the compensation values of the new tool are to be considered for the first time.
2. The PLC can also decide which tool is to be used. In this case, the PLC can reject the proposed tool with a negative acknowledgement. If rejected by the PLC, the NCK selects a new, different tool (only if MD 20300: MC_TOOL_MANAGEMENT_MASK bit 5 = 1, see also FC 8 description, Section 4.2).
3. Even if the function “GETSELT(...,x)” is programmed, the preprocessor again has to wait until a decision has been made as to which tool is to be used.

Prepare to change tool in a secondary spindle

1. The main run does not wait. There is no synchronization.

Notice

During a synchronization operation where the new compensation is used or allowed for by the preprocessor, “block splitting” must be performed. This ensures that a preprogrammed tool change T or M06 is actually performed and not collected in run blocks.

Unlike the STOPRE command, the preprocessor does not necessarily wait until all blocks have been processed, but only waits if tool selection has not taken place by the appropriate time. The appropriate time is when programming new compensations after tool change or when programming GETSELT.

3.2.5 Traverse axes while tool is being changed

After the tool change command M06 the axes can continue travel without having to wait for the tool change acknowledgement and, e.g., execute traversing blocks without tool compensation. Travel only stops in a block with an compensation selected (D no.) until the tool change is signaled by the PLC.

Requirement: MD 20270: CUTTING_EDGE_DEFAULT= 0 or = –2

Example: Traversing blocks between tool change and cutting edge selection

```
N10  T="Drill18"        ; Tool change preparation
N15  M06                ; Tool change
N20  D0                 ; Compensation deselection
N25  G00 X100 Z200      ; Traverse machine axes
N30  Y150 M79           ; Traverse machine axes
N35  G01 D1 X10         ; Activating the tool compensation.
                        ; Check whether tool has been changed. preproces-
                        ; sing stop until tool change preparations are com-
                        ; pleted.
                          Main run waits until tool change is acknowledged
                          from PLC
```

The preprocessing stop is maintained until the tool change preparations have been completed. The main run waits at N35 (D1) until the tool change has been executed and acknowledged.

3.2.6 Tool change to the spindle for chain and box-type magazines

Spindle/buffer DB 72

Data block **DB 72** changes tools in the spindle. This data block also prepares the tool change. This data block has an interface for every spindle.

User data is available at each interface (sequence in accordance with the spindle number) as for the loading and unloading points. The data block also contains additional data for the new tool. This data includes location type, sizes, tool status and the T number internally assigned in the NC.

The buffer address of the spindle is contained in DB 72. DBW(n+16) and DBW(n+18) as the destination for the new tool. This position is communicated as the target position of the new tool in parameters “NewToolMag” and “NewToolLoc” when the tool change has been successfully completed. The target position for the old tool (DB72. DBW(n+24) and DBW(n+26)) is transferred to FC 8 in parameters “OldToolMag”, “OldToolLoc” together with parameter “Status = 1” after the change tool command has been executed.

Description of tool exchange in spindle

The tool in location 1, magazine 1 is to be loaded to the spindle (magazine no. 9998, location 1) and the tool in the spindle is to be returned to magazine 1 location 8.

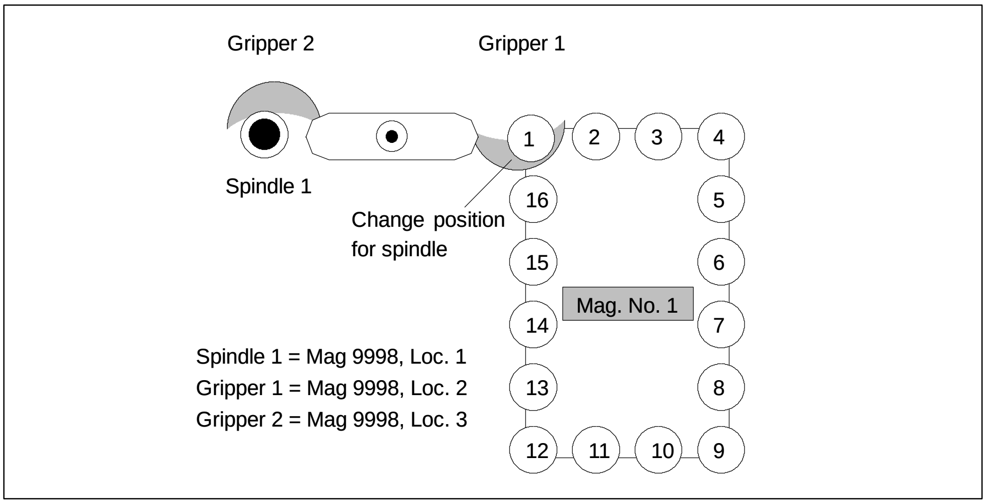


Bild 3-7 Load tool into spindle

The tool change in the spindle is split into two steps (for TOOL_CHANGE_MODE=1):

1. Prepare change:
   Search for new tool and move to the change position
2. Perform change:
   New tool into the spindle and old tool into magazine in correct location

1. Prepare change

Bit 2 is set in DB72.DBB n+0. In preparing for the change, the current positions of the tools are forwarded to FC 8 in the associated parameters once the preparation step has been completed. “Status” = 1 is parameterized in FC 8 at the same time. This means that the “old tool” is still in the spindle and that the “new tool” is either still in the source magazine at the same location or has been placed in a buffer.

The following information is passed to FC 8:

- The new tool is in the change position, but is still located in magazine (NewTool-Mag = 1 and NewToolLoc = 1).
- The old tool still remains in the spindle.
  (OldToolMag = 9998 and OldToolLoc = 1).

FC 8 parameter	Values	Comment
Start		starts job
TaskIdent	2	DB 72 interface
TaskIdentNo	1	No. of active interface
NewToolMag	1	(n+20) Mag No. new tool
NewToolLoc	1	(n+22) Loc. No. new tool
OldToolMag	9998	(n+16) Mag No. target new tool
OldToolLoc	1	(n+18) Loc. No. target new tool
Status	1	Action completed
Ready		Feedback signal from FC 8
Error		Feedback signal from FC 8

2. Change tool

If the preparation command has been correctly acknowledged with status = 1, the “Change” bit DB72.DBB n+0 bit 1 is set with the M06 command in the part program. The user parameters are also transferred again. All other values remain unaffected by the “Change tool” operation.

Two tools are involved in the tool change procedure. The old tool is in the spindle and the new tool is in the magazine. The tool transport is executed in this example with gripper 1 and gripper 2. Any change in the position of the tools must be communicated to the tool management with FC 8. FC 8 must be called twice for this purpose.

FC 8 call with status 105 “Change in progress”

The tool with the dual gripper is pulled from the magazine and the spindle. The old tool is now in gripper 2 at location no. 3 and the new tool in gripper 1 in location no. 2. The following FC 8 call results:

FC 8 parameter	Values	Comment
Start		starts job
TaskIdent	2	DB 72 interface
TaskIdentNo	1	No. of active interface
NewToolMag	9998	(n+16) Mag. No. spindle
NewToolLoc	2	(n+18) Loc. No. new tool New tool now in gripper 1
OldToolMag	9998	Mag. No. old tool
OldToolLoc	3	Loc. No. old tool Old tool now in gripper 2
Status	105	Procedure running
Ready		Feedback signal from FC 8
Error		Feedback signal from FC 8

Notice

The operator uses FC 8 to notify the tool management of the new positions of the exchanged tools.
Tool management knows which is the new (called) tool and which is the old (spindle) tool.
The current positions are also known to the tool management. If these positions change, the tool management is only informed about this through FC 8.

Notice

If T preparation and change signals are present at the same time, the tool call and change command (T and M) are programmed in one block. When FC 8 is called in such a case, only the change and not the selection need be acknowledged.

FC 8 call with status 1 “Tool change complete”

While the gripper is moving the tools, the PLC can read the magazine location for the old tool (from the spindle) from DB72.DBW (n+24) and (n+26) and move the magazine to the change position. This position is location 8 in magazine 1 in this example. The tool change can now be mechanically ended by “inserting” the tools. Tool management shall be informed of this change in tool positions by a FC8 call with status = 1. The new tool is placed in the spindle of magazine No. 9998, location No. 1 and the old tool in magazine No. 1 at location 8.

FC 8 parameter	Values	Comment
Start		starts job
TaskIdent	2	DB 72 interface
TaskIdentNo	1	No. of active interface
NewToolMag	9998	(n+16) Mag. No. spindle
NewToolLoc	1	(n+22) Loc. No. spindle
OldToolMag	1	(n+24) Mag No. old tool
OldToolLoc	8	(n+26) Loc. No. old tool
Status	1	Procedure completed
Ready		Feedback signal from FC 8
Error		Feedback signal from FC 8

If the dual gripper is to place the spindle tool in the magazine location of the new tool, the user must ensure that the magazine location is of the same size and location type as the spindle tool.
Here too, a 1:1 replacement is supported through appropriate setting of the search strategy by the tool management.

If this is the case, the transfer can be performed simultaneously (on the dual gripper in the spindle and in the magazine location at the change position).

FC 8 must be parameterized as follows.

FC 8 parameter	Values	Comment
Start		starts job
TaskIdent	2	DB 72 interface
TaskIdentNo	1	No. of active interface
NewToolMag	9998	(n+16) Mag No. new tool
NewToolLoc	1	(n+18) Loc. No. new tool
OldToolMag	9998	(n+20) Mag No. old tool
OldToolLoc	3	(n+26) Loc. No. old tool
Status	1	Procedure completed

FC 8 parameter	Values	Comment
Ready		Feedback signal from FC 8
Error		Feedback signal from FC 8

3.2.7 Special cases “TO”, empty spindles, multiple T selection

TO: Empty spindle

DB72.DBX(n+0).3 indicates that TO has been programmed. If TO has been programmed to empty the spindle, DBW (n+20), DBW(n+22) – data for new tool – in DB72 are assigned the value “0”.

Parameters NewToolMag and NewToolLoc of FC 8 must then be set to “0”.

This applies to the preparation and to the change procedure.

Spindle is empty

The tool must be changed. This status is indicated by the fact that parameters OldToolMag and OldToolLoc are set to “0”.

In this case, FC 8 parameters OldToolMag and NewToolLoc must be set to “0” for tool preparation and change.

Multiple T selection

It can happen with multiple T selection that the program cannot be aborted by a reset.

The interruption response can be enhanced as follows:

- Cancel the read-in enable to prevent following blocks from being accepted in the main run.
- Then acknowledge with status 3 via FC 8 (the tool command is denied by the PLC).
- When the acknowledgement has been issued, the RESET can be activated for the channel.

3.2.8 Tool change with turret

Turret DB 73

DB 73 is the block used to “change” tools in the turret (i.e. by rotating the turret so that the required tool is in working position). This data block has an interface for every turret. The turrets are numbered using ascending magazine numbers. User data are available at each interface as for the loading and unloading point. The data block also contains additional data for the new tool. These data are amongst others, location type, sizes, tool status and the T number internally assigned in the NC.

Following completion of the tool-change operation, loading the new tool shall be acknowledged by FC 7. To this end, the parameter “ChgdRevNo” receives the turret number of the new tool that has been inserted.

FC 8 parameter	Values	Comment
Start		starts job
ChgdRevNo	1	1. Revolver
Ready		Feedback signal from FC 8
Error		Feedback signal from FC 8

3.2.9 Number of replacement tools

Machine data MD17500: MN_MAXNUM_REPLACEMENT_TOOLS can be set to select the maximum number of replacement tools.

Once the set threshold for the number of replacement tools has been reached it is no longer possible to:

- create a tool with ID (alarm) or
- assign a tool by renaming an already fully assigned group (alarm).

Alarms

For operation via the HMI, alarm 17192 is output as an indication as soon as the defined limit is violated.

If programming via a part program an additional interpreter alarm is triggered (e.g. 14020 if NEWT fails).

Notice

Machine data MD 17500: MAXNUM_REPLACEMENT_TOOLS is limited (up to 600) only by the upper limit value set in machine data MD 18082: MM_NUM_TOOL.

3.2.10 Tool changing errors

If an error is detected by the NCK in the programmed tool preparation (e.g. no tool available, no free position in magazine) program processing is terminated with an alarm.

The operator can assess and rectify various problems without terminating the program.

The following problems can be solved:

- The tool data record is not or not entirely in the NCK.
- The part program contains a programming error.
- No more replacement tools of the tool group in question are available (only applies when tool management is active).
- Alarm 22067 or 22069 is stored. The tool data record has been loaded into the NCK but is not assigned to a magazine location or the magazine of the tool is not accessible to the tool search (only applies when tool management is active). The tool must be reloaded “manually” (e.g. directly onto the spindle).

Notice

The case of “Invalid D number” can occur either if there is an error in the part program or the data block for the D number is not in the NCK.

Programming example

```
N10 ...
N100 T=“Drill”       ; NCK detects an error
N110 ...
N200  M06            ; to the extent that the tool change is not
                     ; explicitly programmed in the same program for
                     ; tool preparation
N210 ...
```

Notice

As a rule **M06** is not programmed at the program level of tool preparation but rather in a subroutine, cycle or macro.

Bit 0 of machine data MD 22562: TOOL_CHANGE_ERROR_MODE determines the block at which the program must stop.

TOOL_CHANGE_ERROR_MODE, bit 0=0:

```
N10 ...
N100 T="drill"        ; NCK detects an error, program stops at this
                      ; block
N110 ...
N200  M06
N210 ...
```

TOOL_CHANGE_ERROR_MODE, bit 0=1:

```
N10 ...
N100 T="drill"        ; NCK detects an error
N110 ...
N200  M06             ; program stops at this block
N210 ...
```

The fault is found during tool preparation yet is ignored by the NCK. The program continues and stops at M06. Tool preparation has been completed at this point in time for a regular program run. In the event of an error, tool preparation with the correct data can be subsequently effected.

The programming error (in block 100 in this example) is corrected by adding the compensation to the tool change block:

```
N200 "T=Drill_1" M06
```

If a tool change (with M06 programming) is realized by means of a subroutine or cycle program, then the error can be rectified by inserting an overstore block (in the example).

3.2.11 Manual tools (retrofitting tools during machining)

Bit 1 in MD 22562: TOOL_CHANGE_ERROR_MODE can be set to select additional tools without magazine assignment during tool changes. The automatically selected tool must be inserted in the machine manually and removed again manually after machining.

Responsibility of the user

The user must ensure that

- the data record of the tool positioned on the spindle is actually in the NCK and
- that he or she places the tool that corresponds to the data record in the NCK on the spindle.

Tools which are loaded manually during machining are referred to as **“manual tools”**.

Notice

The responsibility is on the users themselves to comply with the safety regulations via the PLC program.

Sequence

Internally, the NCK initiates an automatic sequence until the user can perform the tool change with a manual tool. The NCK searches for the selected tool and detects that a suitable tool is not available in the magazine. After determining there is no suitable tool in the tool-holding magazine, the tools are investigated that are not assigned to any magazine. The tool with the status active is selected from these. If there is no active tool, then the tool with the lowest duplo no. is selected.

If no suitable tool is found, then loading a manual tool can take place. The manual tools are identified in the interface to the PLC (VDI) by the **Magazine location no. 1** in the **magazine 9999**. The PLC can detect from this identifier that a manual tool is to be loaded. The PLC ensures that the machine is in a safe state in order to allow the user to perform the manual tool change.

Notice

If the manual tool is loaded, alarm 17212: “Channel %1, Manual tool %2, Duplo No. %3, Load to toolholder %4” is output. The alarm is confirmed by the tool-change acknowledgement from the PLC.

Notice

The PLC is not allowed to reject a manual tool preselected by the NCK (for tool rejection, see also MD 20310: TOOL_MANAGEMENT_MASK).

Block search, program testing

As regards block searches, there is no difference to a normal tool change. However, the corresponding alarms are not generated.

No change commands are output to the PLC during the block search. If a manual tool needs to be loaded when the NC is first started, this can be programmed via **magazine location 1 in magazine 9999** and output of the corresponding alarm.

The data for the tools and magazines have to remain unchanged during the **program-testing mode** in the NCK. A manual tool that has been loaded during program-test selection is therefore removed in terms of its data from the toolholder and saved internally. The stored manual tool is loaded back into the toolholder in response to PLC task “Return manual tool from magazine 9999, location 1”.

Notice

Several toolholders and manual tools can exist in the progam-test mode because of the technology used for the internal storing.

Restrictions

In conjunction with tool selection, tool change and compensation selection, only problems associated with the block correction technique can be rectified that have arisen because of programming errors or incorrectly defined data in the NCK.

Problems or errors resulting from faulty communication between NCK tool management and the PLC cannot be rectified in this way. This type of errors does however only occur when a new PLC program is installed at the machine and not during production by the machine.

The manual tool function is only implemented if TOOL_CHANGE_MODE=1.

3.2.12 Tool changes in NCK by synchronized actions

At tool change and at loading/unloading it is often necessary to supply the NC cycles with the data for the participating tools.
Usually this is done via the “fast data channel” (dual port RAM) using FC 21.
The PLC user program checks the interface in DB71/72/73.
If a new command is pending, the data (new location, old location, T_number,...) are read, pre-processed and supplied to the cycles via FC 21. There they are (usually in synchronized actions) read as variable $A_DBB[...] and e.g. magazine movements are derived from them.

To reduce the overhead involved and create simpler mechanisms, most of the data of the tool management interface was mapped onto the NC variables for read access.

This means that all information about the old tool and the new tool can be read directly in the part program or in synchronized actions; the “detour” via the PLC is no longer needed.

The following variables are used for the mapping process:

$AC_TC_FCT	Function No. (NCK command No.)	
$AC_TC_STATUS	Acknowledge status from PLC	
$AC_TC_THNO	Tool holder or spindle No. on which the change was executed	
$AC_TC_TNO	Internal T No. of the tool to be changed or prepared	
$AC_TC_MFN	Source of **new** tool:	magazine number
$AC_TC_LFN	Source of new tool:	location number
$AC_TC_MTN	Target of new tool:	magazine number
$AC_TC_LTN	Target of new tool:	location number
$AC_TC_MFO	Source of **old** tool:	magazine number
$AC_TC_LFO	Source of old tool:	location number
$AC_TC_MTO	Target of old tool:	magazine number
$A_TC_LTO	Target of old tool:	location number

$AC_TC_CMDT	Trigger variable on command output of the NCK (is set for one IPO)
$AC_TC_ACKT	Trigger variable on acknowledgement from PLC (is set for one IPO)
$AC_TC_CMDC	Counter for the command output
$AC_TC_ACKC	Counter for the acknowledgements

Notice

The variables are read-only (exceptions: $AC_TC_CMDT and $AC_TC_ACKT). The acknowledgement mechanism remains unaffected (as before, the PLC acknowledges all commands from the NCK via FC 8/FC 7).

Method of operation

The variables are written.

1. with **each** command from the **NCK** (**CMD**)
2. with **each** acknowledgement from the **PLC** (**ACK**)
3. with **PowerOn** all are set to value "–1"

The data is retained until it is overwritten by a new command. This means that with commands of the same type, it is not possible to tell from the function number ($AC_TC_FCT) whether a new task is present.

The exceptions are:
$AC_TC_TNO and $AC_TC_THNO
If, for example, the NCK outputs a T preparation, both these variables are set to "–1" with the first PLC acknowledgement via FC 8 (e.g. state 105).

Notice
Scanning should only take place in synchronized actions.
Depending on the application, this can then trigger the variables $AC_TC_CMDT and/or $AC_TC_ACKT.

Example 1

Positioning a tool chain onto the old location

Description: The tool chain has 36 locations, is defined as rotary and indexing axis, increments are 10 degrees therefore each graduation corresponds to one magazine location.
Tool_Change_Mode=1, Tool_Change_M-Mode=6

```
Ids=1 every(($AC_TC_CMDT==1)and(($AC_TC_FCT==2)or($AC_TC_FCT==5)))
        do $R10=itor($AC_TC_LTO)
. . .
if ((R10>0)and($A_DBB[x]==5))
        pos[U1]=cdc(R10)
endif
```

The trigger is sent to the command output of the NCK and with command “2” (T preparation) or command “5” (T/M06 in one block) the old location is read out and stored in R10
(itor=IntegerToReal – format conversion if the variable is stored in the R-variable in synchronized actions).
Later in the program, when the enables from the PLC are present (for example as $A_DBB[x]==5), the magazine axis is traversed to the saved position (old location=$AC_TC_LTO).

A magazine movement could also be started as follows (shown here in simplied form):

```
Ids=1  every(((($AC_TC_FCT==2)or(AC_TC_FCT==5))and
       ($AC_TC_STATUS==105))and(($AC_TC_LTO>0)))
       do pos[U1]=cdc($AC_TC_LTO)
```

With commands “2” and “5” (T preparation or T/M06), with old location>0 and PLC acknowledgement status “105” (serves as enable), the magazine axis is traversed.

Old location>0: If the spindle was empty, there is no old tool and the old location is 0. Therefore, the magazine axis does not need to move.

Example 2

Swiveling a turret

Description: Turret, 6 locations, the turret is defined as an indexing axis, 60-degree increment, corresponds to one tool location, 1xSpindle, Tool_Change_Mode=0

```
Ids=1 every($AC_TC_CMDT==1)and($AC_TC_FCT==4)and($AC_TC_LFN>0)
        do $R10=itor($AC_TC_LFN)
. . .
if ((R10>0)and($A_DBB[x]==5))
        pos[B]=cac(R10)
endif
. . .
```

The trigger is sent to the command output of the NCK and with command "4" (change with T command) the new location is read out and saved in R10 (itor=IntegerToReal – format conversion if the variable is stored in the R-variable in synchronized actions).
Later in the program, when the enables from the PLC are present (for example as $A_DBB[x]==5), the turret is traversed to the saved position (new location=$AC_TC_LFN).
The logic operation $AC_TC_LFN>0 prevents a movement from taking place if, for example, TO was programmed.

3.2.13 Tool change cycle (shopfloor-oriented interface)

The tool change is initiated by a cycle for the shopfloor-oriented interface. For a more detailed description please refer to the documentation:

References: /BAS/ ShopMill Operation/Programming

References /FBSP/ Description of Functions, ShopMill

3.2.14 Example for cycle T function replacement (SW 6)

Both a turret head as well as a magazine with several buffer locations can be realized in one channel for transporting the tool into the spindle with the new function "T-function replacement".

The in the turret can be called with T Dxx and the tool in the tool-holding magazine can be pre-selected with a T call and be loaded with M6 Dxx.

Prerequisite is the channel-specific setting for the spindle ($MC_TOOL_CHANGE_MODE=1). It can furthermore be defined in the NCK by the type of magazine those magazines where the spindle display shall be suppressed.

General

Considered is a turning unit with tool feed between SAT spindle and tool disk-type magazine via gripper (turret head and chain in one channel).

NCK magazine configuration

Magazine no.	Location No.	Meaning	Assignments
Loading magazine			
9999	1	Spindle loading point	
9999	2	Loading point turret/magazine	Magazine 1–2, distance=0
Buffer magazine			Distance=0
9998	1	Spindle 1	Magazine1
9998	2	Spindle 2	Magazine2
9998	3	Gripper 1	Spindle 2, magazine 2
9998	4	Gripper 2	Spindle 2, magazine 2
Magazine 1 (turret 1), defined as chain			
1	1	Magazine location 1	
1	2	Magazine location 2	
1	.	Magazine location ...	
1	.	Magazine location ...	
1	12	Magazine location 12	
Magazine 2 (disk right), type chain			
2	1	Magazine location 1	
2	2	Magazine location 2	
2	.	Magazine location ...	
2	.	Magazine location ...	
2	32	Magazine location 32	

Assignment DB4

The magazine type in OB100 was changed when presetting the DB4 from “turret” to “chain”.

Address in DB4	Value	Meaning
DBB 64	4	Maximum number of magazines
DBW 65	1	Magazine no.
DBB 67	1 (3)	Type of magazine
DBW 68	12	Number of locations
DBW 70	2	Magazine no.
DBB 72	1	Type of magazine
DBW 73	32	Number of locations
DBW 75	9998	Magazine No. intermediate memory
DBB 77	7	Type of magazine
DBW 78	4	Number of locations
DBW 80	9999	Magazine No. loading magazine
DBB 82	9	Type of magazine
DBW 83	2	Number of locations
DBB 85	2	Number of spindles

Type of magazine:1=chain
3 = Turret,
5 = Box-type magazine,
7 = Buffer,
9 = Loading magazine

Machine data

The setting Spindle has been activated in each channel (basic setting = turret) and the T-function replacement used.

MD 22550: $MC_TOOL_CHANGE_MODE = 1

MD 22560: $MC_TOOL_CHANGE_M_CODE = 6

MD 10717: $MN_T_NO_FCT_CYCLE_NAME = T_CYCLE

PLC program

The interface to the PLC now lies in DB 72 for T or M6 respectively because of TOOL_CHANGE_MODE =1. Acknowledgement is however only given via FC 8.

The program Testwzv.awl from the tool box has been linked for acknowledging the jobs with FC 8. The default selection of FC8 parameters in DB 62 has been changed by the variable Monitor/controlbefore the acknowledgement and in accordance with the operational sequence in the machine, i.e. preparation with tool in the gripper has been acknowledged for a T call of a tool in magazine 2. For the tool change M6, the tool in the spindle is first held by the gripper and the new tool then put into the spindle.

Interface to data changes prior to the acknowledgement:

DB62.DBW 2 = Magazine for new tool
DB62 DBW 4 = Location for new tool
DB62.DBW 6 = Magazine for old tool
DB62.DBW 8 = Location for old tool
DB62.DBW 10 = Status

T function replacement

The function replacement of the T number in combination with the setting Spindle has been changed since channel-specific, only changing with T or M06 (TOOL_CHANGE_MODE =0/1) can be set and the requirement exists to program the turret with T Dxx and to prepare a tool from the disk-type magazine with T and to load with M6 into the spindle.

A cycle entered in MD 10717: $MN_T_NO_FCT_CYCLE_NAME is called if the T number is programmed. In this cycle, the T number is first programmed and evaluated as to whether there is a job “Prepare tool” pending for a turret or for a disk-type magazine.
With pre-selected magazine = turret, the function M06 Dxx is programmed in the cycle; if a tool is selected from a disk-type magazine, only the T number is output in the cycle.

Notice

It is not possible to replace the language commands TCA and TCI via this machine data.

Cycle T-function replacement

```
%_N_T_ZYKLUS_SPF
;$PATH=/_N_CUS_DIR
IF $C_T_PROG==TRUE          ;T number numerical?
   T[$C_TE]=$C_T            ;Select T number
ELSE
   IF $C_TS_PROG==TRUE      ;T number=string?
      T[$C_TE]=$C_TS        ;Select T number
   ENDIF
ENDIF
IF ($C_TE==2)               ;expanded T address 2?
   M17                      ;T output only as tool in disk selected
ELSE
   IF $C_TE==0              ;expanded T address=0?
      IF ($P_MTHNUM==2)     ;MasterToolHolder 2?
         M17                ;T output only, as tool in disk selected
      ENDIF
   ENDIF
ENDIF
M6                          ;Tool change as new tool in turret
IF $C_D_PROG==TRUE          ;D number selected?
   DL=$C_DL                 ;Select DL number
ENDIF
M17
```

For cases where you have to replace the TCA or TCI command for specific applications, this can be achieved via the NC functionality “Reconfigure NC codes”.

Example with TCA command

Set machine data:

$MN_NC_USER_CODE_CONF_NAME_TAB[0]="TCA" original NC code
$MN_NC_USER_CODE_CONF_NAME_TAB[0]="_TCA" reconfigured code

User cycle:

Create a cycle (in the Customer directory) with the name of the original NC code which is going to be reconfigured – i.e. TCA.

```
%_N_TCA_SPF
;$PATH=/_N_CUS_DIR
proc TCA(string[64]identifier,int Duplo,int TH_No)
. . .                              Scanning and logic
_TCA(identifier,Duplo,TH_No)       the original TCA command is called here
. . .
M17
```

Part program

```
%_N_A_MPF
;$PATH=/_N_MPF_DIR
. . .
TCA("Tool1",1,1)                   is replaced by the cycle
. . .
M06
. . .
M02
```

This procedure can also be used for T function expansion.

3.2.15 Block search

Block search with calculation

On a block search, selection on reset or start, the tool is selected during preprocessing. In this case the PLC is not allowed to reject the tool (see bit 4 in MD 20310). If it does, an alarm is generated. The block search must then be repeated. Use of the active tool can only be prevented from an external source (HMI, PLC).

In block search with calculation the program is generally put into a state where the selected block can be executed. With respect to the tool management function, this means that the tool that should be located in the spindle when the machining block is reached must now be loaded to it.

If another tool is located in the spindle a “replace” command is initiated. In such a case, the signals “Prepare change” (DB72.DBX(n+0).2 and “Execute change” (DB72.DBX(n+0).1 are present at the same time since the Help functions are outputted together.

Example: $MC_TOOL_CHANGE_MODE=0

Tool “Drill1” is loaded in the spindle. The new search target has T = “Drilling machine 2” as the momentary tool programming.

NCK initiates the tool change. PLC must not intervene.

Notice

Tool rejection by PLC: If bit 4 of machine data MD 20310: TOOL_MANAGEMENT_MASK set, then PLC usually has the possibility to again request preparation for a tool change, yet this time with different parameters, i.e. to reject the tool. This is not possible during block search. In this case, the machine data setting is ignored.

Notice

Because the tool change is frequently performed using cycles, a “replace command” generated by the block search must be executed in an asynchronous subroutine (ASUB). Modal and static motion-synchronization action is retained at the beginning of ASUB and is also effective in the asynchronous subroutine. If the asynchronous subroutine is not continued with Repos, the modified modal and static motion-synchronous actions in the main run remain operative.

Alternatively, execution of the NC part program can be stopped by halting feed and read-in, and a fault message “Wrong tool in spindle after the block search” can be generated.

Tool cannot be used

If the tool to be loaded is not located at the search destination, an attempt is made to enable a disabled tool. If no suitable tool is found, alarm 22068 is output. The alarm can only be cleared by a RESET.

If further tool changes are programmed, the disabled tool is not tagged for future block searches and the search operation is not interrupted. However, if an attempt is made to load the disabled tool on a start after the end of the block search, the NCK outputs alarm 22067. The program cannot be resumed. With SW 5.1 and later, the PLC can be used to control whether or not the disabled tool is loaded anyway.

Example of a search with block splitting effect

```
N100 T="Tool1" M6 D1
N110 SETMTH(1)                   ; Toolholder1 becomes master tool-
                                 ; holder
N120 T="Tool2" M6 D2             ; Target block: is not yet inter-
                                 ; preted
   N1000 IF($P_PROG_EVENT ==5)   ; ASUP is started
   .............
   N1020 SETMTH(2)               ; Toolholder2 becomes master tool-
                                 ; holder
   .............
   N1040 ENDIF
   N1099 REPOSA
N110 SETMTH(1)                   ; The interrupted main program is
                                 ; continued/started after the last
                                 ; executable block before the search
                                 ; target.
                                 ; Toolholder1 becomes master tool-
                                 ; holder again.
```

3.2.16 Block search (SSL) in conjunction with active tool management

The block search is described in Section 3.2.15. Here we shall deal with the specific features in conjunction with active tool management.

The block search establishes the start position of the target block. Auxiliary functions programmed in SSL are collected and output in action blocks at the end of the SSL.

For now this also applies to the T command and M06. This depends on the setting in machine data 20128: $MC_Collect_Tool_Change

0 = From software version 7: Neither T preparation nor M06 are output.

1 = < SW version 7: T preparation and M06 are collected and output (and must be acknowledged to end the block search).
Default setting.

The following examples show how to proceed with block search.

Configuration: milling machine, one spindle
Settings: $MC_Tool_Change_Mode=1, i.e. change with M06
$MC_Collect_Tool_Change=1

Block search up to software version 5

In principle, you need to define whether to subsequently effect a tool change or not. The check can be performed as follows.

Do not subsequently effect tool change

- PLC checks whether a tool change request is pending (DB72.DBX4.2) after executed block search (DB21.DBX33.4) and before output of the last action block (DB21.DBX32.6).
- If this is not the case, the spindle tool does not correspond to the tool requested for the block search, and the PLC must prevent an NC Start from taking place.

Subsequently effect tool change

- After the block search, the collected change request is acknowledge as negative via FC 8, the internal T number is saved in the PLC.
- A collected T preparation is acknowledged as positive via FC 8.
- Now the PLC starts an ASUP which carries out the change and subsequent T preparation.
  The tool to be loaded is saved in intermediate memory in the PLC and must be made available to the ASUP, e.g. via dual-port RAM variable.
  The tool to be prepared is read in the ASUP via GETSELT.

Block search with software version 6

Tool change is subsequently effected:

<u>Situation:</u>
T=“Face_80mm” is located in the spindle
Block search to N98 (block search with contour calculation)

<u>Destination:</u>
In order to continue in the program:
a. Tool “1537” must be changed
b. Tool “Drill_6mm” must be prepared

```
. . .
N10 T="1231"                ;TNo. 1
. . .
N20 M06
```

```
. . .
N30 T="Face_80mm"          ;TNo. 2
. . .
N70 M06
. . .
N80 T="1537"               ;TNo. 3
. . .
N90 M06
. . .
N95 T="Drill_6mm"          ;TNo. 4
. . .
N98 ...
```

Settings:
$MC_Tool_Change_Mode=1
$MC_Collect_Tool_Change=1
$MN_Search_Run_Mode Bit 1=1

$MC_Collect_Tool_Change=1 means: **Output** of T and M06 to block search.

Procedure:

- The following takes place in the action blocks:
  The change from tool "1537" (Tx/M06) is output
  –> PLC sends a negative acknowledgement
  Preparation of tool "Drill_6mm" is output
  –> PLC sends a positive acknowledgement
- The program "Prog_Event.SPF" is started with the last action block.
  The change and preparation are now carried out.

Prog_Event.SPF

```
. . .
def int T_Vor, T_Spi, T_active
. . .
```

T_active=$P_TOOLNO	The active tool (block N80 and N90 are read T_active=3 This tool must be changed.
GETSELT(T_Vor)	T preparation is read from block N95 T_Vor=4
T_Spi=$TC_MAP6[9998,1]	The actual spindle tool is read T_Spi=2
If...	scan (see next example)

Block search with software version 7

Tool change is subsequently effected:

Situation:
T="Face_80mm" is located in the spindle
Block search to N98 (block search with contour calculation)

Destination:
In order to continue in the program:
a. Tool “1537” must be changed
b. Tool “Drill_6mm” must be prepared

```
. . .
N10 T="1231"                ;TNo. 1

. . .
N20 M06

. . .
N30 T="Face_80mm"           ;TNo. 2

. . .
N70 M06

. . .
N80 T="1537"                ;TNo. 3

. . .
N90 M06

. . .
N95 T="Drill_6mm"           ;TNo. 4

. . .
N98 ...
```

Settings:
$MC_Tool_Change_Mode=1
$MC_Collect_Tool_Change=0
$MN_Search_Run_Mode Bit 1=1

$MC_Collect_Tool_Change=0 means: **No output** of T and M06 to block search.

Procedure:

- Negative acknowledgement is not required in the PLC.
- The program “Prog_Event.SPF” is started with the last action block. The change and preparation are carried out in retrospect.

Prog_Event.SPF

```
. . .
def int T_Vor, T_Spi, T_active

. . .
```

GETEXET(T_active)	The spindle tool is read from NCK view (block N80 and N90) T_active=3
GETSELT(T_Vor)	T preparation is read from block N95 T_Vor=4
T_Spi=$TC_MAP6[9998,1]	The actual spindle tool is read T_Spi=2

. . .

```
;Load correct tool
if ((T_Spi< >T_active)and(T_active>0))
  T=$TC_TP2[T_active]                Preparation of tool "1537"
  L6 ;change cycle                   Load tool "1537" for change
Endif

. . .
if T_Vor< >T_active
  if T_Vor>0
  T=$TC_TP2[T_Vor]         Preparation of tool "Drill_6mm" from block N95
  Endif
  If t_Vor==0
  TO
  Endif
endif
```

Notice

1. If a change is output by the action blocks (in example block N80 and N90), it is always a command "5", i.e. "Prepare change" and "Perform change" are pending in DB 72 at the same time.
2. If the correct tool is already placed in the spindle (i.e. in the block search example at block N70 and $MC_Collect_Tool_Change=1 is set), the T preparation is issued (from block N30).
   The setting for bit 12=0/1 in the MD $MC_Tool_Management_Mask machine data is not evaluated.
3. Difference between the commands GETEXET and $P_TOOLNO:
   GETEXET
   Reads the T No. of the tool in the spindle from the NCK's point of view.
   Independent of an offset selection.
   Was specifically developed for use with block search.
   $P_TOOLNO
   Reads the T No. of the active tool.
   This does not refer to the "active status" of the tool which is set via the T preparation, instead it refers to the tool whose offset is being calculated. This view of the tools means that a tool doesn't become an active tool until the offset is selected – which is what read with $P_TOOLNO.
   This command is dependent on machine data $MC_Cutting_Edge_Default.

Example:

```
. . .
N100 T="Counterbore"          ;T No. 5
N110 M06
N108 G90 G00 D1 X...

. . .
N200 T="Drill"                ;T No. 32
N210 M06
N208 G90 G00 D1 X...
```

Block search to block N200

1. Setting $MC_Cutting_Edge_Default=–2
   GETEXET = 5
   $P_TOOLNO = 5
2. $MC_Cutting_Edge_Default=1
   GETEXET = 5
   $P_TOOLNO = 5

Block search to block N212

1. Setting $MC_Cutting_Edge_Default=–2
   GETEXET = **32**
   $P_TOOLNO = **5**
2. $MC_Cutting_Edge_Default=1
   GETEXET = 32
   $P_TOOLNO = 32

3.2.17 Program testing

The "program testing" function can be used to traverse a program without axis motion.

All other data is determined and calculated. For tool management, this means that the tools are searched and the appropriate values transferred to the PLC interface when the tool is called.

The PLC must acknowledge these jobs without moving the magazine or changing a tool. Special measures are therefore required on the PLC.

The tool management acts in exactly the same way as it would when a program is running. In the case of tools without fixed location codes and acknowledgement, this can result in the PLC data indicating different tool locations to the actual mechanical locations in the magazine. This can be prevented by configuring FC 8 such that a fixed location is simulated for the duration of the program test rather than the calculated empty location applied as a parameter.

The old location of the tool is stored in the function block which handles program testing and returned again to this location in the software (data settings). Any existing spindle tool is also returned to the spindle in the software at the end of the program test or on a reset. This ensures that the magazine assignments in the software match the mechanical assignments after program testing.

For testing programs involving manual tools, see Section 3.2.11.

Example of how to adapt the PLC in test mode

The following example program can be used as a template for adapting the PLC to program testing mode. Only the first channel and a spindle are supported as tool change locations.

The tool is always changed directly into the spindle. The spindle is used as the change position (DB 72). Access to the NCK/PLC interface (DB 21, 72) is symbolic. The standard UDTs (UDT 21, 72) are included for this purpose. These are stored on the basic program diskette and must be copied into the project and subsequently compiled.

The following must be entered in the symbol table:

Symbol	Address	Data type	Comment
Channel1	DB 21	UDT 21	
SpChPos	DB 72	UDT 72	
TC_VAR	DB 119	DB 119	For testing the tool change

All necessary variables are stored in the instance data block.

If program testing mode is **deselected**, no action occurs. The target positions suggested by tool management are confirmed by the PLC.

If program testing mode is **selected**, the target positions are defined by the PLC. These correspond to the source positions of the respective tools. The target position is only confirmed and saved by tool management on the first tool change. It is thus possible to undo the first tool change after program testing mode is selected.

Two asynchronous transfers are required for this purpose. The first one returns any tool present in the spindle to the magazine. The second asynchronous transfer is intended to return a tool which was loaded in the spindle before program testing mode back into the spindle.

Notice

The relevant PLC example is stored in the toolbox. The sample file WZV_PROG.AWL is packed in file WZV_BSP.EXE.

Program testing – extended

A setting can be selected with the machine data $MC_TOOL_MANAGEMENT_MASK – Bit 20 such that the NCK cannot issue any tool-changing commands to the PLC in the status Program test active. It outputs its own acknowledgement such there is no further data-related tool motion.

The disabling of tool change command outputs is selected intentionally as the default, even though this renders the software incompatible with earlier NCK versions.

The following applies for the tool used during program test mode:
The tool status “active” can still be set and the tool status “Was in use” is set. This does not have any further detrimental effects since tool monitoring is not normally active in the test mode.

With **bit 20**, value **1** is set, generated commands are output to the PLC. Tool / magazine data can be change in the NCK here depending on the type of acknowledgement by the PLC. If the acknowledgement parameters for the “target magazine” are set to the values of the “source” magazine, then the tool is not moved and the data therefore not altered in the NCK.

Exception: The tool status of the tool activated in the test mode can take the status “active”.

Notice

It may not be derived – to the extent the setting “No tool-change commands to PLC” – that the tool on the spindle in the toolholder during “Program test active” is the active tool.

3.2.18 Several spindles in one channel or TO units

When using tool management and more than one spindle please note the following:

Two spindles in one channel

Only one tool offset can be active per channel. Spindle 1 defined as master spindle with $MC_SPIND_DEF_MASTER_SPIND = 1. Spindle 2 is a secondary spindle.

The master spindle is spindle no. 1 in each case.

Two channels each of which access the same TO memory have been set in the machine data. One spindle is assigned to each channel. Two spindles are assigned to one magazine in the machine configuration.

The master spindle is spindle no. 1 for both channels In order to change a tool in spindle no. 2 as well, the second spindle must be defined as master spindle in the second channel before the tool is changed. In the TM system, the spindle number is sent to the PLC. This number is determined from the extended address of T. If this is not programmed, the NCK assigns the master spindle number of the channel in which the program is running (Fig. 3–9).

Each channel has its own master spindle

Two channels each of which access the same TO memory have been set in the machine data. One spindle is assigned to each channel.

Two spindles are assigned to one magazine in the machine configuration.

In each channel the assigned spindle is defined as the master spindle. Tool change is possible without making any additional definitions in the program.

3.2.19 Decoupling the tool management from the spindle number

The program must specify the location (spindle number on milling machines) at which the tool is to be changed before the tool management can insert a tool.

Using the machine data **MD 20124: TOOL_MANAGEMENT_TOOLHOLDER** can be set to determine whether a toolholder number must be assigned to define the location of the tool to be loaded instead of a spindle number. Thus the appropriate designation (spindle number or toolholder number) can be used in the event of use.

The following figures show which variable definitions you require for the following variants:

- Working with two spindles in one channel and one TO unit (standard functionality)
- Working with two spindles in one channel (standard function)
- Working with 2 toolholders in 2 channels (one TO unit)
- Working with two toolholders in one channel

Working with spindle numbers

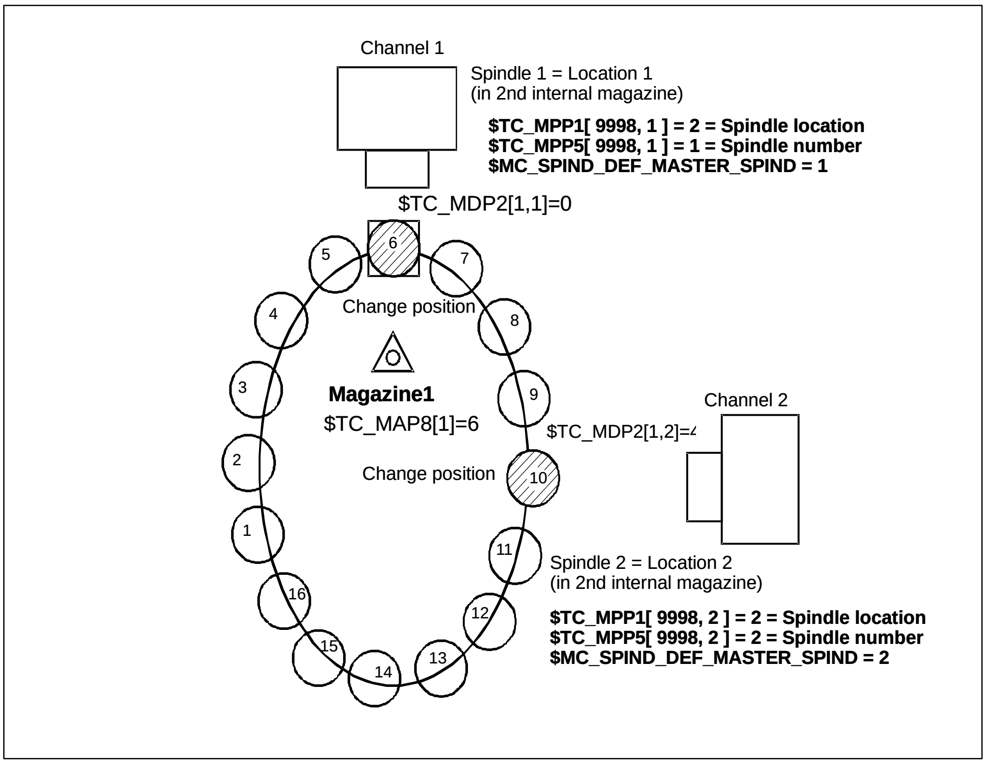


Bild 3-8 Two spindles in two channels and one TO unit

Two channels operating with the data of one TO unit (with one magazine). One spindle is defined in each channel.

Spindle 1 in channel 1 has been declared the master spindle with MD SPIND_DEF_MASTER_SPIND=1. Spindle 2 on channel 2 is the master spindle.

Both spindles must be assigned different numbers because the assignment of the spindle to the second internal magazine (buffer magazine) must be unique.

This assignment is realized by $TC_MPP1 (spindle location) and by $TC_MPP5 (spindle number).

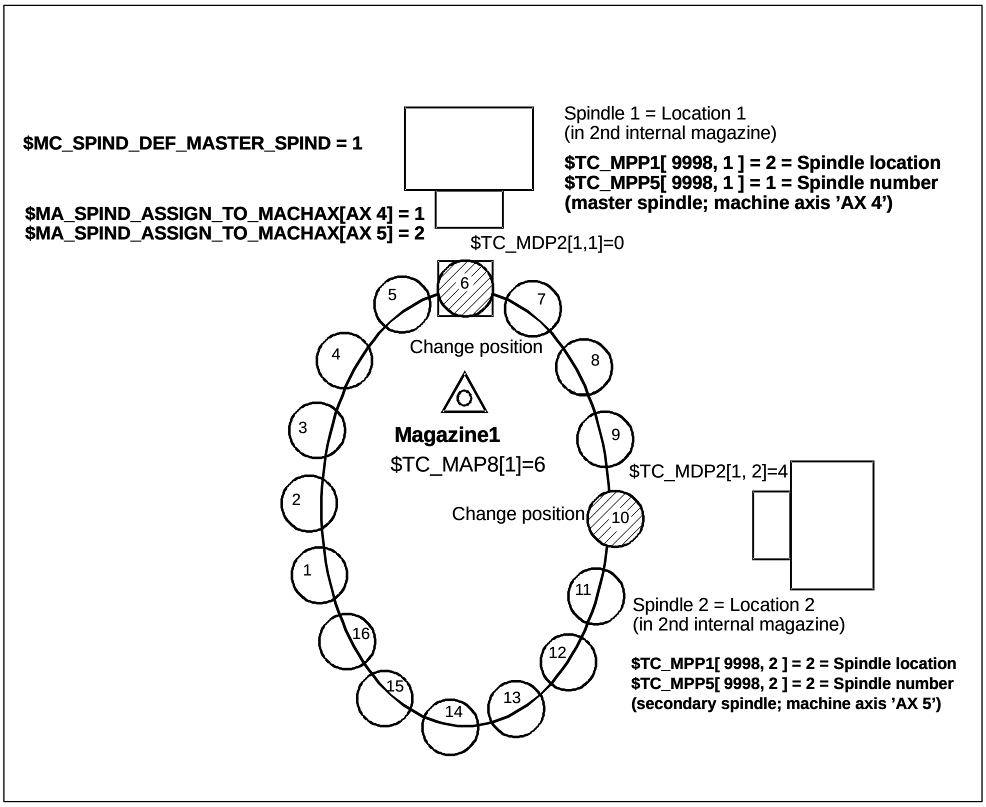


Bild 3-9 Two spindles in one channel

Two spindles of a single channel are operating with one magazine.

Spindle 1 defined as master spindle with SPIND_DEF_MASTER_SPIND = 1.

Spindle 2 is not a master spindle (secondary spindle).

References: **/PGA/**, "Programming Guide Advanced"
(description of system variables)

Example of a part program (for a channel with two spindles)

(Requirement: CUTTING_EDGE_DEFAULT=1;
i.e. D1 is implicitly active with the tool change M6):

```
T="Milling tool" M06   ; No address extension programmed -> the master
                       ; spindle is addressed, i.e. spindle 1 = value of
                       ; machine data $MC_SPIND_DEF_MASTER_SPIND.
                       ; The tool change takes place in spindle 1.
                       ; The path is corrected with the tool offsets
....
T2="Drill" M2=6        ; Address extension for secondary spindle has been
                       ; programmed. The tool change takes place in the PLC
                       ; at the tool management interface for spindle 2.
                       ; The path is not corrected
. . .
SETMS(2)               ; Declares spindle number 2 as master spindle
T="Milling tool_2" M6  ; No address extension programmed -> The master
                       ; spindle is addressed (spindle 2).
                       ; Tool change takes place in spindle 2.
                       ; The path is correctedwith the tool offsets.
....
T1="Drill_1" M1=6      ; Address extension for current secondary spindle
                       ; was programmed.
                       ; Tool change takes place in spindle 1.
                       ; The path is corrected with values from tool
                       ; T="Milling tool_2".
. . .
SETMS                  ; Declares the spindle defined by $MC_SPIND_DEF_MAS-
                       ; TER_SPIND as master spindle
T="Milling tool_3" M6  ; No address extension programmed -> The master
                       ; spindle is addressed (spindle 1).
                       ; Value of machine data $MC_SPIND_DEF_MASTER_SPIND).
                       ; Tool change takes place in spindle 1.
                       ; The path is corrected with the tool offsets.
```

Further example (starting conditions as above):

```
N10 SETMS           ; Declare spindle number 1 as master spindle
N20 T2=3
. . .
N50 M2=6            ; Address extension for secondary spindle has been pro-
                    ; grammed. Tool change is performed and tool is placed
                    ; into buffer location 2.
                    ; The path is not corrected
. . .
N70 D3              ; The path is correctedwith the offsets of the active
                    ; tool (activated before block N10).
N80 SETMS(2)        ; Declare spindle number 2 as master spindle
T3
M06
N90 D2              ; The path is corrected with the offsets of the active
                    ; tool T3.
```

Notice

SETMS does not change the active tool. The new master spindle definition cannot be referenced until the subsequently programmed tool change.

Working with toolholder numbers

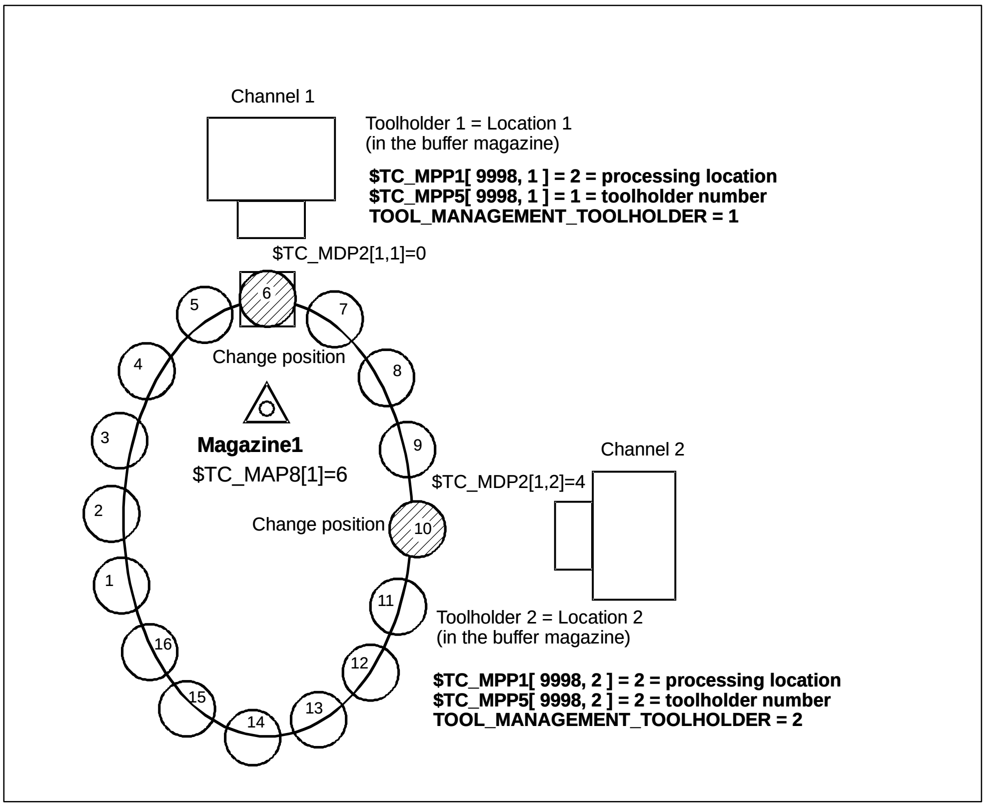


Bild 3-10 Two channels with one toolholder each and one TO unit (the zero position is at the tool change position of toolholder 1)

Two channels are operating with the data of one TO unit (with one magazine). Tool change no longer requires that a spindle number be specified. The address expansions of T and M now refer to the value for machine data MD 20124: TOOL_MANAGEMENT_TOOLHOLDER

Instead of "spindle location" the general term "tool machining location" is used (spindle is standard). If no address extension is programmed, the value in MD 20124: TOOL_MANAGEMENT_TOOLHOLDER is added as the extension.

TOOL_MANAGEMENT_TOOLHOLDER = 0

The previous function remains active (default).

A value greater than zero activates the new function.

TOOL_MANAGEMENT_TOOLHOLDER > 0

If a tool change is programmed to a buffer location of the type “Tool processing location” with $TC_MPP5 = TOOL_MANAGEMENT_TOOLHOLDER, then the compensation data defined for this tool (TO unit) correct the path.

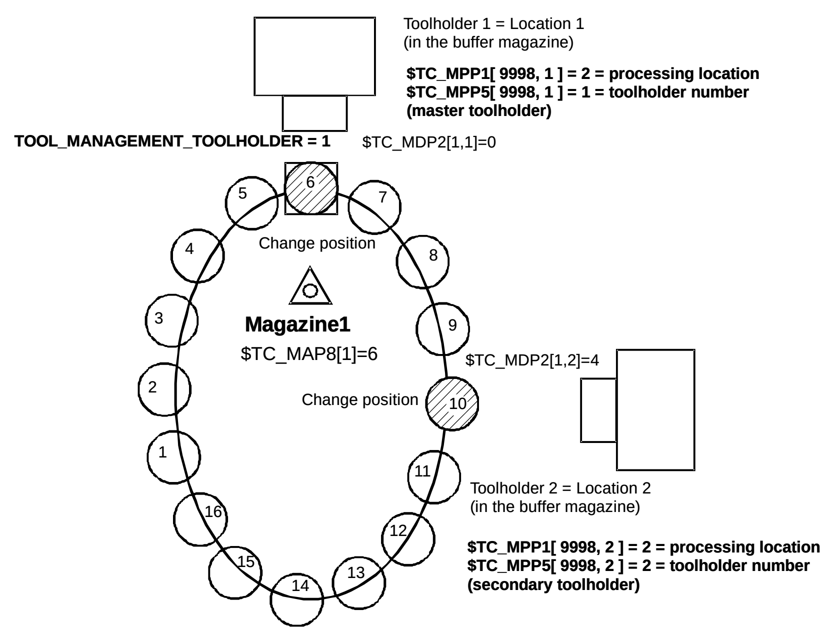


Bild 3-11 One channel with two toolholders (zero position is at tool change position of toolholder 1)

Two toolholders in a channel are operating with one magazine. Toolholder 1 has been declared the master via **TOOL_MANAGEMENT_TOOLHOLDER = 1**. Toolholder 2 is thus the secondary toolholder.

Programming example:

In order to declare different toolholders as master toolholder, the NC command

SETMTH (toolholder number),

is used.

```
T=“Milling tool” M6    ; No address extension programmed -> The master
                       ; toolholder is addressed (toolholder 1 - value of
                       ; machine data $MC_TOOL_MANAGEMENT_TOOLHOLDER).
                       ; Tool change is performed and tool is placed into
                       ; buffer location 1.
                       ; The path is corrected with the tool offsets.
. . .
T2=“Drill” M2=6        ; Address extension for secondary toolholder has
                       ; been programmed.
                       ; Tool change is performed and tool is placed into
                       ; buffer location 2.
                       ; The path is not corrected
. . .
SETMTH(2)              ; Declares toolholder 2 the master toolholder
T=“Milling tool_2” M6  ; No address extension programmed -> The master
                       ; toolholder is addressed (toolholder 2).
                       ; Tool change is performed and tool is placed into
                       ; buffer location 2.
                       ; The path is corrected with the tool offsets
. . .
T1=“Drill_1” M1=6      ; Address extension for the secondary toolholder has
                       ; been programmed.
                       ; Tool change is performed and tool is placed into
                       ; buffer location 1.
                       ; The path is not corrected!
. . .
SETMTH                 ; Declares the toolholder specified in $MC_TOOL_MAN-
                       ; AGEMENT_TOOLHOLDER as the master toolholder
T=“Milling tool_3” M6  ; No address extension programmed -> The master
                       ; toolholder is addressed (toolholder 1 - value of
                       ; machine data $MC_TOOL_MANAGEMENT_TOOLHOLDER).
                       ; Tool change is performed and tool is placed into
                       ; buffer location 1.
                       ; The path is corrected with the tool offsets.
```

References: /PGA/ Programming Guide Job planning
(description of system variables)

Notice

SETMTH does not change the active tool. The new master toolholder definition cannot be referenced until the subsequently programmed tool change.

3.2.20 Several spindles/toolholders

Tool management can work in one channel with more than one toolholder. If several channels of one TO unit are supplied with data, then make sure that the toolholder numbers have different (= unambiguous) numbers in the magazine configuration ($TC_MPP5 of buffer locations of the type ($TC_MPP1) “Spindle”). The spindle numbers of the channels must then be unique as well (if $MC_TOOL_MANAGEMENT_TOOLHOLDER=0).

Example

This example shows how to differentiate between an active tool and a programmed tool.

Channel 1 has the spindle numbers 1, 2 and channel 2 has the spindle numbers 3, 4. The TO unit assigned to these channels then has the four spindle locations 1, 2, 3, 4 defined at the buffer location.

```
SETMS(2)
T                ; 12 is a programmed tool
12
M6 D3            ; 12 is an active tool, 3 is an active cutting edge
SETMS(4)
T22              ; 12 remains an active tool, 22 becomes with respect to
                 ; toolholder=4 programmed tool
T3=33 M3=6       ; T33 is neither programmed nor active
SETMS(1)         ; Toolholder=1 becomes master toolholder T12 remains
                 ; active, T22 remains programmed
D5               ; D5=active cutting edge; refers to the active tool, i.e. T12
M00
```

The following situation is given:

Tool holder number	T number	D number
1 master spindle	–	–
2	12 active	5 active
3	33 –	–
4	22 programmed	–

3.2.21 Several magazines in one channel or one TO unit

The NC address T can be programmed with an address extension. The tool management function interprets the programmed address extension as a spindle number or toolholder number. The NC address T without programmed address extension then refers to the main spindle (master spindle).

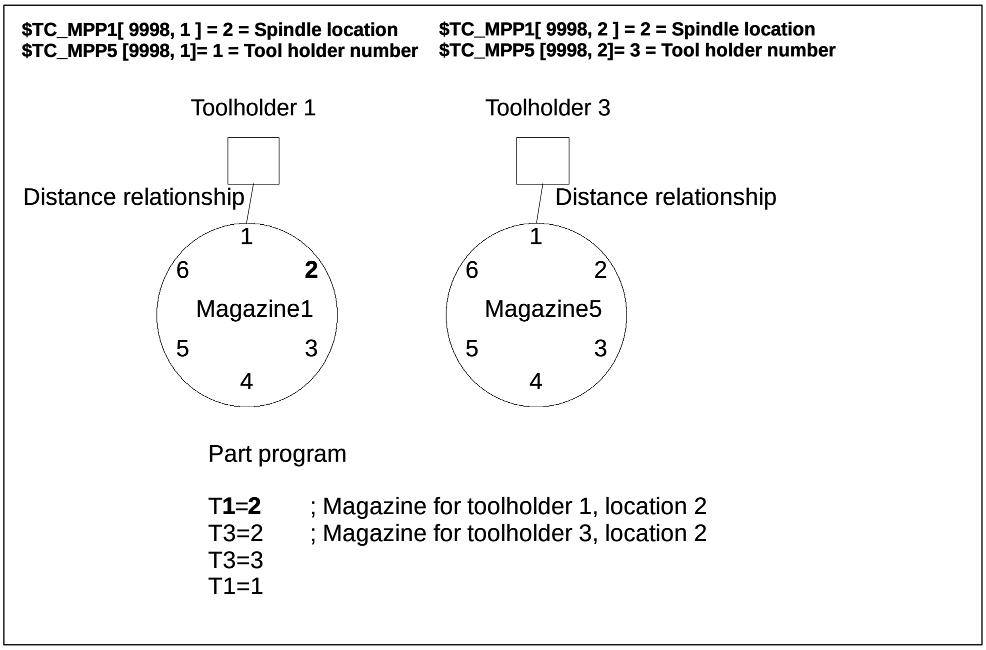


Bild 3-12 T="location" and several magazines in the same channel

The figure shows the procedure for using more than one magazine in a channel (when programming with T="location" this is usually a turret).

Notice

The tool offset is only calculated for the toolholder that is assigned at that point in time to the master spindle or the master toolholder.

3.2.22 Reset and start mode

The tool offset selection/deselection can be set in the machine data for program end or reset as well as for NC Start.
It is also possible to permanently preset the change for a specific tool. e.g. at NC Start.

The settings are made in the following machine data:

MD 20310: $MC_TOOL_MANAGEMENT_MASK

MD 20110: $MC_RESET_MODE_MASK

MD 20112: $MC_START_MODE_MASK

MD 20122: $MC_TOOL_RESET_NAME

MD 20130: $MC_CUTTING_EDGE_RESET_VALUE

The function and interaction of the machine data are displayed in Fig. 3-13 “Reset and start mode”.

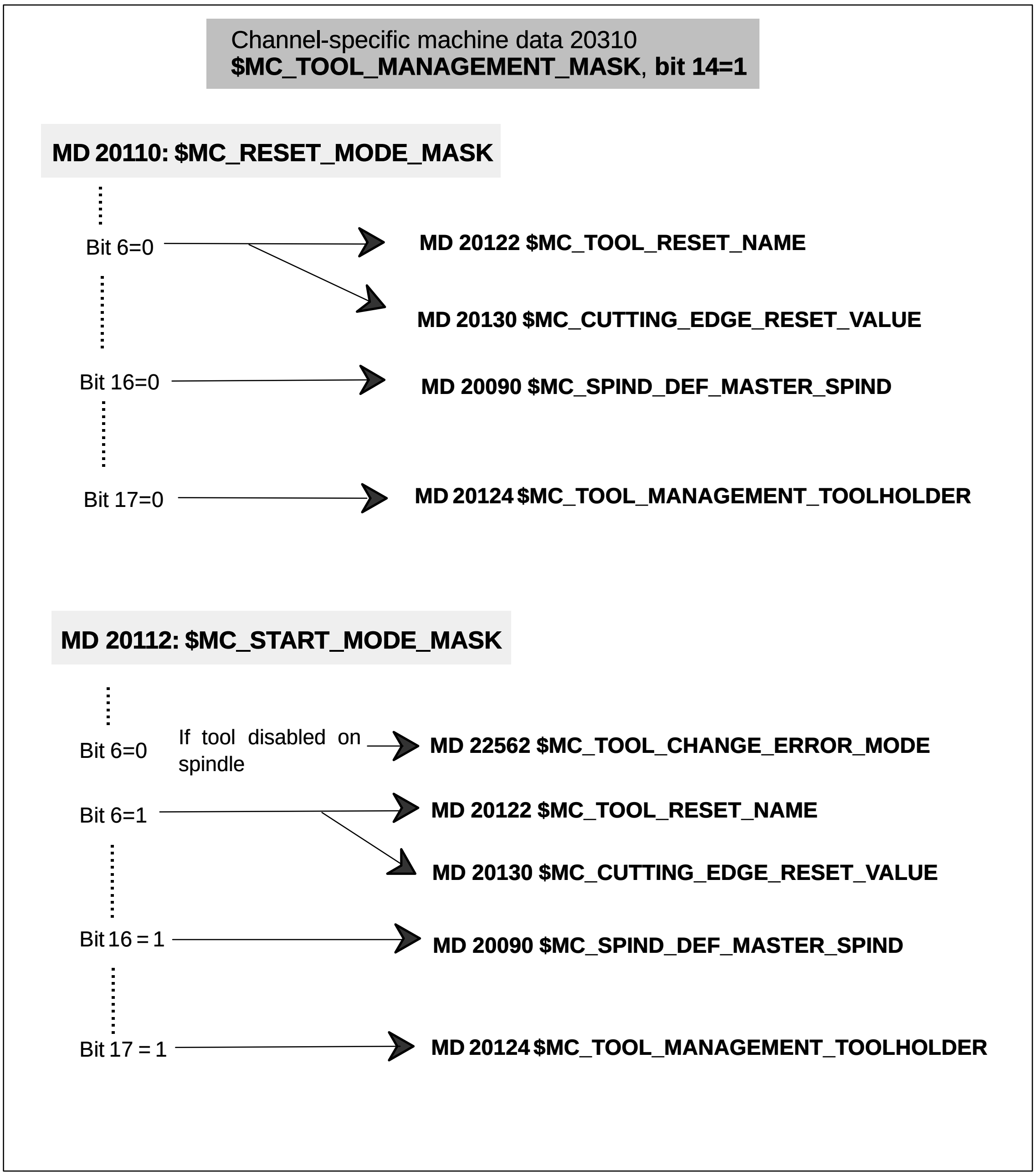


Bild 3-13 Reset and start mode

MD 20110: $MC_RESET_MODE_MASK

Bit 0=0:
Compatibility bit for SW 1
Meaning: leave offset unchanged, i.e. after end of part program and reset, the offset last programmed remains active (behavior as with bit 0 and 6=1).

Bit 0=1:
Reset mode, i.e. evaluation of bits 4 ..11

Bit 2=1:
Reset behavior (tool offset) with tool management not active. No effect if tool management active

Bit 6=0:
Reset behavior corresponds to MD $MC_TOOL_RESET_NAME and $MC_CUTTING_EDGE_RESET_VALUE

Bit 6=1:
Current setting for active tool length compensation is retained beyond reset/end of part program.
If tool management is active, the tool is selected which is is positioned on the master spindle (general: master toolholder).
If the tool on the spindle is disabled, this state is ignored, there is no selection of a replacement tool! (Replacement tool with Start_INIT only).
Activation takes place on the master spindle defined in MD $MC_SPIND_DEF_MASTER_SPIND, or on the master toolholder defined in $TC_TOOL_MANAGEMENT_TOOLHOLDER.
From software Version 6.3, if you want to deviate from the setting in the MD, you can also activate the tool on the master spindle/master toolholder that was last programmed. Bit 16 or 17 are used for this purpose.

Reset behavior for spindles

Bit 16=0:
The master spindle is the spindle defined in MD $MC_SPIND_DEF_MASTER_SPIND.
The settings in the machine data below refer to this data
$MC_TOOL_MANAGEMENT_MASK
$MC_RESET_MODE_MASK
$MC_START_MODE_MASK
$MC_TOOL_RESET_NAME
$MC_CUTTING_EDGE_RESET_VALUE

Bit 16=1:
The spindle last programmed with SETMS(x) remains the master spindle after end of program and reset, regardless of the machine data setting.
This means that if bits 0/6=1, the offset remains active for the tool which is placed in the spindle.
Power ON behavior
The machine data setting is effective after Power ON.
This means the offset for the tool which is placed in the spindles specified in MD $TC_SPIND_DEF_MASTER_SPIND becomes active; the offset value is that of

the smallest available D number No. for this tool.

Reset behavior for toolholder

Bit 17=0:
The master toolholder is the toolholder specified in MD $TC_TOOL_MANAGEMENT_TOOLHOLDER. The settings in the machine data below refer to this toolholder
$MC_TOOL_MANAGEMENT_MASK
$MC_RESET_MODE_MASK
$MC_START_MODE_MASK
$MC_TOOL_RESET_NAME
$MC_CUTTING_EDGE_RESET_VALUE

Bit 17=1:
The toolholder last programmed with SETMTH(x) remains the master toolholder after end of program and reset, regardless of the machine data setting.
This means that if bits 0/6=1, the offset remains active for the tool which is positioned in this toolholder.
Power ON behavior
The machine data setting is effective after Power ON.
This means the offset for the tool which is positioned in the toolholder specified in $MC_TOOL_MANAGEMENT_TOOLHOLDER becomes active; the offset value is that of the smallest available D number No. for this tool.

MD 22562: $MC_TOOL_CHANGE_ERROR_MODE

Bit 3=0:
Change command for a replacement tool is output.

Bit 3=1:
The disabled status of the spindle tool is ignored. The tool becomes active with the last programmed offset.

Bit 4=0:
Change command for a replacement tool is output.

Bit 4=1:
The spindle tool is set down – “T0” is output.

MD 20122: $MC_TOOL_RESET_NAME

Identifier of tool to be loaded
This tool is either loaded when end of program is reached or at reset or Power ON if the associated setting is made in MD $MC_RESET_MODE_MASK, or with NC Start if the respective setting was made in MD $MC_START_MODE_MASK.
If there are no entries here ($MC_TOOL_RESET_NAME=“”) this corresponds to “T0”.

MD 20130: $MC_CUTTING_EDGE_RESET_VALUE

D number of tool which is to be loaded via $MC_TOOL_RESET_NAME.
This means the tool becomes active with the offset set here.
If no entries are made in this machine data, the behavior corresponds to “D0”.

MD 20124: $MC_TOOL_MANAGEMENT_TOOLHOLDER

Specifies whether a tool holder number or spindle number is to be specified to define the location of the tool to be loaded.

MD 20090: $MC_SPIND_DEF_MASTER_SPIND

Definition of the master spindle in the channel. The number of the spindle is set here.

MD 20310: $MC_TOOL_MANAGEMENT_MASK

Bit 14 is used to activate the reset and start behavior. If bit 14 is not set, the settings in machine data $MC_RESET_MODE_MASK and $MC_START_MODE_MASK which are specific to tool management have no meaning.

MD 20112: MC_START_MODE_MASK

Bit 6=0:
Keep the last programmed offset active.
If the tool is disabled on the spindle, bits 3 and 4 are also evaluated in MD $MC_TOOL_CHANGE_ERROR_MODE.

Bit 6=1:
Start behavior (tool and offset selection) according to MD $MC_TOOL_RESET_NAME and $MC_CUTTING_EDGE_RESET_VALUE

Start behavior for toolholder

Bit 16=0:
The offset that was last selected remains active.
It does not matter whether the offset was selected in the part program or via settings in MD $MC_RESET_MODE_MASK.
With software Version 6.3.19 and higher, the offset for the tool placed in the master toolholder that was last programmed can also be active (see $MC_RESET_MODE_MASK)

Bit 16=1: (SW 6.3.19 and higher)
The toolholder specified in MD $MC_Tool_Management_Toolholder becomes active.
This means an offset selection refers specifically to this toolholder.

Start behavior for spindles

Bit 17=0:
The offset that was last selected remains active.
It does not matter whether the offset was selected in the part program or via settings in MD $MC_RESET_MODE_MASK.
With software Version 6.3.19 and higher, the offset for the tool placed in the master spindle that was last programmed can also be active (see $MC_RESET_MODE_MASK)

Bit 17=1:
The spindle defined in MD 20090 $MC_SPIND_DEF_MASTER_SPIND becomes active.
This means an offset selection refers specifically to this spindle.

Notice

NCK Power ON/control ON also triggers start mode and a change command is generated.
In this case, the issued command must be acknowledged (even if negative) as otherwise NC Ready status cannot be achieved.

Example 1:

In this example, the tool on the spindle is to remain active after End of Program (M02/M30) and Reset.

MD set as follows: $MC_TOOL_CHANGE_MODE = 1
$MC_CUTTING_EDGE_DEFAULT = –2

The following settings need to be made:

$MC_TOOL_MANAGEMENT_MASK	**Bit 14=1**	for reset and start behavior to be active
$MC_RESET_MODE_MASK	**Bit 0=1**, **Bit 6=1**	for the tool offset to remain active
$MC_START_MODE_MASK	**Bit 6=0**	for the tool offset to remain active

<u>NC program</u>

```
%MPFxxx1
. . .
N110 T="MILLER_10"
N115 M06                  ; Tool "MILLER_10" is loaded at change
N120 G90 G00 D2 X...      ; Offset D2 becomes active
. . .
N850 M30                  ; The offset D2 remains active
```

At the next program start tool “MILLER_10” is active with offset D2.

```
%MPFxxx2
N10 G90 G00 Z100          ; This block is executed with offset D2
```

Example 2:

In this example, the spindle tool is to be set down again at End of Program (M02/M30) and Reset ("automatic TO").

MD set as follows: $MC_TOOL_CHANGE_MODE = 1
$MC_CUTTING_EDGE_DEFAULT = –2
One spindle

The following settings need to be made:

$MC_TOOL_MANAGEMENT_MASK **bit 14=1**		for reset and start behavior to be active
$MC_RESET_MODE_MASK	**Bit 0=1**	Reset behavior
	Bit 6=0	TOOL_RESET_NAME is effective and CUTTING_EDGE_RESET_VA-LUE
$MC_TOOL_RESET_NAME**=""**		Name of tool that was loaded with reset. If no name is entered here, this has the same meaning as TO
$MC_CUTTING_EDGE_RESET_VALUE**=0**		The above mentioned tool becomes active with this offset ("0" has the same meaning as D0)
$MC_START_MODE_MASK	**Bit 6=0**	for the tool offset to remain active In this example D0 remains "active"

Example 3:

In this example, a specific tool is to be loaded at NC Start, e.g. a probe.

MD set as follows: $MC_TOOL_CHANGE_MODE = 1
$MC_CUTTING_EDGE_DEFAULT = –2
One spindle

The following settings need to be made:

$MC_TOOL_MANAGEMENT_MASK **Bit 14=1**	for reset and start behavior to be active
$MC_START_MODE_MASK **Bit 6=1**	Start behavior TOOL_RESET_NAME and CUTTING_EDGE_RESET_VAL UE are effective
$MC_TOOL_RESET_NAME**="Probe_1"**	Name of tool that was loaded with reset/start. In the example here, this tool is "Probe_1"
$MC_CUTTING_EDGE_RESET_VALUE**=1**	The above mentioned tool becomes active with this offset, here D1
$MC_RESET_MODE_MASK **Bit 6=0**	Is not meaningful for this example

Example 4:

In this example, the tool on the masterspindel that was last programmed is to remain active following End of Program (M30/M02) and Reset.

MD set as follows: $MC_TOOL_CHANGE_MODE = 1
$MC_CUTTING_EDGE_DEFAULT = –2
Two spindles
$MC_SPIND_DEF_MASTERSPIND=1

The following settings need to be made:

$MC_TOOL_MANAGEMENT_MASK	**Bit 14=1**	for reset and start behavior to be active
$MC_RESET_MODE_MASK	**Bit 0=1** **Bit 6=1**	for the tool offset to remain active
$MC_RESET_MODE_MASK	**Bit 16=1**	Keep the last programmed master spindle active
$MC_START_MODE_MASK	**Bit 6=0**	for the tool offset to remain active

<u>NC program</u>

```
N05 SETMS(1)           ; Spindle becomes master spindle (is also set via MD)
N10 T="TL1"
N15  M06               ; Change to Spindle 1
N20 G90 G00 D1 Z...
. . .
N80 SETMS(2)           ; Spindle 2 becomes master spindle
N85 T="TL2"
N90 M06                ; Change to Spindle 2
N95 G90 G00 D2 Z...
. . .
N230 M02               ; Active:  TL2 with offset D2 on spindle 2
                       ;          No offset is active on spindle 1, although
                       ;          "TL1" is positioned in spindle 1
```

Notice

If a change is triggered by Reset mode at Power ON, the NC remains idle with “No NC Ready” until an End acknowledgment has been received for this change.

3.2.23 Repeating a tool change with the same tool identifier

The behavior for repeated tool changes with identical tool identifier is influenced via MD 20310 $MC_TOOLMANAGEMENT_MASK. The default setting (bit 12=0) is selected so that the preparation command is not executed if the tool is already located in the spindle/toolholder.
Exception: Block search
Here the preparation command is always issued even if the tool is already positioned in the spindle.

With bit 12=1, the tool preparation command is also issued if the tool is located in the spindle/toolholder, however it is only issued one more time.

In the following examples, T is the tool preparation command and M6 is the tool change command.

New program for the tool that is still able to be used on the toolholder

```
N10 T = "TL1"    ;  Tool preparation command to PLC
N12 M06          ;  Tool change command to PLC
N20 T = "TL2"    ;  Tool preparation command to PLC
N30 T = "TL1"    ;  This tool preparation replaces the tool preparation from N20; tool
                 ;  management detects that a tool that can still be used from group
                 ;  "TL1" is loaded. This preparatory command is not output to the
                 ;  PLC.
N32 M6           ;  The tool preparation command from N30 was not output to the PLC
                 ;  and was deleted in NCK. The programming appears as if N10 –
                 ;  N12 – N32 were programmed. The tool change command N32 is
                 ;  also not output to the PLC.
```

New program for the tool that is still able to be used on the toolholder

```
N10 T = "TL1"
N12 M06
N20 T = "TL1"    ;  Preparatory command is output
N30 T = "TL1"    ;  No command output to the PLC
N32 M06          ;  Change and preparation are output together
```

N20, N30 and N32 are not output to the PLC.

New program for the tool that is no longer able to be used on the toolholder (time monitoring has for example assigned the status “blocked” to the tool)

N10 T = “TL1”	;	Tool preparation command to PLC
N12 M6	;	Tool change command to PLC
N20 T = “TL2”	;	Tool preparation command to PLC
N30 T = “TL1”	;	This tool preparation replaces the tool preparation from N20; tool
	;	management detects that a tool from group “TL1” is loaded but that
	;	the tool can no longer be used. A replacement tool is sought in the
	;	tool group and the tool preparation command is output to the PLC.
N32 M6	;	The tool change command N32 is output to the PLC.

Condition for processing a new tool preparation command in the NCK

N10 T = “TL1”		
N20 T = “TL2”	;	A command is only processed in the main run if the preceding
	;	command from the PLC was acknowledged with “End”.

With software Version 7 and higher, this rule only applies if N20 is output ot the PLC. Then the “End” acknowledgement must be present for a new tool preparation command to be output to the PLC.

Condition for processing a new tool change command in the NCK

N10 T = “TL1”		
N12 M06	;	A command is also processed in the main run if the preceding
	;	command from the PLC was not yet acknowledged with “End”.

3.3 Search for tool

The tool search is initiated by the preparation command (T selection). The search begins for a tool to load in the spindle.
Tool searches are generally performed on a magazine-specific basis, i.e. with the selected setting for the search strategy, the search is performed in the magazine from which the last change was carried out.

3.3.1 Strategies for tool searches

Tool search

The tools with the same identifier (name or Ident) but different duplo numbers are combined to form one tool group. The tool identifier is programmed in the part program with the NC address, i.e. only the tool group is specified during preparation.

In order to move a tool from a physical magazine to a spindle it must have the following characteristics:

- Tool status must be “enabled”
- Tool status may not be “disabled”
- Tool status may not be “currently being changed”
- Tool must not already be assigned a spindle other than the requesting spindle
- Tool must be present in the magazine location (except for manual tools)
- This magazine must be linked to the requesting spindle via a distance relationship ($TC_MDP2)
- This magazine must not have the status “disabled”.

The explicit tool is requested at the time of the tool call. The request is made for a special toolholder (general toolholder); this is the number of the address extension of T. At this point in time, user interface DB 72 is written for the relevant spindle and must be evaluated by the PLC application program.

The tool search strategy is defined with the system variable **$TC_MAMP2**. With **bit 0** to **bit 2** you select the conventional search strategies. These strategies start searching at the magazine from which the loaded tool was fetched previously.

Expanded tool search strategies

As in earlier versions, the search strategy is defined via system variable **$TC_MAMP2**. The older strategies are selected via bits 0, 0,1 and 2. Bits **3, 4** and **bit 5** provide additional functions.

By setting **bit 7** as well, you can start the search strategies defined by **bits 0, 1, 2** with the search as of the 1st magazine in the distance table (order in the distance table is defined via the programming order of $TC_MDP2). The standard setting is **bit 7=0**. The search starts in the magazine from which the last tool changed was taken.

Notice

Bit 3 = 1 to **bit 5 = 1** are only significant when the monitoring function is active (defined by $TC_TP9). Otherwise they have no effect on the suitability check.

Activation

The following conditions must be fulfilled for the tool search strategies:

- The tool-monitoring function must be active within tool management.
- The appropriate monitoring parameter values must be set for the cutting edges of the tools.
- The monitoring must be defined for the appropriate tool (system variable $TC_TP9).

Notice

If a monitoring type is activated for the tool with $TC_DP9, the current monitoring parameters are evaluated and, if necessary, the tool status set to ‘disabled’ or ‘prewarning limit reached’. An existing tool disable is not canceled, however, even if the monitoring function is deactivated for this tool.

3.3.2 Example of a tool search

Search routine tool search

A tool change at a spindle shall take place.

The search sequence for the correct tool is as follows:

1. The control checks whether the tool which is called is already located on the spindle.
2. If buffer locations are linked to the spindle (see $TC_MSLR), the control checks whether a suitable tool is already located in one of these.
3. The tool search starts in the 1st magazine of the distance table ($TC_MDP2) according to the selected search strategy.
(Applies only if bit 7 of $TC_MAMP2 = 1; otherwise, the search starts in the magazine from which the last loaded tool was fetched.)
4. If no tool is found in the first magazine, the distance table of the search is repeated in the next magazine.
5. If all the magazines that are linked to the spindle have been searched and no suitable tool found, the search is terminated with an alarm (22069 or 22068).

Any suitable tool with the programmed identifier found (not disabled) in one of the stages described above will be used.

3.3.3 Search in box magazines

Tool-search strategy for box magazines

The special tool-search strategy **"Shortest distance"** is available for box magazines. The search strategy is set in system variable **$TC**_MAMP2.

Definitions

The special tool search strategy **"Shortest distance"** is defined as follows for box magazines:

- Location number with the smallest absolute value of the difference to the current magazine position.

The term **"current magazine position"** is defined as:

- the location number from which the previously loaded tool was taken.

Requirements

The search strategy can only be used if the box magazine is assigned a “current magazine position”. This is set in system variable **$TC_MAP8**.

The NCK sets the current magazine position for box magazines. Since box-type magazines are not moved, the magazine position serves as the formal value needed for the tool-search strategy.

Example

The machine tool has a box-type magazine with 3x6 locations (=18). The current position **$TC_MAP8** is location 3. Suitable tools are stored in locations 9 and 18.

The search strategy detects the tool at location 9, because the absolute value of the difference is only 6, compared with the difference of 15 to location 18.

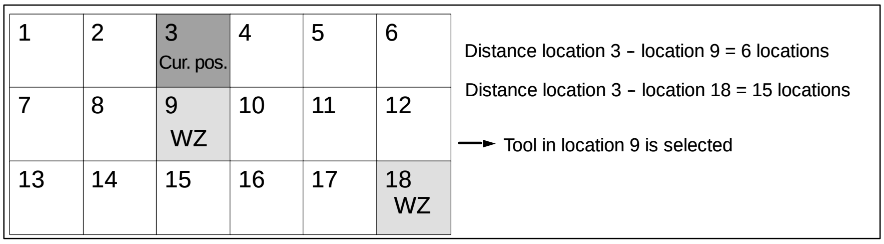


Bild 3-14 Search strategies in box magazine

3.4 Empty location search

3.4.1 Empty location search for a tool – from spindle to magazine

With the T preparation command, a matching empty location is automatically searched for the spindle tool. The location in which the new tool is stored is still occupied at this time and **cannot** therefore be identified as an empty location (see also “Replace tool search strategy”, Subsection 3.4.4)

Notice

Generally, a search is made for an empty location in that magazine from which the current tool in the toolholder was taken.

Fixed location coding

When searching for an empty location for fixed-location coded tools its previous location in the magazine is usually retained.

If the search for an empty location for a fixed-location-coded tool is started with a specific magazine number, that number is ignored. The old tool location is defined as an empty location.

If this number is however an internal magazine number (for a loading or tempora-rystorage magazine), then the number is explicitly taken into consideration and the fixed location coding is ignored. This case arises when loading/unloading tools.

If a location search for a fixed location coded tool is initiated using a specific magazine number and magazine location number, the fixed location coding is ignored and the specified location checked as a suitable location for the tool. This is used in the HMI function “Relocating”.

Variable location coding

Initially, the procedure for an empty location search is the same as that for a fixed location-coded tool. If this check fails, the search for a free location is continued. The search is performed according to the programmed search strategy ($TC_MAMP2). If the search cannot find an available location with the specified location type in this magazine, a new search operation based on the location type hierarchy (see Subsection 4.4) is started in the magazine. A location is only then considered as a suitable type of location when it applies that “Location type of the location” is larger than “Location type of the tool”, whereby the “larger than” relationship is defined by the location-type hierarchy. If no free location is found in this magazine, the search is continued in the next magazine (search strategy).

3.4.2 Search strategy for empty locations

Search strategy

The strategy can be defined with the magazine configuration according to which the search is made in chain magazines of TO units for a location not occupied. If chain magazines are not concerned here, then the search is executed according to the default strategy (forwards search starting at the first location number).

Possible strategies are listed in the table.

$TC_MAMP2	Search strategies	Meaning
Bit 8 = 1 256	Forwards search	The search takes place in ascending order.
Bit 9 = 1 512	Forwards search	The search takes place from the current location at the change position in ascending order.
Bit 10 = 1 1024	Reverse-order	The search takes place in the reverse order starting at the last location no.
Bit 11 = 1 2048	Reverse-order	The search takes place from the current location at the change position backwards.
Bit 12 = 1 4096	Symmetrical search	The search starts at the current location number at the change position (1st location left, 1st location right, 2nd location left, 2nd location right, etc.).

Definition of the current magazine position

The current magazine position in relation to the zero point is stored in magazine parameter (system variable) $TC_MAP8. The value is automatically updated by the PLC acknowledgement of a command initiated by tool management. If the magazine is moved without a task from the NCK, the user must adjust the actual position in $TC_MAP8 independently.

This can be done via a part program or by the PLC by writing $TC_MAP8 (selection via NC VAR selector block TM, variable magNoPlaces and assignment via PLC with FB 3, see Subsection 5.4.1).
Also via FC 8 with parameters TaskIdent = 4, TaskIdentNo = channel no., status = 5, OldToolMag = 9998, OldToolLoc = 1. The current position is parameterized (referred to spindle) in NewToolMag and NewToolLoc.

3.4.3 Empty location search criteria

Criteria for location search

- Location type must coincide with location type of tool. A hierarchy is taken into account.
- Check the tool size.
- Location must have the status “free”.
- Location must not be “disabled”.
- Magazine must not be “disabled”.

Magazine location type

The essential search criterion for the empty location search is the magazine location type. The magazine location type must match the magazine location type entered in the tool-specific data ($TC_TP). The magazine is searched. Each location is checked. If a suitable location is found the search is terminated.

If a matching location is found, then the check is made whether there is a magazine location type hierarchy for the magazine-location type that is entered in the tool. If there is none, the next magazine is taken if there are further magazines available. If there is a defined hierarchy, then the search routine is repeated starting at the magazine that has just been searched. If this search is also unsuccessful, the search moves to the next magazine, assuming another one is available.

Notice

With oversized tools, the location types of the adjacent location are not considered.

3.4.4 “Replace tool” search strategy (old for new)

With this search option, the magazine location of the “new” tool (tool to be loaded) is made available as the empty location for the ’old’ tool (tool to be unloaded).

It is not necessary for the “new” tool to be stored in the magazine location. It only needs to have been loaded (it may be located on a gripper, for example). If the location in question is not suitable for the “old” tool, then another appropriate empty location is sought.

Description of function

The new search for empty location is preset in the already existing bit-coded system variable **$TC_MAMP2** with **bit 13**.

Restrictions

With this empty-location search strategy, the NCK checks a magazine location that at the point of time of making the check is normally identified as still occupied by the “new” tool or is still “reserved for tool from buffer location”. This location is defined as an empty location for the “old” tool if the check gives a positive result.

The strategy cannot be applied if the new or old tool is coded to a fixed location because fixed-location coding takes precedence.

Notice

The PLC program has to execute the tool transportation operations in the correct sequence for the tool change:

- Remove “new” tool from the magazine location
- Bring the “old” tool to the magazine location

Otherwise damage may occur to the machine or tool.

The empty-location search strategy is only effective within tool changes program in the part program. The PI services (e.g. TMFDPL, TMFPBP) for the empty location search can make any use of this (see Subsection 5.12.5).

Example

This strategy is especially suitable for use with grippers and tools of the same type (same size and same location type or type that is compatible with defined location type hierarchy).

The already defined system variable $TC_MAMP2 includes an additional setting option for the new empty-location search strategy.

Bit	Value	Meaning
0		
. . .		Tool search strategy
7		
8		
. . .		Search strategy for empty location
13		The magazine location for the “new” tool is transferred to the “old” tool to be replaced and vice versa. Precondition is that the **tool sizes and location types of the tools match** or are compatible in terms of location hierarchy. The location of the “new” tool is detected as empty location for the “old” tool even if the “new” tool is still positioned at this location at the time the check is performed. The tool transportation must be designed such that the “new” tool is first removed from the magazine location before the “old” tool is taken to it. This order is vital to prevent damage to the machinery following completion of the mechanical tool transportation motions. The type of empty location search is determined via the bits 8 through 12. It is not possible for tool change to take place if the “old” tool does not have a magazine location assigned to it. The tool-search strategy is then determined via bits 8 to 12.

3.4.5 Tool search in wear group

If “Wear group” function is used:

In the case of existing tool-search strategies, the search refers only to the active wear group, i.e. only those tools are considered during a search within a tool group that are at magazine locations of the active wear group.

Tools in magazine locations with wear group number 0 are also checked for suitability.

If there are no spare tools available, then all $TC_MPP5 parameters of the current groups are negated and all locations are individually disabled by this. $TC_MAP9 is also negated (wear group disabled). All active tools are reset if this response has been configured via $TC_MAMP3 (bit 1 = 1).

The next wear group is called ($TC_MAP9 is assigned the number of the next wear group that can be activated).

If no further groups are available the search is terminated with an alarm. In such a case, the disabled tools should be replaced, if necessary. In order to enable the wear groups again, the wear group numbers of the magazine locations must be reset to values >0.

Search strategies

There are two search strategies for finding the next available wear group for activation:

- Starting from the lowest magazine location number, the replacement tools are searched through location by location according to the way they are sorted internally (time-optimized search). The wear group that is the subject of the search is found by searching for the first tool that is assigned to a wear group that can be activated.
- A search is made for the wear group with the lowest enabled wear group number (the first that can be activated).

Search in several magazines

The magazine definition for a machine defines whether the search is to be performed in one or several magazines.

If the search is conducted in several magazines while several wear groups are being used, always make sure that a wear group can only ever be assigned to one magazine.

The search is conducted acc. to the following priorities:

1. The search is performed in a magazine according to the configuration and strategy.
2. The search is performed in the active wear group.
3. The set tool-search strategy is taken into consideration.

Activation

In order to work with wear groups, the magazine locations must be assigned to wear groups via system variable **$TC_MPP5** and the function must be activated via the machine data.

In addition, the number of the wear group with which machining is to commence must be assigned to system variable **$TC_MAP9** of the magazine to be selected (value > 0).

For the configuration of the machine, it is defined by **$TC_MAMP3** how the tool status shall change when switching from one wear group to the next (defaulted is an unchanged tool status).

Example: Tool search in wear group

$TC_MAMP3 = 3 – Change "active" status of tools

Destination

- The tools must be set to "active" when a wear group is activated.
- When a wear group is disabled all the tools contained in that wear group are also to be disabled.

Inputs

- Circular magazine number 1 (6 locations)
- The magazine is to be divided into two parts:
  - Locations 2 and 3 from wear group 1.
  - Locations 4, 5, 6 and 1 form wear group 2.
- $TC_MAP9 = 1 (wear group 1 is "active")

Assignment to the wear group is achieved by:

```
$TC_MPP5[1,2]=1
$TC_MPP5[1,3]=1
$TC_MPP5[1,4]=2
$TC_MPP5[1,5]=2
$TC_MPP5[1,6]=2
$TC_MPP5[1,1]=2
```

The tools with T=10 and T=11 are assigned to wear group 1. As wear group 1 was activated, tools T=10, 11 were therefore also set to "active" (via $TC_MAMP3, bit 0=1).

Notice

Language command SETTA (see Subsection 5.8.20) can also be used to set the tools to active.

Tool assignment:

```
$TC_MPP6[1,2] = 10 ;T=10 has identifier "TL1"/duplo no.=1 TL status "active"
$TC_MPP6[1,3] = 11 ;T=11 has identifier "TL2"/duplo no.=1 TL status "active"
$TC_MPP6[1,4] = 12 ;T=12 has identifier "TL1"/duplo no.=2
$TC_MPP6[1,5] = 13 ;T=13 has identifier "TL2"/duplo no.=2
$TC_MPP6[1,6] = 14 ;T=14 has identifier "TL1"/duplo no.=3
$TC_MPP6[1,1] = 15 ;T=15 has identifier "TL2"/duplo no.=3
```

$TC_MAMP2 = 1

The active tool is to be searched for. If none is available, the next possible tool is to be located.
This tool-search strategy is superimposed by a check for the number of the active wear group. That is to say only those tools are considered during the search for a tool with the status “active” that are at magazine locations and that have the number of the currently activated wear group.

T=“TL2”

Tool group “TL2” consists of tools
T=11, 13, 15.

T=11 is positioned in a location of the active wear group (No. 1) and is “active”.
The result of the tool search is T=11.
Machining is continued. T=11 is disabled during machining.

T=“TL1”

Wear group 1 is still active. T=10 is identified as active and suitable.

T=“TL2”

The tool group of identifier “TL2” now has no active tool (has been disabled) and a new tool has not yet been set to “active”. This step is not taken until “TL2” is reprogrammed. The tools of the group are examined. In the locations of wear group 1, which is still active, there is no tool with identifier “TL2” or any other suitable tool.

This condition causes the next wear group (2) to be activated. Wear group 1 is now no longer the active wear group. The status of the tools in wear group 1 has been reset (not “active”), as configured by **$TC_MAMP3, bit 1=1**.

The tool search is now centered on wear group 2. Its tools were set to “active” when the wear group was activated (one tool from each tool group in the wear group because setting of $TC_MAMP3, bit 0=1).

The turret is now assigned as follows:

```
$TC_MPP6[1,2] = 10 ;T=10 has identifier “TL1”/duplo no.=1.
                   ;Tool status “not active”

$TC_MPP6[1,3] = 11 ;T=11 has identifier “TL2”/duplo no.=1.
                   ;Tool status “disabled”

$TC_MPP6[1,4] = 12 ;T=12 has identifier “TL1”/duplo no.=2,
                   ;tool status “active”

$TC_MPP6[1,5] = 13 ;T=13 has identifier “TL2”/duplo no.=2,
                   ;tool status “active”

$TC_MPP6[1,6] = 14 ;T=14 has identifier “TL1”/duplo no.=3 -
```

```
$TC_MPP6[1,1] = 15 ;T=15 has identifier “TL2”/duplo no.=3 -
```

In the example **T=13** is now taken as the next available tool “WZ2”.

Notice

The tool search only then generates an alarm when no further spare tool available in the tool group with the given identifier is found **and** no further wear group can be activated.

Control system response

Control behavior on Power On, Mode group change, Reset, Block search and REPOS is described below.

Configuration $TC_MAMP3, bit 0=1 (activate internally)

During Power On, the NCK checks whether the value of $TC_MAP9 is >0, i.e. whether a wear group has been selected. In this case the tools of that wear group are checked again and the value for $TC_MPP5 of each location in question is set to positive. In addition, the status of the tool in the location is set to “active”.

Configuration $TC_MAMP3, bit 1=1 (disable internally)

On Power On, the NCK checks whether $TC_MAP9 is negative, i.e. a wear group has been disabled. In this case the tools of the disabled wear group are checked again and the value for $TC_MPP5 of the location in question is set to negative. The “active” status of the tool in the location is reset.

3.5 Load

When a tool is loaded, it is taken to its magazine location and the associated data entered. Tools can be loaded via the spindle or a loading magazine.

With HMI Advanced, tool data can be transferred from the tool catalog, tool cabinet or via a code carrier system (see Subsections 2.8 and 3.13). The tool data can be entered directly into the magazine list with HMI Advanced.

- Manual loading only
- Empty location search
- Load tool at current location (location at the loading point/station)

3.5.1 Loading sequence

The loading operations supply the magazines with tools and write data to the relevant data areas of the TM system (magazine list with tool data, offset memory). Various methods of loading are available depending on the magazine configuration (loading magazine yes/no) and the data flow (when and from where are tool data written to the relevant data areas).

The loading method is mainly relevant to the HMI. As regards the tool management system on the NCK, only the result is important, e.g. that the tool is in the magazine and enabled for use after transfer of all its data.

References: /BAD/ Operator's Guide HMI Advanced

Loading is a channel specific operation that is possible when the part program is running. System variable $TC_MAP3 = 16 (enabled for loading) must be programmed if tools are to be loaded during part program runs.

There are two basic loading methods:

Free loading

With this method, the user can specify a magazine location to which the tool must be loaded.

Prompted loading

With prompted loading via the HMI, the location is determined by the tool management using an empty location search (see Subsection 3.4.2).

3.5.2 Tool data

HMI Advanced offers various options for loading and unloading the data of a tool and for storing the data.

These options can be used either individually or in parallel by the user.

When a tool is unloaded, the data can

- stay on the NCK (tool list)
- be written to code carrier (floppy, ext. hard disk, etc.)
- be stored in the tool cabinet (int. hard disk).

The tool data can be fetched again from these “data carriers” on loading. Tool data can also be entered directly by the user into the magazine list and/or the tool list.

Notice

The type of data backup can be defined by access rights in the PARAMTM.INI. file.

Master data can be stored in the tool catalog. Other enabled functions, such as interactive programming, can access tools which are defined here.

Selecting a tool for loading

- Select tool from the tool catalog (new tool)
- Select a tool from the tool cabinet (operating data)
- Enter tool data directly in the magazine list (HMI Advanced)
- Select a tool from the tool list (TO memory)
- Read in tool data via a code carrier system (see Subsection 4.5.2)

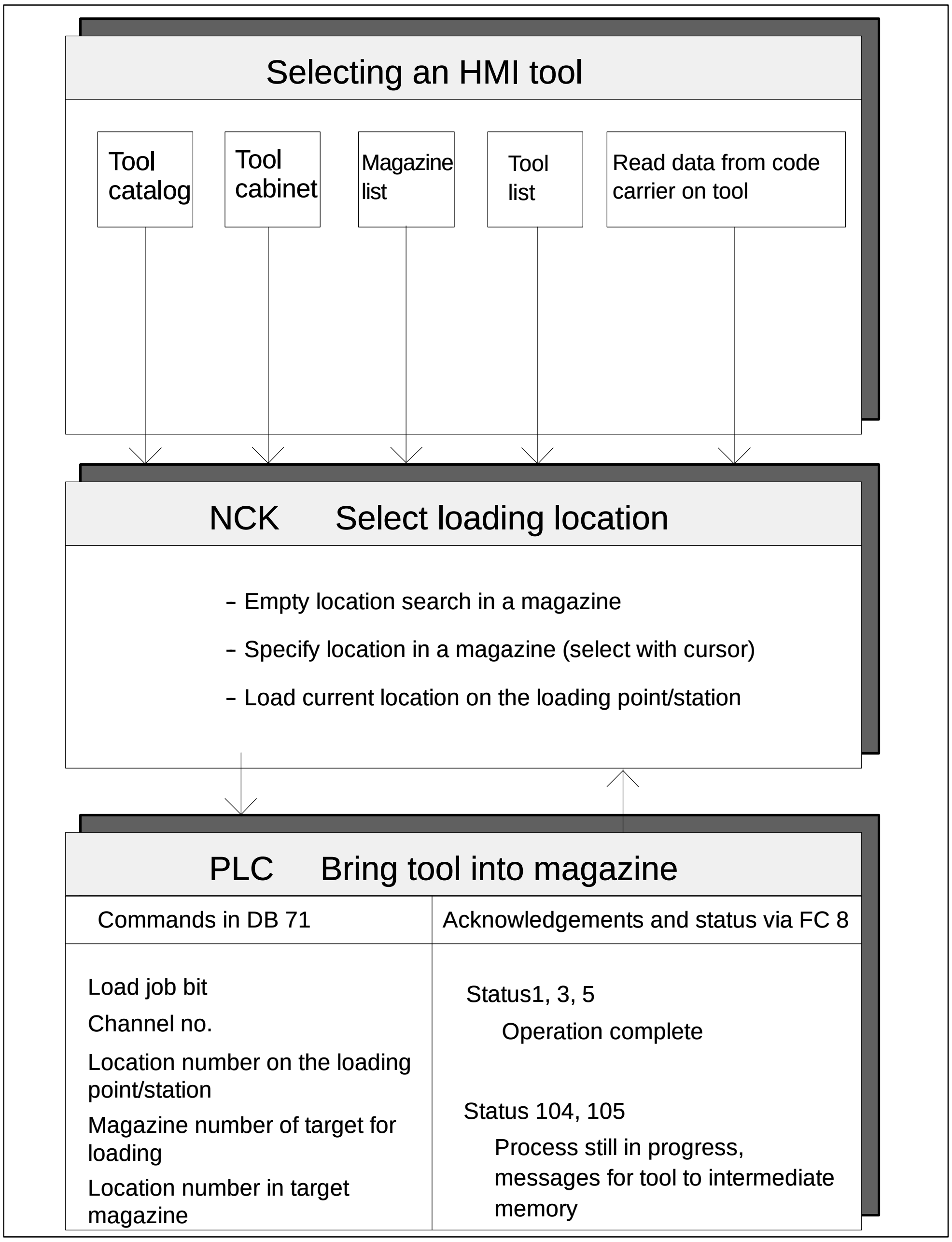


Bild 3-15 Loading-related functions of HMI, NCK and PLC

3.5.3 Select magazine location for loading

Find location in magazine

There are three possible ways of selecting an empty location:

- Initiate an empty location search (softkey)
- Input desired location number in magazine list (cursor)
- Move the desired empty location manually to the loading magazine and then load this location with softkey “Current location”.

3.5.4 PLC function at tool loading

Loading sequence

When loading from the NCK, the PLC is controlled by magazine and location numbers. It receives the request to move the magazine to the appropriate loading magazine for tool loading.

When a tool is loaded, the target address is the magazine and the loading location for the tool (DB71. DBW (n+24) and (n+26)). FC 8 receives this target address as parameters **“NewToolMag”** and **“NewToolLoc”** and “Status = 1” once the load operation has been successfully completed. Parameters “OldToolMag”, “OldToolLoc” must be set to zero. The number of the active interface identifies the loading magazine (location no.).

The loading procedure is performed as follows:

1. A request is sent to the PLC to load the tool. The information is transferred to the PLC in DB 71.

   Example:

   Data in DB71 when loading for the 2nd interface,
   (location 5 in magazine 1 is to be loaded from loading magazine 2)

   ```
   DB71.DBX0.1     = 1       ;Interface 2 active
   DB71.DBX34.0    = 1       ;Command: Load
   DB71.DBW50      = 9999    ;Magazine no. of loading magazine
   DB71.DBW52      = 2       ;Location no. of loading magazine
   DB71.DBW54      = 0       ;Magazine no. for unloading
   DB71.DBW56      = 0       ;Location no. for unloading
   DB71.DBW58      = 1       ;Target magazine no. for loading
   DB71.DBW60      = 5       ;Magazine no. of target for loading
   ```

2. The PLC must now move “location 5” from “magazine no. 1” (in which tool must be loaded) to “loading magazine 2” and execute the load operation.

3. When the tool is in the magazine, the user program must call FC 8. This notifies the tool management that the tool has been loaded.

Example of FC 8 call on loading

FC 8 parameters	Values	Comment
Start	1	Starts task
TaskIdent	1	DB 71 interface
TaskIdentNo	2	No. of active interface
NewToolMag	1	Mag. no. 1
NewToolLoc	5	Location no. 5
OldToolMag	0	During loading = 0
OldToolLoc	0	During loading = 0
Status	1	Operation complete
Ready		Checkback from FC 8
Error		Checkback from FC 8

Problems during loading

A tool cannot be loaded. Check the following:

- Is the location type correct?
- Is a suitable empty location available?
- Has the number of tools enabled in the NCK (MD 18082) been reached?
- Does the tool variable include a “0”, e.g. “1011”? (This is illegal.)

Alarms on the operator panel:

- No suitable empty location available
- “Create tools” command cannot be output to the NCK

3.5.5 Load tools via a part program

T number

The data required for a tool can also be loaded via a part program.

There are two possibilities to get the T number that addresses the data. You can:

- assign the T number yourself or
- allow the NC to assign the T number
  (via the command NEWT(...), see Subsection 5.8.8).

The other data can be addressed by the T number determined in this way. Otherwise the T number can be assigned by the user (refer to the following example):

Example

```
DEF INT TNo
TNo=NEWT("test",1)
$TC_TP3[4711]=2              ;  Size left
$TC_TP4[4711]=2              ;  Size right
$TC_TP5[4711]=1              ;  Size top
$TC_TP6[4711]=1              ;  Size bottom
$TC_TP7[4711]=2              ;  Location type
$TC_TP8[4711]=2              ;  Tool status
$TC_TP9[4711]=0              ;  Monitoring mode
$TC_TP10[4711]=0             ;  Substitute-tool strategy
$TC_TP11[4711]=0             ;  Tool info
$TC_DP1[4711] = 120          ;  Tool type:
                             ;  (all the compensation data is provided
                             ;  here)
$TC_MPP6[MagNr,PlatzNr]=4711 ;  Tool with T number 4711 is written/loaded
                             ;  to the location
```

The tool described here also occupies adjacent locations. These are automatically reserved for/assigned to the tool by the TM system (see Subsection 4.4.1).

It is also possible to delay assignment of a tool to a location, in which case the command $TC_MPP6 is not required. After execution of the part program the tools are contained in the tool list and can be loaded at a later time.

3.5.6 Retroload tool data

When tool data are “retroloaded” this means that the compensation data are not entered or loaded until after the tool loading operation.

Procedure

- The tools are already located in the magazine, both physically as well as their data, i.e. the “Tool <–> Location” assignment has taken place
- There is either no tool offset data in the NC or it is no longer up to date.

The offset data are assigned via the part program, i.e. the existing data are overwritten. If not already known, the internal T number of the particular tool first has to be determined in the “post-load” program to do this.
The internal T number is the tool number that the NC works with. It is unique and describes a tool. All parameters of this tool are addressed by this T number.

The T number can be assigned either by the operator during loading or by the NC.

If the operator knows the T number (e.g. specified by the entries made at the measuring station), then this number can be retrieved during the retroload program.

If the T number is not known, then it has to be determined for each tool to be retroloaded and be supplied from a variable. This reduces the overhead for the user and also reduces the scope for errors.

Create the retroload program

The tool is measured at a measuring station and the measured data stored. For this purpose, the tool must already be defined, i.e. by both an identifier ("Drill 12 mm" or "Mill 23" below) and the relevant duplo number. (The combination of tool identifier and duplo number uniquely defines the tool.) The internal T number of this tool is determined prior to the data block using the command GETT(...) and saved a a variable ("T no." here, see Subsection 5.8.10). The data required for the tool are written and then the entire program is transferred to the NC where it is processed.

Only the variables for which data are entered have to be written. The first tool in the next retroload program contains all the data, the second tool only contains the relevant data.

The T number does not have to be determined in the retroload program if already specified during loading because the data can then be assigned directly.

For a tool "1" with length L1, the program would be as follows:

```
$TC_DP1[1,1]=120;    ;Tool type

$TC_DP3[1,1]=4711;   ;Length1
```

Program for retroloading tool offset data

```
DEF INT Tno                  ;  Definition of variable TNo
tl1:
TNo=GETT ("Drill 12mm",1)
if TNo==1 goto tl2
$TC_DP1[TNo,1]=120           ;  Tool type
$TC_DP2[TNo,1]=0
$TC_DP3[TNo,1]=4711          ;  Length1
$TC_DP4[TNo,1]=0
$TC_DP5[TNo,1]=0
$TC_DP6[TNo,1]=24            ;  Radius
$TC_DP7[TNr,1]=0
$TC_DP8[TNr,1]=0
$TC_DP9[TNo,1]=0
$TC_DP10[TNo,1]=0
$TC_DP11[TNo,1]=0
$TC_DP12[TNo,1]=0
$TC_DP13[TNo,1]=0
$TC_DP14[TNo,1]=0
$TC_DP15[TNo,1]=0
```

```
$TC_DP16[TNo,1]=0
$TC_DP17[TNo,1]=0
$TC_DP18[TNo,1]=0
$TC_DP19[TNo,1]=0
$TC_DP20[TNo,1]=0
$TC_DP21[TNo,1]=0
$TC_DP22[TNo,1]=0
$TC_DP23[TNo,1]=0
$TC_DP24[TNo,1]=0
$TC_DP25[TNo,1]=0
$TC_MOP1[TNo,1]=0
$TC_MOP2[TNo,1]=0
$TC_MOP3[TNo,1]=0
$TC_MOP4[TNo,1]=0
tl2:                          ;  Next tool
TNo=GETT (“Mill23”,2)
if TNo==-1 goto error         ;  Possible error routine if tool does not
                              ;  exist
$TC_DP1[TNo,1]=120
$TC_DP3[TNo,1]=4712
$TC_DP6[TNo,1]=25
Error:                        ;  Error
:
:
M17
```

3.6 Unload

On unloading, the tool is removed from the magazine and the magazine list. You can:

- unload manually or
- unload at the current location (location at the loading point/station)

The unloading sequence is as follows:

1. Select a tool for unloading.
   To do this, position the cursor on the tool in the magazine list or, on HMI Advanced, on the tool in the tool list and select softkey “Unload”.
2. Select unloading point.
3. Transport the tool to the unloading point (by user PLC program).
4. Save or delete tool data.

References: /BAD/ Operator’s Guide HMI Advanced

3.6.1 Data backup during unloading

On unloading, the data for this tool is removed from the magazine list.

The following options are available for backing up the particular tool data:

1. Save tool data on code carrier
2. Save tool data in tool list (TO memory)
3. Back up particular tool data in tool cabinet

It is still possible to delete the tool data without saving them.

Notice

You can back up data on HMI Advanced in the following ways:

- From the tool list
- From the tool cabinet, or
- From the tool catalog

3.6.2 PLC function at tool unloading

During unload operations, FC 8 receives the identifier of the load/unloading point as the target address of the tool (DB71.DB(n+16) and DBW(n+18), basic address "n" is included in the interface list). FC8 receives this target address as parameters "**OldToolMag**", "**OldToolLoc**" and "Status" = 1 once the load operation has been successfully completed. The "NewToolMag" and "NewToolLoc" parameters must be assigned the value zero.

Unloading sequence

Unloading is controlled via DB 71. The unloading sequence is as follows:

1. The PLC receives the command to unload the selected tool. The information is transferred to the PLC in DB 71. Example of unloading data in DB 71 for the 2nd interface. Location 7 of magazine no. 1 must be unloaded at loading magazine 2.

   Example:
   ```
   DB71.DBX0.1= 1              ;Interface 2 active
   DB71.DBX34.1= 1             ;Command: Unload
   DB71.DBW50= 9999            ;Magazine no. of unloading point
   DB71.DBW52= 2  ;Location no. of unloading point
   DB71.DBW54= 1  ;Magazine no. for unloading
   DB71.DBW56= 7  ;Location no. for unloading
   DB71.DBW58= 0  ;Target magazine no. for loading
   DB71.DBW60= 0  ;Magazine no. of target for loading
   ```

2. The PLC must now move "Location 7" of "Magazine no. 1" (from which the tool must be unloaded) to "Loading/unloading point 2" and then unload the tool.
3. When the tool is from the magazine, the user program must call FC 8. This signals the tool management where the tool has been transported.

Example: FC 8 call on unloading

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	1	DB 71 interface
TaskIdentNo	2	No. of active interface
NewToolMag	0	During unloading = 0
NewToolLoc	0	During unloading = 0
OldToolMag	9999	Mag. No. 9999
OldToolLoc	2	Location no. 2
Status	1	Operation complete
Ready		Checkback from FC 8
Error		Checkback from FC 8

The PLC user program then has to traverse the magazine to the correct unloading point and execute unloading. If the tool comes via a buffer location (gripper, loader) to the unloading point or station, then the NCK is to be notified of each position change by means of the FC 8 with status 104, 105. Status “1” is not set via FC 8 until the tool is in the specified unloading point/station. The unloading operation is now complete.

Position for unloading (with OP030 and HMI Advanced)

When a magazine is being **positioned** at a loading magazine, the target address is stored in DB71.DBW(n+16) and DBW(n+18). This target address is passed to FC 8 as parameters “NewToolMag” and “NewToolLoc” and “Status” = 1 once the magazine has been successfully positioned. Parameters “OldToolMag”, “OldTool-Loc” must be set to 0.

The magazine and magazine location to be positioned are stored in DB71.DBW(n+20) and DBW(n+22). Positioning here only concerns magazine positioning a free location or a location with a tool to a loading/unloading station. The number of the active interface identifies the loading magazine (location no.).

Example: Position for unloading

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	1	DB 71 interface
TaskIdentNo	2	No. of active interface
NewToolMag	**9999**	**Mag. No. 9999**
NewToolLoc	**3**	**Location no. 3**
OldToolMag	0	During positioning = 0
OldToolLoc	0	During positioning = 0
Status	1	Operation complete

Notice

The function Positioning to unload can only be triggered from operator panel OP030 in SW 3.2.

3.7 Relocate, find and position tools

3.7.1 Relocate (task from TM system)

The target address for **relocation** is the magazine and location for the tool to be relocated (DB71.DBW(n+24) and DBW n+26). The tool source address is stored in DB71.DBW(n+20) and DBW(n+22). The target address is passed to FC 8 as parameters “NewToolMag” and “NewToolLoc” and status = 1 when relocation has been successfully completed. Parameters **“OldToolMag”** and **“OldToolLoc”** must be set to zero because the tool management recognizes the location of the old tool.

References: /BAD/ Operator’s Guide HMI Advanced

Example of relocating a tool

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	1	DB 71 interface
TaskIdentNo	1	Channel no. for tool management
NewToolMag	2	New magazine no.
NewToolLoc	17	New location no.
OldToolMag	**0**	**Old magazine number not used**
OldToolLoc	**0**	**Old location number not used**
Status	1	Operation complete

3.7.2 Relocation by the PLC

Task from PLC

A job can also be given to tool management by the PLC in order to reload a tool. This is done by notifying a new location for the tool to the tool management. The FC 8 block (TaskIdent := 4) is called with the following parameters:

- Old magazine no. (OldToolMag)
- Old location no. (OldToolLoc)
- New magazine no. (NewToolMag)
- New location no. (NewToolLoc)

1. Example

Relocation by PLC

The tool in magazine no. 1, location no. 5 is to be relocated to magazine no. 2, location no. 17. The PLC takes responsibility for ensuring that the location type is correct for the transfer. This example for calling a FC8 does not take any check-back signal to tool management for intermediate positions of tools into consideration.

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	4	Task from PLC
TaskIdentNo	1	Channel no. for tool management
NewToolMag	2	New magazine no.
NewToolLoc	17	New location no.
OldToolMag	1	Old magazine no.
OldToolLoc	5	Old location number
Status	1	Operation complete

2. Example

Relocation by PLC

Example: The tool is to be relocated from mag. no. 1, location no. 5 via grippers 3 and 4 to mag. no. 2, location no. 17.
FC 8 must be called up four times in this procedure. Only the important parameters are listed. All other parameters are as for the example above.

The tool is transported in four steps:

1. Move from magazine 1, location 5 to gripper 3 (location no. 4)

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	4	Task from PLC
TaskIdentNo	1	Channel no. for tool management
NewToolMag	9998	New magazine no.
NewToolLoc	4	New location no.
OldToolMag	1	Old magazine no.
OldToolLoc	5	Old location number
Status	1	Operation complete

2. Move from gripper 3 to transfer location 2, location no. 6

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	4	Task from PLC
TaskIdentNo	1	Channel no. for tool management
NewToolMag	9998	New magazine no.
NewToolLoc	6	New location no.
OldToolMag	9998	Old magazine no.
OldToolLoc	4	Old location number
Status	1	Operation complete

3. Move from transfer location 2, location no. 6 to gripper 4, location no. 5

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	4	Task from PLC
TaskIdentNo	1	Channel no. for tool management
NewToolMag	9998	New magazine no.
NewToolLoc	5	New location no.
OldToolMag	9998	Old magazine no.
OldToolLoc	6	Old location number
Status	1	Operation complete

4. Relocate from gripper 4, location no. 5 to magazine 2, location 17

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	4	Task from PLC
TaskIdentNo	1	Channel no. for tool management
NewToolMag	2	New magazine no.
NewToolLoc	17	New location no.
OldToolMag	9998	Old magazine no.
OldToolLoc	5	Old location number
Status	1	Operation complete

Relocation by PLC with location reservation TaskIdent 5

When a tool is transferred from a magazine location to a buffer with initiation from the PLC, it can be useful to reserve the magazine location.

You can do this with Task Ident 5.

The magazine location is now reserved when a tool is transferred to a buffer.

Notice

TaskIdent 5 may be programmed only for a tool transfer (magazine –> buffer location). Otherwise an error message is outputted, even though the tool is transferred.

Reservation “ Z ” is automatically reset when the tool is transferred from the buffer back to the magazine.

3.7.3 Find and position

With a find and position operation, a traversing task is sent to the PLC by the tool management. Bit 3 in DB71.DBB(n+0) “Position at loading magazine” is set. The magazine no. and the location no. (as destination) are transferred in the parameters DB71.DBW(n+20) and (n+22) during positioning.

The PLC then has to move this location to the loading magazine. The number of the loading magazine is entered in DB71.DBW (n+18) or determined by the number of the interface. If the PLC has moved the magazine location to the loading magazine, FC 8 must be called and the operation acknowledged with status 5 “Position changed”.

Example:

Location 5 in magazine 1 (source) must be moved to the loading magazine 2 (target).

DB71.DBX0.1	= 1	Interface 2 active	
DB71.DBX34.3	=1	Initiate positioning	(n+0)
DB71.DBW50	=9999	Magazine no. of the loading magazine	(n+16)
DB71.DBW52	=2	Location no. of loading magazine	(n+18)
DB71.DBW54	=1	No. of magazine to be positioned	(n+20)
DB71.DBW56	=5	No. of location to be positioned	(n+22)
DB71.DBW58		Magazine no. to which positioning is to be carried out	(n+24)
DB71.DBW60		Location no. to which positioning is to be carried out	(n+26)

Parameters “OldToolMag” and “OldToolLoc” in FC 8 are not required for positioning because only the PLC requires the information for traversing the magazine. The values for NewToolMag and NewToolLoc are from DB71 (n+24 and n+26). The PLC has to execute the positioning task and acknowledge it with an FC 8 call as follows:

Example of FC 8 call for positioning

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	1	DB 71 (load/unload, positioning, relocating)
TaskIdentNo	2	No. of active interface
NewToolMag	9999	Magazine no. 9999 (loading magazine)
NewToolLoc	2	Station 2
OldToolMag	0	During positioning = 0
OldToolLoc	0	During positioning = 0
Status	5	Operation complete
Ready		Checkback from FC 8
Error		Checkback from FC 8

3.8 Job processing of tools

Loading and unloading, as well as canceling and storing tools in the tool cabinet was only possible up to now via the Windows HMI user interface for one single tool per routine.

The new function “Job processing of tools” (“Batch”) allows the operator to enter all the specified operations for multiple tools at the same time in the job and then monitor how execution is progressing. The function “Reactivate tools” is also provided.

Filters are used in the selection of tools. They can be used to give an up-to-date picture of the tool data status for the NCK containing all the tools with the characteristics specified in the filter definition. This can be e.g. all tools having certain set tool-status bits, being of a certain tool type, having specific OEM data etc.

The search for the tools is carried out exclusively in the NCK.
The search is conducted by means of the OPI module TF (Parameterization, return parameters of _N_TMGETT, _N_TSEARCH) and the PI service _N_TSEARCH.

The job processing of tools can be initiated and monitored at the user interface. Loading, unloading and reactivation can take place in the background, even if the associated interface is not active.

References: /BAD/ Operator’s Guide HMI Advanced

Settings

Parameterization of Filter lists takes place in the file “paramtm.ini” in the section [BatchTools].

Country-specific sections are parameterized in the “language\patm_xx.ini” file in the [BatchTools] section; here xx stands for the two characters identifying the country, the file names is therefore, for example, PATM_GR.ini.

Notice

Any user-specific modifications should be made in the “user” directory files.

The ini. files are limited in size to about 63KB. Almost all commentary has therefore been removed from the file paramtm.ini. This is now to be found in the file paramtm.text.

Detailed information on the settings can be found in Subsection 4.4.1 and in Section 4.4.4.

3.9 Tool monitoring (workpiece count, tool life, wear)

3.9.1 Monitoring types

Number of workpieces

The workpiece counter must count all the tool cutting edges that are used to produce a workpiece. If the number changes, the monitoring data of all tool cutting edges involved must be updated. It should be remembered that the machine may have several spindles and that different tool cutting edges can be used simultaneously.

Tool life

Tool life monitoring is only performed on the tool edge that is currently in use. As soon as the path axes start traversing (except with G code G00), the tool life monitoring data is updated for the tools in the toolholder/spindle. If the tool life for a cutting edge expires during machining, the tool is blocked as a whole.

Wear

As is the case for the timer and workpiece count, the prerequisite for using the wear monitoring function is that tool monitoring is enabled in the machine data. In addition, wear monitoring must also be enabled in the respective machine data. The wear parameters of the cutting edge correspond to the local offsets (total offset parameters), see Subsection 3.10.4.

Tool life, workpiece count and wear

The monitoring type is defined for the tool when it is loaded. You can change the monitoring type at any time by changing the setting for system variable MD $TC_TP9.

The tool management performs monitoring for tool life, workpiece count with prewarning limit and degree of wear, or additive offset monitoring.

All types of monitoring can be active for different tools in operation simultaneously. If values have been entered for several types of monitoring, all monitoring counters are decremented.

The monitoring counter triggering the tool status change depends on the system variable $TC_TP9 (= type of monitoring):

- $TC_TP9 = 0 –> No monitoring
- $TC_TP9 = 1 –> Time-monitored tool
- $TC_TP9 = 2 –> Workpiece count-monitored tool
- $TC_TP9 = 4 –> Wear-monitored tool
- $TC_TP9 = 8 –> Additive offset

Several monitoring types can be activated simultaneously for one tool. Of the monitoring types, only wear monitoring and additive-offset monitoring are mutually exclusive.

If the monitoring criterion (tool life/workpiece count and wear) for a tool that is currently located in the spindle expires, it remains in use. Machining is not automatically interrupted to replace the tool with a fresh backup tool. The tool is not disabled until the next time it is selected. Since it is no longer “available”, a search is made for a replacement tool and the replacement tool is then loaded into the spindle. The tool change has to be organized by the PLC or come from the NC cycle.

The monitoring counters count from a set value > 0 down to zero. The limit value is reached when the counter has decremented to a value of ≤ 0. When a tool’s cutting edge (one of maximum 12 cutting edges) has reached its limit value, the whole tool is set to status “disabled”.

A **“G”** then appears next to the status for the tool in the magazine table.

Prewarning limit reached

If a cutting edge has reached its warning limit, then the whole tool is set to the status “Warning limit reached” (SLTD_SUSPENDED ($TC_TP8[i]=4)).

A **“V”** then appears next to the status for the tool in the magazine table.

At the same time a message is issued to inform the operator that a replacement tool may be required. If an operator action resets the monitoring counter from zero or the prewarning limit back to a value >0 or >prewarning limit, the tool status changes automatically to reflect the change in the data. This allows the operator to selectively suspend a “disabled” status caused by the tool expiring when its monitoring limit was reached.

If the tool has several cutting edges, all of the cutting edges must be beyond the monitoring limits.

Tool monitoring alarms

When the prewarning limit or the monitoring limit of a tool is reached, one of the alarms 6010, 6011, 6012, 6013 (abort clear acknowledgement condition) is output for information purposes.

With the NC command SETPIECE(...) (see Subsection 5.8.11) or PI command _N_TMPCIT (= other workpiece counters) it is possible for several tools to reach a limit value and therefore result in multiple alarms being issued.

No alarm is output if a limit value is reached as the result of data manipulation via Variable services.

Check monitoring status

A check can be made during program execution by issuing the programmed tool change command (e.g. “M06” for milling) without a T call to see whether life monitoring has expired for a tool. If it has, tool life management will search for a replacement tool and issued a request for tool change.

Enable memory and function

In general, in machine data

- MD 18080: MM_TOOL_MANAGEMENT_MASK and
- MD 20310: TOOL_MANAGEMENT_MASK

at least the bits 0 and 1 (3) must be set. This prepares the memory for the monitoring data and enables the function.

Enabling tool life monitoring

To implement tool life monitoring, the spindle (toolholder) or spindles which require this type of monitoring must also be specified in channel-specific MD 20320: TOOL_TIME_MONITOR_MASK. This machine data is bit-coded.

Example: MD 20320: TOOL_TIME_MONITOR_MASK

- Value = 1 Spindle number 1 only
- Value = 2 Spindle number 2 only
- Value = 3 Spindle numbers 1 and 2 only

Refer to Chapter 8 for machine data.

3.9.2 Tool life monitoring

Monitoring of tool cutting edge

Tool life monitoring is only performed on the tool edge that is currently in use. The spindle (toolholder) must have been activated beforehand
(MD 20320: TOOL_TIME_MONITOR_ MASK = spindle no.).

If MD 20124: TOOL_MANAGEMENT_TOOLHOLDER > 0, the toolholder number is selected in MD 20320 instead of the spindle number.

Tool life	The time is entered with 1 minute resolution up to SW 5.1 and can be entered on loading or set in the program with $TC_MOP2=500. The tool life is decremented internally in milliseconds and displayed in milliseconds. Data backup during unloading takes place in milliseconds with SW 5.1 and higher.
Inhibited	If the remaining tool life is ≤ 0, the tool is set to “disabled”. After the next tool change it is no longer used.
Monitoring from the NCK	The residual tool life is decreased whenever one of the 3 path axes is traversed at machining feedrate (e.g. G01). G00 traverse blocks are not “counted”.
Monitoring from PLC	The user can start and stop the time monitor using PLC signal “Time monitor active” (DB 21 DBX 1.3). The active control mode is set using the machine data 20310.
Prewarning limit	Input when tool is loaded or via part program with $TC_MOP1=50. When the prewarning limit has been reached, the tool is assigned the status “Prewarning limit reached” (display in the magazine list).
Special case, limit values	The tool life of a tool expires while it is in use. A check is made if this disabled tool is re-programmed by a change operation (e.g. M06 without T word), whether the monitoring time has already elapsed. If yes, a replacement tool is used.

$A-MONIFACT factor

By entering a channel-specific factor which is set before a tool is used for the first time, it is possible to monitor the different degrees of tool wear resulting from machining different types of workpiece material. The value is multiplied by the current time unit before the time value of the cutting edge is decremented. The write operation is performed synchronously with the main run. For more information please refer to Chapter 5.8.30.

Start and stop the tool life decrementation

Tool life monitoring runs when geometry axes are not traversed with **G00** (default setting).
The user can start and stop the time monitor using PLC signal “Time monitor active” (DB 21 DBX 1.3).

Which type of control is active is set via machine data: MD 20310: TOOL_MANAGEMENT_MASK bit 17. The default setting (bit 17=0) is standard; i.e. motion blocks not equal to G00 will make the time counter increment.

Time monitoring hierarchy

The combination of system variable $A_MONIFACT and function “Program testing active” produces the following nested time monitoring structure:

Machine data MD 20310: TOOL_MANAGEMENT_MASK defines the monitoring control by G00 or a PLC signal. Tools on spindles activated in machine data MD 20320: TOOL_TIME_MONITOR_MASK are time-monitored.

The VDI signal “Program test active” switches the momentarily valid time monitoring on or off; i.e. “Program test active” has a higher priority than the current time monitoring.

When time monitoring is active, the real time (as defined by the internal clock) is multiplied by the factor $A_MONIFACT and the result subtracted from the current time count of a tool edge mounted on the spindle.

3.9.3 Workpiece count monitoring

Changing the number of workpieces

The number of workpieces can be changed:

- Operation on HMI
- With a part program command (SETPIECE)
- PI service (TMPCIT) by PLC or HMI-OEM

Workpiece counter per spindle

Every spindle has a “memory” for the cutting edges used on it. With program command SETPIECE (1) the workpiece counter for the cutting edges that are used on the main spindle is decremented by 1. The workpiece counter of each spindle can be addressed individually.

The workpiece counter must count all the tools that are used to produce a workpiece. It will take into account that fact that the machine may have several spindles and that different tools can be used simultaneously.

If a tool is located on the main spindle with an offset number D > 0 during a count, this is stored in the "memory" when the next block is loaded during the main run, and then included in the next count.

The cutting edge of a tool is only counted once per spindle.

The part program author who programs **SETPIECE** can program the parameter as a function of the material.

SETPIECE (factor * no. of workpieces)

Like the factor for time monitoring, this function allows a workpiece count that depends on the process, the workpiece material or other factors.

Workpiece count can be deactivated via the channel DB DBX29.5.

Monitoring from the NCK	When the workpiece counter has reached the prewarning limit this is displayed in the magazine list. The tool is disabled when the workpiece count reaches zero. The next time the tool is called, the replacement tool is inserted.
Set workpiece counter	Entered during loading or via part program with e.g. $TC_MOP4=500.
Decrement number of workpieces	The number of workpieces must be decremented at the relevant point in the part program with the NC command SETPIECE (x, y) (e.g. SETPIECE(1) –> workpiece counter for main spindle tools is decremented by 1). The function for updating the quantity is activated from within die PLC program by a PI command.
Inhibited	When the workpiece count has reached zero the tool is disabled.
Prewarning limit	Entered during loading or via part program with e.g. $TC_MOP3=50. When the prewarning limit has been reached, the tool is assigned the status "Prewarning limit reached" (display in the magazine list).
Special case, limit values	It is not possible to activate a workpiece count for any number of cutting edges simultaneously! If the monitoring function has been enabled and activated by machine data, then all spindles can be monitored together at a time = "Number of cutting edges in the TO area" (= MD) for the number of cutting edges. An edge of a tool is only counted once per spindle.

3.9.4 Wear monitoring

The wear monitoring function is available only if the “Tool monitoring” system has been enabled (via machine data, see Subsection 8.1.2).

The wear monitor must also be enabled via machine data (MD 18080: MM_TOOL_MANAGEMENT_MASK; bit 5).

Definition

$TC_TP9 = 4; Wear monitoring is active for the tool.

$TC_TP9 = 8 can be set to select the “Additive offset” monitoring function if this is required. For bit assignments, see Section 5.3.

$TC_TP9 = 4

The wear parameters for a tool edge are defined with system variables $TC_DP12, ..., $TC_DP20.
These are assigned directly to the edge geometry values TC_DP3, ... , $TC_DP11.

$TC_DP10 and $TC_DP11 describe “angles”. The other parameters stand for the tool edge lengths and radii.

Only these values are included in the monitoring, i.e. wear parameters $TC_DP19 and $TC_DP20, which are analogous to system variables $TC_DP10 and $TC_DP11, are not taken into account. For bit assignments, see Section 5.2.1.

Notice

Wear monitoring does not monitor every single value but rather only the largest absolute value of each of these maximum seven wear parameters ($TC_DP12, ..., $TC_DP18).

$TC_TP9 = 8

Parameters that are analogous to the cutting edge wear parameters (system variables) are the **additive offset parameters**.

Analogous to wear, the following system variables are monitored for the additional offsets that are dependent on the location of use (location-specific offsets) of the cutting edge:

- $TC_SCP12, ... $TC_SCP18
  first additive offset for the cutting edge (to the extent defined)
- $TC_SCP22, ... $TC_SCP28
  second additive offset for the cutting edge (to the extent defined) etc. for the other additive offsets for the cutting edge

---

Notice

Wear monitoring does not monitor every single value but rather only the largest absolute value of each of these maximum seven additive-offset parameters*number of defined additive offsets for the cutting edge ($TC_SCP12, ..., $TC_SCP18, $TC_SCP22, ..., $TC_SCP28, ...).

Most tool geometries are described by a subset of the named data records.

If a parameter is changed (written), the NCK then checks whether the new value is higher than any of the other parameters and, if necessary, this value is subtracted from the wear setpoint. The result is the new actual value for the wear.

Analogous to other monitoring variables, the actual wear runs from the positive setpoint towards zero.

Monitoring parameters (system variables)

- $TC_MOP15 Wear setpoint or additive offset value
- $TC_MOP5 Wear pre-warning limit or additive offset pre-warning limit
- $TC_MOP6 Wear value or additive-offset setpoint

The physical quantity of the new monitoring parameters is “Length”. The unit is the same as for the wear values.
Wear monitoring can be deactivated via the channel DB DBX29.6.

The signal only acts on changes in wear data that occur during execution of the NC program. The PLC signal is suppressed if these data change because of OPI (e.g. during HMI operation).

Example

Let us assume the parameters are set as follows and wear monitoring is active for the tool with T no.=3:

```
$TC_MOP5[3,1]    = 0.002      ;= wear pre-warning limit
$TC_MOP6[3,1]    = 0.003      ;= actual wear value
$TC_MOP15[3,1]   = 0.007      ;= wear setpoint
```

These have already been set

```
$TC_DP12[3,1]    = –0.004     ;= wear component 1
$TC_DP13[3,1]    = +0.00      ;= wear component 2
```

Wear component 3 is now set

```
$TC_DP14[3,1]                 ;= –0.006.
```

Thus the maximum absolute value is given for the wear components = 0.006.
The resulting new actual value is

$TC_MOP15[3,1] – 0.006 = 0.001 = $TC_MOP6[3,1].
The prewarning limit has been reached.

Note: The wear components can be negative or positive – or be mixture of each.

3.9.5 Signals to and from the PLC

Previously, an alarm message was output as soon as the prewarning limit or limit value was reached. Alarms **6410** and **6411** are output when the prewarning limit is reached and **6412** and **6413** when the limit value is reached. Alarms 6410 and 6412 are triggered via the OP interface and alarms 6411 and 6413 via the NC program. The alarm texts identify the affected tool via the tool ID, duplo number and D number.

The following information is returned to the channel interface for one DB1 cycle (internal T numbers):

- Prewarning limit reached
- Limit reached

A strobe signal is set for one PLC cycle (DB channel.DBB344) which indicates that new data is available.

VDI signal “Warning limit reached” channel DB.DBD348

If a tool reaches its prewarning limit with tool life, workpiece count or wear monitoring, the internal T No. of the tool is entered here and the associated strobe signal is set.

VDI signal “Limit value reached” channel DB.DBD352

If the tool life, workpiece count or wear value has expired for a monitored tool, the internal T No. of the tool is entered and the associated strobe signal is set.

Notice

If machining is being performed with tools that are monitored for their workpiece count, it is possible for several tools to reach their prewarning limit or limit value simultaneously (SETPIECE is programmed at the end of program).
In this scenario, only the T No. of the tool that was last programmed is output.

VDI signal “T number of new replacement tool” – channel DB.DBD356

If during the tool change where the status of a tool found in the tool search is **set** in the NCK to “active”, then this is evaluated as the “the first time the spare tool has been selected”.

If the magazine contains several replacement tools with the status “active”, then this signal is <u>not</u> set at transition to a new replacement tool.

This process state change is output to the PLC via the T number of the replacement tool.

The action of the operator changing the tool status does not cause any change to the signal.

VDI signal “Last spare tool in the tool group” – channel DB.DBD360

If during the tool change where a tool is found during the tool search in the NCK and at this point ion time there are no further spare tools available for the programmed spindle/toolholder, then this is evaluated as the “Last tool found in the tool group”.

If there is only one tool (i.e. there is no replacement tool) it is also a tool group. When this tool is programmed, the interface signal is set immediately.

This process state change is output to the PLC via the T number of the replacement tool.

The action of the operator changing the tool status does not cause any change to the signal.

Notice

For tool groups containing many tools, the function increases the time required in the NCK for the main run when the tool is selected.

The following function must also be enabled
MD 20310: TOOL_MANAGEMENT_MASK. It is activated by setting **bit 18=1**.

Disable monitored tools – PLC-controlled by the VDI signal

In earlier versions, a tool has assumed the status “disabled” as soon as the actual value of the active monitoring function reaches the value zero. A tool in processing that is set to “disabled” remains in processing for such a time until the next tool change takes place. After that the tool can no longer be used.

The PLC can also determine when a disabled tool can no longer be used, i.e. when the “disabled” status is taken into account in the tool search.

- With the **VDI signal “Do not disable tool”** = 1
(channel DB. DBX29.7 = 1) the NCK does not take the tool status “disabled” into account during the tool search.
- With the **VDI signal “Do not disable tool”** = 0
(channel DB. DBX29.7 = 0) the NCK does take the tool status “disabled” into account during the tool search.

The bit is channel-specific.

“Search for active tool” strategy

This search strategy can ensure that a machining operation is not performed with different tools from the same tool group.

When the tool is disabled, a monitoring function and the set VDI signal “Do not disable tool” ensure that the status “active” is **not** canceled.

This **tool is therefore assigned** the status **“active”** and **“disabled”**.

If the required machining operation is terminated without a tool change, the status of all disabled tools must be checked. A new PI service (_N_TMRASS, in PLC TMRASS, see Subsection 5.12.5) is provided for this. This service can be used to cancel the status “active” for all tools that have been disabled (e.g. by the PLC program at the end of the program).

The other tool-search strategies

A disabled tool can still be used with the other tool-search strategies as well provided the VDI signal “Do not disable tool” (channel DB. DBX29.7 = 1) is set. The tool selected solely depends on the search strategy.

In the tool search, the **search strategy** therefore takes **precedence over** the **VDI signal “Tool disable invalidated”**. Both the last tool to be disabled or any other disabled tool can be selected.

Another tool which is not disabled might also exist, but is not selected because of the search strategy!

TO unit active in several channels

If a TO unit is assigned to several channels (tool and magazine data are “visible” in several channel), then the setting of the channel-specific VDI signal “Tool disable” is effective in each channel.

3.9.6 Monitoring data for setpoints

Previously, monitoring data has included the actual value and the prewarning limit for the variables to be monitored.

When the actual value reaches the value zero, the tool is disabled. Until now, the original starting value of the actual value has no longer been available in the NCK.

This value is available in the NCK, i.e. every monitored value receives a new data item – the setpoint. The setpoint is defined as a system variable and as an OPI variable (TS).

$TC_MOP11

$TC_MOP11 is the time setpoint
($TC_MOP1 = pre-warning limit for the time)

$TC_MOP13

$TC_MOP13 is the quantity setpoint
($TC_MOP3 = pre-warning limit for the quantity)

Reset to setpoints

Resetting the actual values of wear and additive offset "fine" means that all the parameters for wear offset and additive offset used for monitoring are set to zero.

Boundary conditions for tool monitoring

New system variables are being defined. This means that for the same number of cutting edges more non-volatile memory is used in the NCK than in SW 4.

The monitoring function "Wear monitoring" must be enabled via a machine data. The default value is "not active" so that no additional memory is needed (corresponds to more than 20 KB non-volatile memory for 1000 cutting edges).

Activation

The monitoring function must be enabled via machine data
MD 18080: MM_TOOL_MANAGEMENT_MASK.

Tools can be individually named for the different defined monitoring functions (time, workpiece count, wear, additive offset).

Wear monitoring is performed automatically by the NCK when the user changes the cutting edge offsets.

Control system response

Control behavior on Power On, Mode group change, Reset, Block search and REPOS is described below.

The VDI signal "Activate program test" has no effect on wear monitoring since new wear values are only entered during machining and not during the program test (provided the wear values are not changed by the machining program itself).

3.10 Variants of D number assignments

There are two possibilities to manage D numbers in the NCK:

3.10.1 Relative D no. for each T – standard

D numbers ranging from 1 to max. 12 are available for every T = “identifier” (with TM) or for every T number (without TM). These D numbers are assigned directly to the tool cutting edges.
An additive-offset block ($TC_DPx[t, d]) belongs to each D number = cutting number.
D0 is the offset deselection code.

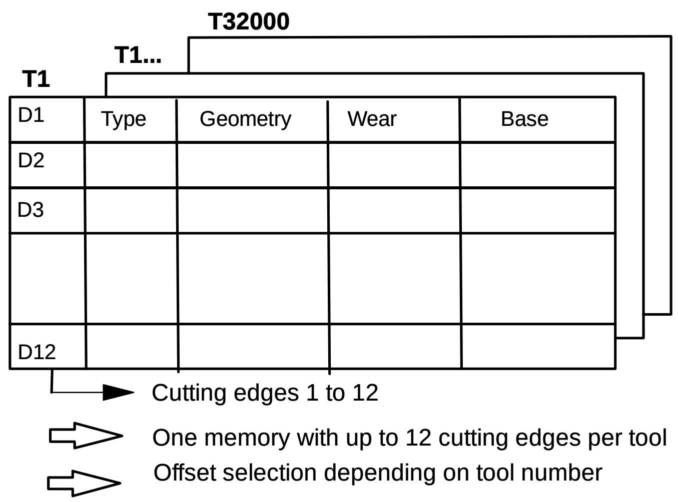


Bild 3-16 Layout of the tool offset memory

3.10.2 Absolute D no. without reference to the T number (Flat D no.)

Independence between D number and T number can be selected as an alternative to Subsection 3.10.1 in systems without tool management.
The reference of T number, cutting edge and offset by the D number is defined by the user.
The D numbers range from 1 to 32000. D0 is the offset deselection code.

Notice

The T number is always outputted to the PLC with an extended address (= spindle or toolholder no.) with this type of tool compensation.

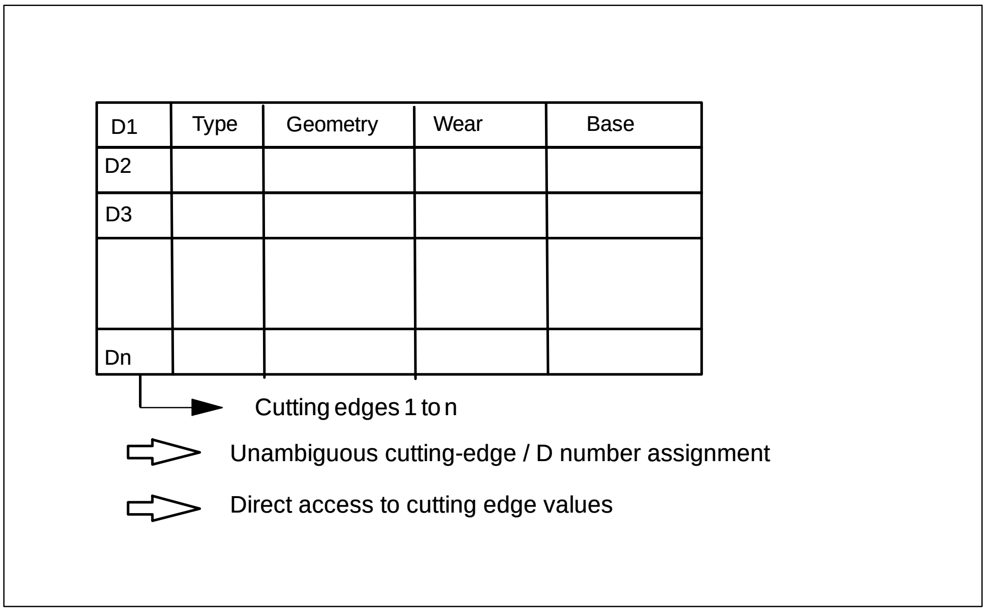


Bild 3-17 Layout of the tool offset memory

3.10.3 Free selection of D numbers for every T

D numbers can be freely assigned to tool edge numbers in systems with and without TM. As described in Subsection 3.10.1, a tool "T" can have a maximum of 12 edges. The upper limit for the D numbers used is limited by the machine data. This assignment option is an extension of the process described in Subsection 3.10.1.
With this setting, additional program commands can be used that make a check for unambiguous assignment of D numbers to T numbers or identifiers possible.
The same D numbers shall be assigned in each case for the cutting edges for duplo tools (same identifiers).

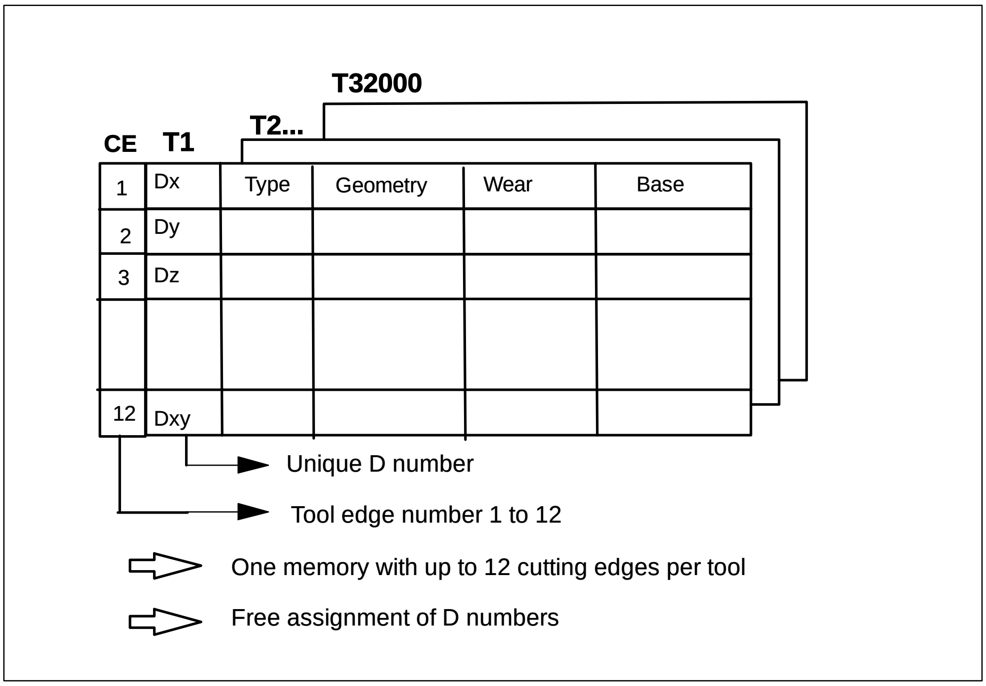


Bild 3-18 Layout of the tool offset memory

Notice

Universal system support (tool cabinet, code carrier) is not available for this function.

Machine data for available (unique) assignment of D numbers

$MN_MAX_CUTTING_EDGE_NO= Maximum permissible D number

Example:

$MN_MAX_CUTTING_EDGE_NO=1	A maximum of 1 offset (D1) can be defined per tool.
$MN_MAX_CUTTING_EDGE_NO=9999	Tools can be assigned unique D numbers here as follows: T1 with D1, D2, D3 T2 with D10, D20, D30 T3 with D100, D200, D300

$MN_MAX_CUTTING_EDGE_PER_TOOL= Assignment of tool edges per tool

Example:

$MN_MAX_CUTTING_EDGE_PER_TOOL=1	Only tools used with 1 cutting edge
$MN_MAX_CUTTING_EDGE_PER_TOOL=12	Up to 12 cutting edges per tool.

Unique use check (CHKDNO)

The NC command **CHKDNO** checks the D numbers assigned within the NCK for uniqueness. The D numbers of all tools defined within a TO unit may not occur more than once. This does not include replacement tools. See also Subsection 5.8.1.

Check within the magazine (CHKDM)

Exactly like **CHKNO**, the NC command CHKDM the D numbers assigned within the NCK for an activated tool management for uniqueness. This check function can be restricted to individual magazines. See also Subsection 5.8.2.

D number to T number (GETACTTD)

When tool management is active, the NC command **GETACTTD** can be used to search with the T number the D number of the tool active in the tool group. Prerequisite for this is that the D numbers have been uniquely assignment in the TO unit being considered. See also Subsection 5.8.3.

GETDNO, SETDNO during setup

The NC commands **GETDNO** and **SETDNO** permit reading and writing the offset number D for a given cutting-edge number CE.

GETDNO (T, CE): Read the D number for the cutting edge CE of the tool T

SETDNO (T, CE, D): Set the D number for the cutting edge CE of the tool T

$TC_DPCE[T, D]=...: Assignment of tool edge number CE to offset number D

Example:

Rename cutting edge CE=3 from D2 to **D17**

- In the following initial situation:
  Internal T number 1
  D number: 2
  Tool 1 cutting edge with:
  $TC_DP2[1, 2]=120 ;tool length T1, D2: 120mm
  $TC_DP3[1, 2]=5.5 ;tool radius T1, D2: 5.5mm
  $TC_DPCE[1, 2]=3 ;tool edge number T1, D2: 3
  (programming: T1,...D2)
- variable definition:
  DEF INT DNoOld, DNoNew=17
  DnOld=**GETDNO** (1, 3) ;value 2 is read in DnOld
  SETDNO (1 ,3, DNoNew) ;the new D no. is assigned to the cutting edge
- The new D value 17 is assigned to cutting edge CE=3
  $TC_DP2[1,**17**]=120
  $TC_DP3[1, **17**]=5.5
  $TC_DPCE[1, **17**]=3

3.10.4 Location-dependent offsets (additive offsets)

Local offsets are a generalized form of wear. They are part of the cutting edge data. The parameters of the sum offset refer to the geometrical data of a cutting edge.

Location-specific compensation can in general be used, i.e. with active/inactive tool management; with flat D-number function.

To meet the requirements of special machine operating modes, the relevant machine data can be set to divide the local offsets into the following categories:

- Local fine offsets
- Local coarse offsets = setup offset

The purpose of the setup offset is to allow the operator to set values prior to the machining operation. These values are stored in their own memory in the NCK, the operator can access the local fine offsets via the HMI. Local offsets “fine” and non-local offsets “coarse” are added NCK-internally and then act like the additive offset itself.

Several local offsets can be defined per D number. Machine data define the absolute number of local offsets, the maximum number of localoffsets per cutting edge and specify which additive offsets are active after the end of program or when the RESET key is pressed.

Applicable only when tool management is active:

Machine data 18104 can be set to define which additive offset must be operative if a tool is assigned the “active” status in the part program in the coarse of a programmed tool change:

- “Fine” tool offset values of tool cutting edges remain unchanged or
- “Fine” tool offset values of tool cutting edges are set to “0”.

The function is enabled by setting bit 8 = 1 in machine data $MN_MM_TOOL_MANAGEMENT_MASK.

DL-programming the additive/setup offset

Programming the additive offset is always relative to the active D number and is executed using the command

DL =“n”

The additive offset with the relative number “n” with respect to the active D number is activated by this. This means that the additive offset “n” is added to the wear of the active D number.

The additive offset is deselected with command

DL=0

Configuration of additive/setup offset

$MN_MM_KIND_OF_SUMCORR, bit 4=0

Corresponds to the default setting; only one data block of additive offset available per DL number. In this case, the term “additive offset” merely refers to the data represented by $TC_SCPx.

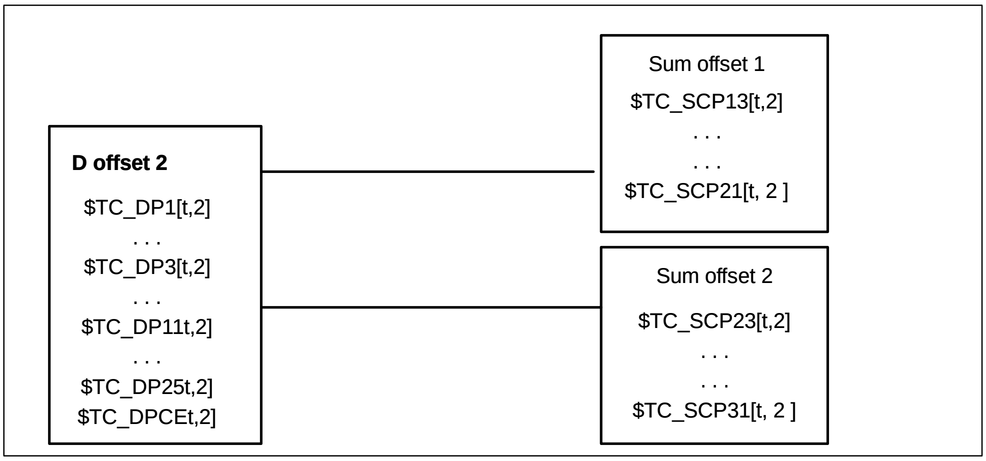


Bild 3-19 $MN_MM_KIND_OF_SUMCORR, bit 4=0

Let us assume the data from Fig. 3-19 for our program (and tool with T=t is active):

```
D2      ; Cutting edge offsets
        ; i.e. $TC_DP3,...$TC_DP11 + wear ($TC_DP12,...$DP29) +
        ; adapter dimensions
          . . .
DL=1    ; Additive offset 1 is added to the existing offsets of D2
        ; i.e. $TC_SCP13,...$TC_SCP21
          . . .
DL=2    ; Sum offset 1 is no longer added to offset D2, but additive offset
        ; 2 instead
        ; i.e.  $TC_SCP23,...$TC_SCP31
```

```
                . . .
DL=0          ; Deselection of sum offset; only the data of D2 are still effective
```

$MN_MMKIND_OF_SUMCORR, bit 4=1

Setup offsets are available. The general term “additive offset” refers to a combination of the “fine” additive offsets, represented by $TC_SCPx, and the additive offset, represented by $TC_ECPx. There are two data blocks for one DL number. The additive offset equals the product of the corresponding components $TC_SCPx + $TC_ECPx.

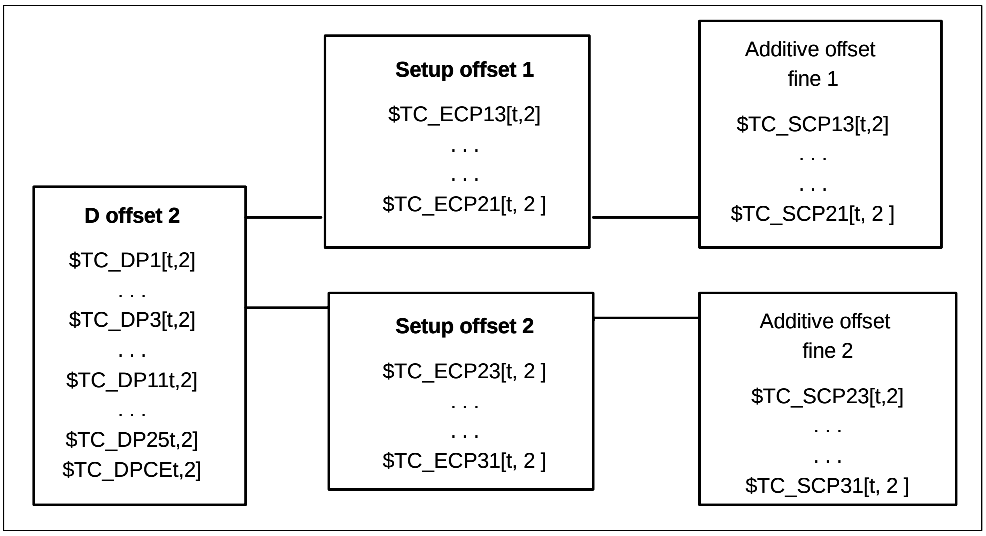


Bild 3-20 $MN_MM_KIND_OF_SUMCORR, bit 4=1

Let us assume the data from Fig. 3-20 for our program (and tool with T=t is active):

```
D2            ; Cutting edge offsets
              ; i.e. $TC_DP3,...$TC_DP11 + wear ($TC_DP12,...$DP29) +
              ; adapter dimensions
                . . .
DL=1          ; Additive offset 1 is added to the existing offsets of D2
              ; i.e. $TC_ECP13 + $TC_SCP13 ,...$TC_ECP21 + $TC_SCP21
              ;
                . . .
DL=2          ; Sum offset 1 is no longer added to offset D2, but additive offset
              ; 2 instead
              ; i.e. $TC_ECP23 + $TC_SCP23,... $TC_ECP31 + $TC_SCP31
                . . .
DL=0          ; Deselection of sum offset; only the data of D2 are still effective
```

The new NC command DELDL can be used to delete location-dependent offsets from cutting edges (see Subsection 5.8.7).

3.11 Adapter data

With the system variables **$TC_DP21, $TC_DP22** and **$TC_DP23**, the standard data block for the tool offset offers the option of entering the dimensions for an adapter (length1, length2 and length3).
This data is defined offset specifically.

Application

If tool management is active the additional adapter data can also be assigned to specific magazine locations.

This function is used for adapters that are fixed to a magazine location for a long period and used by different types of tool.

In individual cases, it is also possible to use identical adapters on several magazine locations. To do this it makes sense to define and store the adapter data records separately from the magazine locations.

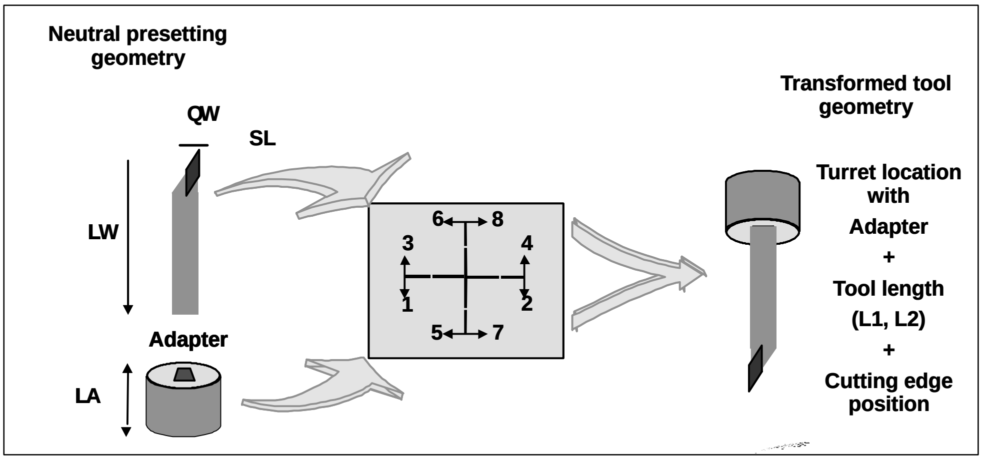


Bild 3-21 Adapter transformation

Adapter transformation

Adapter data “adapter transformation” allows fixed orientation of the tool on the adapter or orientation of the adapter including its tool with reference to the machine.

This function can be used as an alternative to the previous one. If adapter data are used, system variables $TC_DP21, $TC_DP22 and $TC_DP23 have a different reference and are therefore only formally part of the cutting edge data record in the NCK.

3.11.1 Description of function

The adapter data function must be enabled via machine data (MD18104: MN_MM_NUM_TOOL_ADAPTER).

For the setting to become effective, bit 7 must be set in MD 18080: MM_TOOL_MANAGEMENT_MASK.

Requirements

Two types of definition can be set in the machine data for adapter data:

- One adapter data record is assigned to each magazine location as standard.
- Adapter data records can be defined independently of magazine locations. The magazine locations are then assigned as an additional step.

The magazine location is the reference point for adapter **and** tool. Both are assigned to the magazine location.

The following elements are implemented when programming D numbers in the part program:

- The offset must be assigned to a real tool.
- The tool is assigned to a magazine location.
- It is possible to assign an adapter to the magazine location, for which a transformation (orientation) of the tool it contains can be defined.

Thus the working compensation can be clearly computed and the tool path accordingly adjusted.

If an additive offset is programmed, then the valu for this refers to the active D compensation.

3.11.2 Activation

Requirements

- In order to use the magazine-location-oriented data, machine data **MD 18104: MM_NUM_TOOL_ADAPTER** must be set to a value other than zero.
- Adapter data records must be defined.
- If the values of the machine data are > 0 the adapters must be linked to the magazine locations or assigned to them (can be automated via the HMI or using a cycle).

As a result, the adapter data including the defined transformations are always taken into account for the tool located on the magazine location inquestion. The work offset is calculated including the transformation and the adapter data.

The offset data can then be displayed as follows:

- Geometrical data for the tool (system variable $TC_DP3,...DP11); designated as neutral default geometry
- Non-transformed working compensation (sum of the values from tool geometry, wear, additive offset, tool base dimension or adapter)
- Non-transformed working compensation (transformation of the sum of the values from tool geometry, wear, additive offset) and tool base dimension of the adapter)

The quantities to be transformed can be selected via machine data. The mode of transformation of the additive offset can be set.

Magazine-location-related adapter data records

Create new

MM_NUM_TOOL_ADAPTER = –1:

One magazine location and one adapter data record are created. The specified values are put into the adapter data record which is automatically linked to the magazine location.

It is not possible to create a new free adapter at this point. The adapter numbers are assigned automatically (1 ... max. number of available magazine locations).

Delete

If an adapter data record is linked to a magazine location (MM_NUM_TOOL_ADAPTER = –1), it cannot be deleted.

Free adapter data records

Create new

MM_NUM_TOOL_ADAPTER > 0:

The adapter data can be created freely. Adapter data can be created by the user with a write operation to a non-existent data record.

$TC_ADPTi[n] = value;i = T, 1 2, 3, ..., n (number of the adapter)

If data record n does not yet exist and the maximum number of adapter data records that have already been defined is less than the value of MD 18104: **MM_NUM_TOOL_ADAPTER**, a new adapter data record is created with the specified values.

The value “value” is assigned to parameter i. Parameterizing rule: 0 < n ≤3 2000. The index value 0 is reserved.

Notice

The adapters must be assigned explicitly to the magazine locations if MM_NUM_TOOL_ADAPTER > 0.

Delete

If MD 18104: MM_NUM_TOOL_ADAPTER is set to a value of > 0, the adapter data can be deleted as required provided it is not assigned to a magazine location.

```
$TC_ADPTT[n] = -1
```

Adapter data record n is deleted and the memory becomes free again.

Deleting an assigned adapter data record:

The assignment to the magazine location must be undone first. You can only do this if the magazine location is empty. An alarm is issued if deletion fails.

Please proceed as follows:

- Remove the tool from the magazine location (unload, relocate).
- Remove the adapter from the magazine location.
- Delete the adapter data record (with $TC_ADPTTi[n] = –1).

Adapter data record n is deleted and the memory becomes free again.

Deleting all adapter data records

If **MM_NUM_TOOL_ADAPTER > 0** you can delete the adapter data if it is not assigned to a magazine location:

```
$TC_ADPTT[0]=–1
```

All non-assigned adapter data of the TO units are deleted. If you want to delete assigned adapters, you must first undo the assignment of those adapters to magazine locations. An alarm is issued if deletion fails.

Read/write adapter data

You can modify adapter data whenever you want to even if that adapter is assigned to a magazine location and/or a tool is located in the magazine location with the adapter.

Magazine location assignment/release

If **MM_NUM_TOOL_ADAPTER > 0** an adapter record must be assigned to a magazine location explicitly:

```
$TC_MPP7[m,p] = "adapter no."
```

Adapter number "adapter no." is assigned to magazine location p of magazine m. With "adapterno." = 0 any previous assignment is removed.

Notice

Assignment/decoupling is only possible if there is no tool in the magazine location.

Example of an adapter transformation

A turning tool with lengths L and Q is described below.

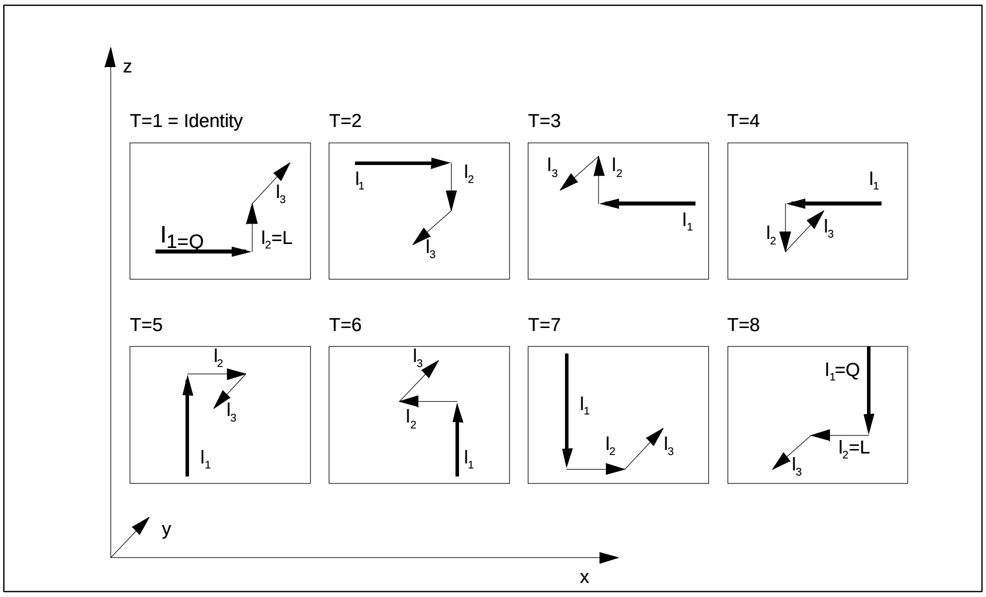


Bild 3-22 The 8 defined transformations (T = 1...8) for the adapter with G 18 and for a turning tool. The diagram shows the assignments of tool lengths l_1, l_2, l_3 to geometry axes x, y, z.

Transformations for numbers 1 to 8 are defined. Number 1 is the identity: no transformation of input data.

Other transformations can be implemented. The available transformations are designed initially for turning tools. These are typically defined by **Q=l_1=$TC_DP3** and **L=$l_2$=$TC_DP4**.

The transformation numbers correspond to the transformations shown in the table. In general:

Length1$_t$, length2$_t$, length3$_t$ = f(length1, length2, length3) = f(l_1,l_2,l_3) = f(Q,L,l_3)

Transformation number	Length1$_t$ length2$_t$ length 3$_t$ transformed values			Transformation with ref. to plane G18
1	$+l_1$	$+l_3$	$+l_2$	–
2	$+l_1$	$-l_3$	$-l_1$	180° about x
3	$-l_1$	$+l_3$	$-l_2$	180° about z
3	$-l_1$	$+l_3$	$-l_2$	180° about z
4	$-l_1$	$-l_3$	$+l_2$	180° about x, z
5	$+l_3$	$+l_1$	$-l_2$	90° about y, 180° about z
6	$+l_3$	$-l_1$	$+_{l2}$	90 about y
7	$-l_3$	$+l_1$	$+l_2$	– 90° about y
8	$-l_3$	$-l_1$	$-l_2$	– 90° about y, 180° about z

l_1, l_2 and l_3 are working offsets of the tool prior to transformation with or without adapter (depending on machine data settings). They are assigned to the geometry axes during compensation.

Notice

In turning, L and Q are also used to describe a tool. In the above table, l_1 corresponds, for example, to variable Q (or x direction) and l_2 to variable L (or z direction), assuming the plane G18 is selected (default setting for turning machines).

As standard, activation of an offset is calculated as follows:

Offset = D offset + x_i (e.g. wear, additive offset)

```
Length1  =    $TC_DP3 + xi
Length2  =    $TC_DP4 + xi+1
Length3  =    $TC_DP5 + xi+2
Radius1  =    $TC_DP6 + xi+3
```

The adapter transformation then acts on the transformed tool compensation values and is added to the transformed offset values.

The transformation number of the adapter causes a transformation of the tool (the cutting edges) located in this adapter (orientation according to the transformation number).

Working compensation = f(offset) + adapter dimensions of the magazine location

```
aLength1   =   Length1t + $TC_ADPT1
aLength2   =   Length2t + $TC_ADPT2
aLength3   =   Length3t + $TC_ADPT3
aRadius1   =   Radius1
```

Depending on the programmed plane selection G17, G18, G19, these values are added to the geometry axes.

G17, G18, G19 – plane selection (declarations)

The following agreements (different for machining and milling tools) apply for assigning tool-length parameters of the tools to the geometry axes:

Machining plane	System variables for tool length description		
	$TC_DP3($l_1$)	**$TC_DP4($l_2$)**	**$TC_DP5($l_3$)**
G17 Milling turning	Z Y	Y X	X Z
G18 Milling **Turning**	Y **X**	X **Z**	Z **Y**
G19 Milling turning	X Z	Z Y	Y X

Transformation of cutting edge position

The cutting edge position described by system variable $TC_DP2 is also transformed.

Transformations for the cutting edge position are performed as shown in the table below:

Transformation number	Cutting edge position								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	1	2	3	4	5	6	7	8	9
2	2	1	4	5	7	6	5	8	9
3	4	3	2	1	5	8	7	6	9
4	3	4	1	2	7	8	5	6	9
5	1	4	3	2	6	5	8	7	9
6	4	1	2	3	8	5	6	7	9
7	2	3	4	1	6	7	8	5	9
8	3	2	1	4	8	7	6	5	9

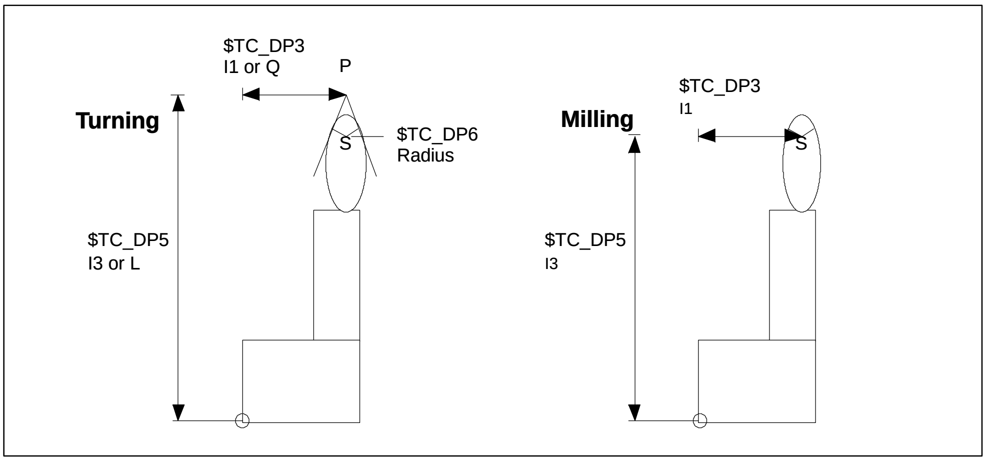


Bild 3-23 Turning and milling tools – relationship between cutting edge position and radius compensation

Turning tool geometries (I_1, I_3 or L, Q) are described in terms of P, the point of approach at the workpiece. However, the center point of cutting edge S with reference to the tool nose radius must be known for radius compensation.

This center point can only be accurately calculated if the tool point direction is known. Point S can then be derived from point P.

The position of the tool in the workpiece coordinate system is described via the cutting edge position (values 1 ... 8). Cutting edge position 9 corresponds to S = P.

Notice

The cutting edge position is only used for turning tools because their geometry is described with reference to P and not with reference to S as is the case for milling tools.

Adapter transformation for tools with three length components

The transformations defined here constitute a subset of all conceivable transformations. Only certain discrete values are considered here – in particular those that meet the requirements for turning tools (2 length components only).

System variables $TC_DP21 ... 23 and $TC_ADPT

If the function “Adapter” is active, then there are no further cutting-edge specific data for the “Base adapter dimension”.

In order to keep cycles that operate with adapter data compatible, the following rules apply:

If a tool is at a magazine location with an adapter and the adapter data can be accessed by system variable $TC_DP21...23, then the adapter parameters of the location can be read and written.

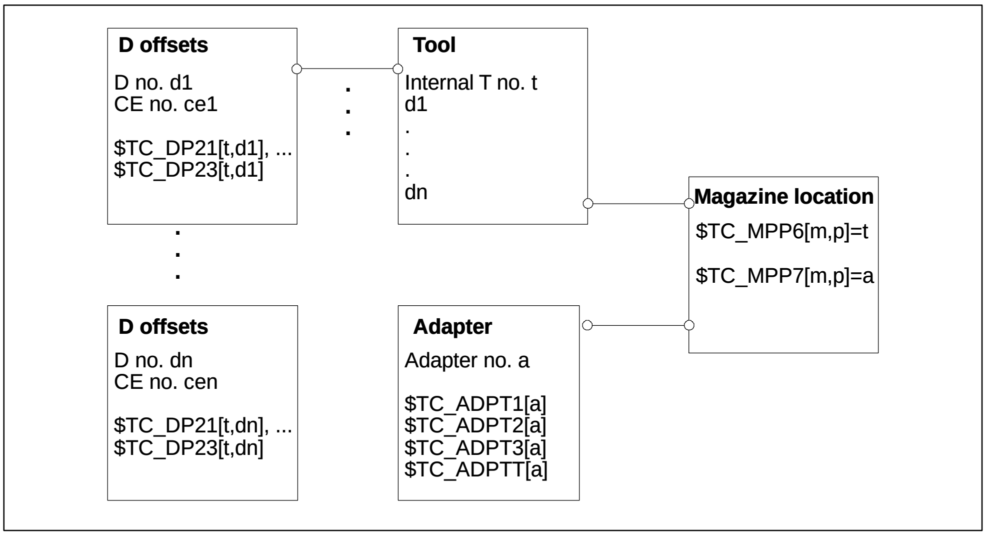


Bild 3-24 $TC_DP21, ...23 – Contents for an active "Adapter" function

Requirements:

- Tool t
- Magazine location p
- Magazine m
- Adapter a
- Tool with D offsets d_1, ... d_n

The adapter is assigned to the magazine location. If, for example, system variable $TC_DP21[t,$d_1$] is read or written in the part program, the programming accesses system variable $TC_ADPT1[a] of the adapter, i.e. the same machine data is accessed for all d_1, ... d_n.

If the assignment of the tool to the magazine location is released or the adapter is removed from the magazine location, no more data can be assigned to the parameters. A read operation returns the value 0, a write operation does not change the data (nor does it generate an alarm).

Transformed and non-transformed offset values

The values included in the path offset are usually the transformed work offsets.

It can generally be said that the data that describe a tool are subject to transformation. The transformation of the adapter is communicated to the tool (orientation in which it is positioned in the adapter). The adapter data itself is not transformed.

Data transfer to the NCK

You need to declare how the data is transferred to the NCK.

- You can transfer the data via the part program by programming the system variable $TC_...
  The parameters are defined as non-transformed values.
- The transfer can take place over the OPI interface using the Variables services. In this case, the data can be transferred either as transformed or non-transformed values.

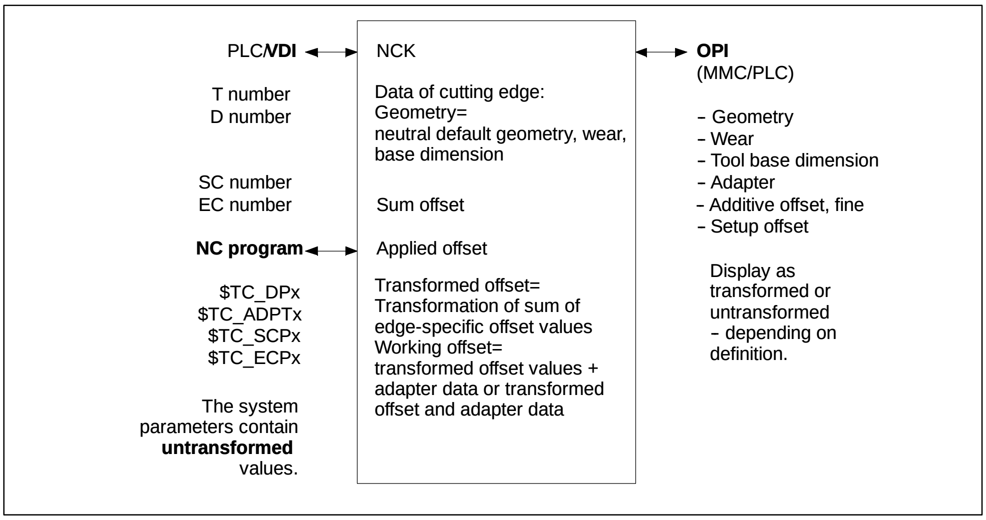


Bild 3-25 Geometry of a tool edge and applied offsets

Restrictions

When using the function (magazine-location-oriented) “adapter data” the user must ensure that the old data records of all the data records with edge-specific adapter data are adapted to the requirements of the new function.

However, using the described edge-specific adapter parameter definition (system variable $TC_DP21,...23), ensures that all old data is converted to the adapter data function by the NCK.

The function “Adapter data” excludes the existence of the cutting-edge specific data “base / adapter dimension”. This data is not meaningful if the adapter is defined specifically for the magazine location.

The function “adapter data” is better suited to the applications of an adapter because it defines the adapter as part of the magazine location and not as part of the tool or cutting edge.

Examples for assigning adapter data

Example 1

Requirements:

- MM_NUM_TOOL_ADAPTER = –1
- MM_NUM_MAGAZIN_LOCATION = 20
- One chain with 16 locations, magazine number = 1
- Two grippers
- One spindle
- One loading and unloading point
- Assignment

When creating the 20 locations in all, 20 adapters should be assigned, i.e. exactly one adapter assigned to each location.

Notice

It does not matter if the real locations are not actually fitted with an adapter. Preassigned adapter data have no effect on the offset. When equipping a location with a real adapter make sure that the appropriate values are assigned to the adapter data.

The transformation number of the adapter in location 3 of the chain magazine (No. 1) is to be changed to the new value 8:

```
$TC_ADPTT[$TC_MPP7[1,3]] = 8   ; $TC_MPP7 contains the number of the
                               ; adapter at the new magazine location
```

Once adapter data records have been automatically generated and assigned, operations such as undoing an assignment, renewed definition of an assignment and deletion of an adapter data record are possible.

Example 2

Requirements:

- MM_NUM_TOOL_ADAPTER = 4
- MM_NUM_MAGAZIN_LOCATION = 20
- One chain with 16 locations
- Two grippers
- One spindle
- One loading and unloading point

There are 4 different adapter geometries in this case. Adapters must be configured for the chain only.

Assignment

These locations (20 in total) are initially created without adapters. Locations 1 to 4 of the chain are equipped with adapters of the same geometry (here adapter 1).
4 chain locations are to be equipped with adapters with the same geometry.

First, you must define the 4 adapter data records. Now you assign them:

```
$TC_MPP7[1,1] = 1,          $TC_MPP7[1,13] = 4

$TC_MPP7[1,2] = 1,          $TC_MPP7[1,14] = 4

$TC_MPP7[1,3] = 1,          $TC_MPP7[1,15] = 4

$TC_MPP7[1,4] = 1,          $TC_MPP7[1,16] = 4

. . .
```

In this way you can assign one adapter data record to several magazine locations.

Notice

If you want to delete an adapter data record with a multiple assignment you must make sure that you first undo all the adapter assignments.

3.11.3 Transformed data of the active tool $P_ADT[n]

A new system variable is introduced that reads the compensation parameters of the active tool offset transformed according to the tool adapter $TC_DP1,... etc. Refer here to the system variable $P_AD that reads the non-transformed parameters. See Chapter 5.8.47.

$P_AD and $P_ADT have the same meaning without the function “Tool adapter” – as sub-function of the function TMMG. In other words the system variable application is only meaningful within the scope of the TMMG function.

With active function “Tool adapter”, the $P_ADT provides when reading the compensation parameters, transformed values of those parameters that are subject to the tool-adapter transformation in the event that the active tool is on a tool adapter at the point in time the parameters are read. The parameters not subjected to the transformation still provide the same values during reading as $P_AD.

When writing, the transformed parameter values subject to the tool-adapter transformation are accordingly transformed back by the NCK and are subsequently saved in a non-transformed form in the NCK. Non-transformed values are still written with $P_AD.

3.12 Power failure while tool command is in progress

If a power failure occurs during an action requested by tool management, defined strategies are executed by the PLC or special part programs, in order to establish a defined and consistent status on the machine and the tool management system. These strategies are machine-specific. SINUMERIK control systems thereby support the following measures:

Buffered data

The tool and magazine data are buffered. The tool attached to the spindle (= magazine location) is identified by the location and the tool block.
(This information is available even without tool management).

Control of data initialized via "Power ON"

The following data is set to zero:

- Tool status "Tool change in progress"
- Magazine status "Motion is active"
- Magazine location status "Reserved for tool to be loaded"
- The PI-command status with respect to magazine operations like e.g. "Motion is active".

Requirements of manufacturer configuration

The PLC must send the last unacknowledged FC 7 or FC 8 prior to power failure (READY did not change to TRUE before power off) back to the NC when the supply is restored. The function "Asynchronous transfer" is used for tool transfer in FC 8.

Without receiving the request from the tool management system, the PLC initiates a relocation of tool data from one location to another. For example, relocation of tool data from gripper to magazine if the tool needed to be returned manually to the magazine when the tool change operation was aborted.

Changes in position of the tools involved must be communicated via FC 8. The NC then updates the data for this tool in the tool management.

Further strategies may be necessary, e.g. if a tool change was interrupted. Tools stored in the buffer must be returned to the magazine for this purpose.

3.13 Code carrier

3.13.1 Function of the code carrier system

A link to a tool identification system is made available in the interactive tool loading and unloading dialog on the SINUMERIK 840D. Manual entry of tool data can thus be substituted by reading and writing the tool code carrier.

It shall thereby be noted that the particular tool data can only be saved on the code carrier **or** on HMI during unloading or deleting.

The data block of the tool is read from the code carrier by HMI during loading and sent to the NCK. Like tool selection from the tool catalog, the tool data can still be processed afterwards (compensation data, ...).

In an existing production line, tools with code carriers may already have been used. The data will be stored in a format suitable for a particular machine control system. When tools of this type are used in combination with a SINUMERIK 840D control, the data formats must be converted to allow the same tool to be used on machines operating with different control systems and thus different data formats.

In addition to the SINUMERIK 840D data, user data (Section 3.10 and Subsection 4.5.2) can be stored on the code carrier and operated via the loading/unloading dialog.

The “tool management data distributor” function block package is available for connecting code carriers via the PLC. Documentation describing the special settings required for each code carrier system is provided on the installation diskette.

3.14 Loading/unloading tools via PLC with PLC tool management data distributor

A tool management data distributor is available for connecting code carriers to the PLC. Refer to Catalog NC 60 for ordering.

3.15 User data

In addition to the data described here, machine manufacturers can utilize their own specific tool management data.

The new Siemens user data can be configured only by persons assigned OEM_HIGH rights and are not described here. The associated machine data is listed in Chapter 8, but not described in detail.

Tool and cutting edge data

During setup, the user can define additional tool and/or cutting edge data. Memory is allocated in the part program memory for this purpose.

The following machine data must be set:

- MD 18080: MM_TOOL_MANAGEMENT_MASK bit 2=1
- MD 20310: TOOL_MANAGEMENT_MASK bit 2=1
- MD 18094: MM_NUM_CC_TDA_PARAM (number of parameters)
- MD 18096: MM_NUM_CC_TOA_PARAM (number of parameters)

Notice

Without an enable in the machine data, no softkeys appear for cutting edge user data or tool user data.

Display screenforms

Depending on the amount of user data defined, screenforms are displayed in which users can enter custom data. This data is only maintained by the tool management and must be evaluated by the user in the part program (see also Section 5).

Defining name and unit

You can define names and units for this user data in file PARAMTM.INI (C:\USER\..) in areas [ToolParams] and [ToolEdgeParams]. The data only applies to the input and display on the HMI (see also Subsection 4.4.2).

Example of the use of user data**:**

- Maximum spindle speed
- Coolant yes/no
- Max. cutting rate

3.15.1 OEM parameters – extensions

Currently OEM parameters are defined for the following system variables:

Default System variable	Type	User System variable
$TC_TP[t]	Tool-specific	$TC_TP**C**[t] (see 5.3.3)
$TC_DP[t,d]	cutting-edge-specific	$TC_DP**C**[t,d] (see 5.2.2)
$TC_MOP[t,d]	monitoring-specific	$TC_MOP**C**[t,d] (see 5.2.4)
$TC_MAP[n]	magazine-specific	$TC_MAP**C**[n] (see 5.4.2)
$TC_MP[n,m]	magazine location-specific	$TC_MPP**C**[n,m] (see 5.4.4)

The C originally stood for compile cycle (users). Now, however, it generally means "user data".

With **software Version 6** and higher, machine data and system variables are included which allow definition of manufacturer-specific (Siemens) user data. The purpose of this new class of system variables is to define variables whose contents cannot be evaluated by the NCK itself but are still part of the system. Only control systems with the appropriate functions and characteristics will have these system variables. A additional difference between a Siemens OEM parameter and a system variable is that a specified, predefined meaning is assigned to system variables whereas Siemens OEM parameters can be assigned to a different meaning in different models and technologies.

Siemens user data

Default System variable	Type	User System variable
$TC_TP[t]	Tool-specific	$TC_TP**CS**[t]
$TC_DP[t,d]	cutting-edge-specific	$TC_DP**CS**[t,d]
$TC_MOP[t,d]	monitoring-specific	$TC_MOP**CS**[t,d]
$TC_MAP[n]	magazine-specific	$TC_MAP**CS**[n]
$TC_MP[n,m]	magazine location-specific	$TC_MPP**CS**[n,m]

For the previous block of machine data for activating user data

$MN_MM_**NUM**_CC_MAGAZINE_PARAM
$MN_MM_**NUM**_CC_MAGLOC_PARAM
$MN_MM_**NUM**_CC_TDA_PARAM
$MN_MM_**NUM**_CC_TOA_PARAM
$MN_MM_**NUM**_CC_MON_PARAM

There is a new block of machine data:

$MN_MM_**NUM**_CC**S**_MAGAZINE_PARAM
$MN_MM_**NUM**_CC**S**_MAGLOC_PARAM
$MN_MM_**NUM**_CC**S**_TDA_PARAM
$MN_MM_**NUM**_CC**S**_TOA_PARAM
$MN_MM_**NUM**_CC**S**_MON_PARAM

The meaning is analogous to the meaning of the respective machine data for the existing user data.

3.15.2 Assigning types to user data

Machine data

$MN_MM_**TYPE**_CC_MAGAZINE_PARAM
$MN_MM_**TYPE**_CC_MAGLOC_PARAM
$MN_MM_**TYPE**_CC_TDA_PARAM
$MN_MM_**TYPE**_CC_TOA_PARAM
$MN_MM_**TYPE**_CC_MON_PARAM

will allow the user to assign **types to user parameters**. Each machine data is an array with a preset size determined by the number of user parameters specified in machine data $MN_MM_NUM_CC_... .

Machine data

$MN_MM_**TYPE**_CC**S**_MAGAZINE_PARAM
$MN_MM_**TYPE**_CC**S**_MAGLOC_PARAM
$MN_MM_**TYPE**_CC**S**_TDA_PARAM
$MN_MM_**TYPE**_CC**S**_TOA_PARAM
$MN_MM_**TYPE**_CC**S**_MON_PARAM

allow the user to assign a type for **Siemens user parameters**. Each machine data is an array with a preset size determined by the number of user parameters specified in machine data $MN_MM_NUM_CCS_... .

The possible types that can be established correspond to a subset defined in the NC command:

Type of NC language	Value for the machine data
BOOL	1
CHAR	2
INT	3
REAL	4
STRING	5 Allows identifiers of up to 31 characters. The type can be assigned for one-dimensional parameters.
FRAME	Not defined
AXIS	Not defined

Examples

Let us assume that we are using 4 tool-related user data with types INT, REAL, STRING and BOOL.

Therefore we need to set bit 2 in the $MN_MM_TOOL_MANAGEMENT_MASK machine data in order to enable the OEM parameters function. Let us also set:

$MN_MM_**NUM**_CC_TDA_PARAM = 4

$MN_MM_**TYPE**_CC_TDA_PARAM[0] = 3
$MN_MM_**TYPE**_CC_TDA_PARAM[1] = 4
$MN_MM_**TYPE**_CC_TDA_PARAM[2] = 5
$MN_MM_**TYPE**_CC_TDA_PARAM[3] = 1

This allows us to use the selected user parameters as follows:

$TC_TPC1[4] = –45
$TC_TPC2[4] = 3.14
$TC_TPC3[4] = “Special tool”
$TC_TPC4[4] = TRUE

3.15.3 Custom user variables

Available with TMMG.

Additional data can be transferred to the PLC at tool change via user variables ($P_VDITCP[x]). This data can then be processed in the PLC program. The user variables must be programmed before the prepare change command T in the part program.

The data transfer to the PLC user interface DB 72 or DB 73 is implemented using the programmed tool change preparation command. Up to three user variables can be transferred simultaneously for each tool change. Data cannot be transferred from the PLC to NC by this method. The value format is DINT.

Software Version 6

As of software Version 6, these variables are also transferred by the change command M06 provided $MC_TOOL_CHANGE_MODE=1 has been set.

See also Subsection 5.7.

3.16 PLC description

3.16.1 Interfaces

The interfaces in the PLC consist of data blocks that are updated by the basic program. Tasks such as Load tool or Prepare tool change with source and target are stored for each tool are stored in the data blocks. Tool number (internal number assigned by the NCK when loading), tool size and tool status are also transferred at the interfaces for spindle or turret.

If the position of the tool changes (e.g. from magazine to gripper...), the new positions must be transferred to the tool management on the NCK. Two function blocks **FC 7** (TM_REV) and **FC 8** (TM_TRANS) are provided for this purpose. The PLC programmer can call these blocks and supply them with the required parameters.

If a magazine or a turret is not driven by an auxiliary axis, the shortest direction of rotation can be calculated with **FC 22** (TM_DIR) and the positioning time optimized. **FC 18** is available if positioning is performed using an auxiliary axis of the 840D.

Start-up of tool management function

Tool management in the PLC is set up by starting tool management in the HMI and activating the NCK option tool management. Before start-up of the PLC part of the tool management can be initiated, block FC 6 (part of the basic program) must be loaded in the PLC. The basic program calls this block; it does not need to be called in the user program as well. FC 8 TM_TRANS (transfer block) and FC 7 and, if necessary, FC 22 TM_DIR (direction selection) must also be loaded and called by the user program.

When installation and startup is complete, the next time the PLC is booted the following data blocks are set up for the user (user interfaces for tool management) in addition to a data block for the tool management FCs. The length of the data blocks are derived from the start-up parameters in tool management (see table below). The following data blocks are available:

Overview of data blocks

Block number	Length in bytes	Meaning
DB 71	4 + 30 bytes * B	Interface for loading/unloading points
DB 72	4 + 48 bytes * W	Interface for spindle as change position
DB 73	4 + 44 bytes * R	Interface for tool turrets as change position
DB 74	Length depends on configuration	Internal data block for tool management

B = Number of loading magazines

W = Number of spindles as change positions

R = Number of turrets

DB 71 to DB 74 occupy approximately 550 bytes for simple configurations of magazines, buffers and loading/unloading points.

Notice

If new PLC data has been “generated”, data blocks DB 71 to DB 74 must be deleted in the PLC and the PLC then cold restarted. The DBs are then set up for the new configuration.

There is one interface (data record) per data block for each loading/unloading point, spindle and turret. The data blocks are assigned to the different tasks (see Section 9).

DB 71

DB 71 assumes the functions of **loading and unloading, positioning** and **relocating**. The relocate and position at buffer functions are generally performed on the first interface in DB 71.

DB 72

DB 72 is the interface for changing tools into the spindle. This change procedure also includes preparation of the tool.

DB 73

DB 73 is the interface for tool changes with a circular magazine.

DB 74

Data block DB 74 is an internal tool management data block used for communication control. You must not write to this DB.

For all the interfaces listed here, source and target positions are available for the tools associated with the machining operation.

FC 6 is called in the basic program for communication between the NCK and the PLC when tool management is active. This block informs the user interfaces (DB 71 to DB 73) if a tool management function is activated via the part program or operator input.

Interfaces within DB 71 to DB 73

A bit field for the active and passive status of each interface is contained in bytes 0 and 1 of each of the data blocks (DB 71 to DB 73). DBX 0 represents the 1st interface, DBX 0.1 the second, etc. A total of 16 interfaces can be addressed. If one of these bits is set to the value = 1 by the tool management, the associated interface is activated. If set to 0, the interface may not be processed by the user.

Principle of interfaces DB 71–73

No. 8	No. 7	No. 6	No. 5	No. 4	No. 3	No. 2	No. 1
No. 16	No. 15	No. 14	No. 13	No. 12	No. 11	No. 10	No. 9
1. Interface							
2. Interface							
....							
15. Interface							
16. Interface							

If the value = 1, the user must evaluate the commands at this interface (see Chapter 9) and initiate the necessary actions (e.g. position magazines, change tools, etc.). Once these actions have been initiated the programmer can also write to this interface (e.g. to store the current positions of the tools involved in the action, or to enter status bits that he has assigned, or to cancel the bit “Prepare change”). Each modification of the tool positions and/or status information (see FC 8 for a description of the status information) via an interface task requires that FC 8 is called with these values.

Notice

Once FC 7/8 has been started, it can be reset after a READY signal or error signal.

Jobs from NCK tool management

Jobs	Interface	Acknowledgment	Applications, special features
Load Chapter 3.5	DB 71	FC 8, TaskIdent = 1, TaskIdentNo = interface no.	NewToolPlace = target position for tool in requested magazine location, state = 1, OldToolPlace = 0
Unload Chapter 3.6	DB 71	FC 8, TaskIdent = 1, TaskIdentNo = interface no.	OldToolPlace = target position for tool in requested loading magazine for unload, state = 1, OldToolPlace = 0
Relocate	DB 71	FC 8, TaskIdent = 1, TaskIdentNo = 1	NewToolPlace = target position tool in requested magazine location, status = 1, OldToolPlace = 0
Position	DB 71	FC 8, TaskIdent = 1, TaskIdentNo = interface no.	Positioning on loading magazine according to interface no. Optional positioning on interface 1, status = 5 NewToolPlace = LMG or BUF OldToolPlace = 0
Prepare change for tool in spindle	DB 72	FC 8, TaskIdent = 2, TaskIdentNo = interface no.	Position NewTL at change point, OldTL remains in spindle. Finally status 1 so that change command can be output. OldToolPlace = BUF (spindle) NewToolPlace = Location NewTL
Change in spindle	DB 72	FC 8, TaskIdent = 2 TaskIdentNo = interface no.	OldTL is unloaded (gripper or directly into magazine), NewTL is loaded to spindle. Status 1 required to ensure part program execution continues. NewToolPlace = BUF (spindle) OldToolPlace = Location OldTL
Without NCK Command: Return OldTL to magazine		FC 8, TakIdent = 4 TaskIdentNo = channel	The OldTL may need to be transferred asynchronously to the location specified in the Prepare Change command to move the tool from the gripper to the magazine.

Jobs	Interface	Acknowledgment	Applications, special features
Change with turret	DB 73	Normally FC 7, or FC 8, TaskIdent = 3, TaskIdentNo = turret no.	When turret has finished swiveling, FC 7 is called with turret no. as parameter ChgdRevNo.

LMG: Loading magazine
BUF: Buffer
TL: Tool
NewToolPlace: FC 8 parameter NewToolMag, NewToolLoc
OldToolPlace: FC 8 parameter OldToolMag, OldToolLoc

Changes to tool positions without job request from NCK

Jobs	Acknowledgment	Applications, special features
Asynchronous transfer	FC 8, TaskIdent = 4, TaskIdentNo = channel for this tool	Is required for notification of position, (e.g. for voltage interruptions, tool change return transportation, turret switching to manual mode)
Asynchronous transfer with location reservation for tool transportation in BUF	FC 8, TaskIdent = 5, TaskIdentNo = channel for this tool	
Asynchronous transfer without location reservation with manual movement of tool turret	FC 8, TaskIdent = 4	

Further interfaces in the channel interfaces for the ToolMAn function

DBD 348	T number for tool pre-warning limit
DBD 352	T number for tool limit value
DBD 356	T number of the new replacement tool
DBD 360	T number of the last replacement tool

See also Subsection 9.4.

Change bits in DBB 344

This information can be evaluated within one OB 1 cycle on the basis of a change bit. The PLC can use this information to derive appropriate action.

Other signals are as follows:

	Channel DB	
Tool missing	DBX 317.7	From NCK
Do not disable tool	DBX 29.7	To NCK
Deactivate wear monitoring	DBX 29.6	To NCK
De-activate workpiece counter	DBX 29.5	To NCK
Activate time monitor	DBX 1.3	To NCK

3.16.2 Definitions of acknowledgement status

Magazine identifier

The location of a tool in the magazine is shown by a magazine identifier and a location identifier. In a real magazine (chain, turret, etc.), the position of the tool is identified by the magazine number and the location within the magazine assigned during start-up.

If the tool is located in a buffer, the “magazine identifier” is the constant 9998 and the location identifier corresponds to the buffer number assigned during start-up.

In a loading magazine, the “magazine identifier” is the constant 9999 and the location identifier corresponds to the buffer number assigned during start-up. In this case, loading magazine number = 1 has a special status. Loading magazine = 1 (spindle) is for manual loading/unloading and also the interface for tool relocation.

Status value 1–10

The status information 1 to 10 (current upper limit 7) leads to the command being terminated. If one of these status data is passed to FC 8, the “active bit” of the interface defined in FC 8 is reset to “0”. thus completing the operation.

StatusValue > 100

When one of this status information data is transferred to FC 8, the “active” bit of the relevant interface remains at “1”, indicating the need for further processing by the user program in the PLC (e.g. continuation of magazine positioning). This item of status information is generally used to transfer changes in position of one or both tools while the operation is still in progress. For a list of the status information for block FC 8 see

References: /FB1/ P3, Basic PLC Program, Chapter 4

Synchronization

There are various methods by which the PLC and NCK can be synchronized (see Subsection 3.2.12). The two devices are forced to synchronize by bits 5, 6, 7 and 8 and, in SW 5.1 and later, bit 19 as well of machine data 20310: TOOL_MANAGEMENT_MASK. During internal communication between the PLC and NCK, the devices wait for each command to be acknowledged.

We distinguish between two types of acknowledgement:

- Transport acknowledgement
- End acknowledgement

Transport acknowledgement

Internal acknowledgement of a NCK command. The transport acknowledgement indicates to the NCK whether the issued command has been accepted by the basic PLC program. Before a new command is output, the system checks whether the previous command was accepted. If this is not the case, the output cannot take place. The NCK waits for the acknowledgement before a new command is output.

End acknowledgement

Status checkback signal from the PLC for an accepted NCK command. Error-free termination is indicated by status value = 1 and abnormal termination by status value = 3.

Output of the command

Synchronization of the NCK and PLC is implemented in three steps:

- The interpolation task from NCK has prepared a command on the basis of the programming and outputs this to the NCK-internal image of the VDI interface.
- The NCK-internal image of the VDI interface is transferred in the same cycle to the VDI.
- The basic PLC program accepts the command from the VDI interface.

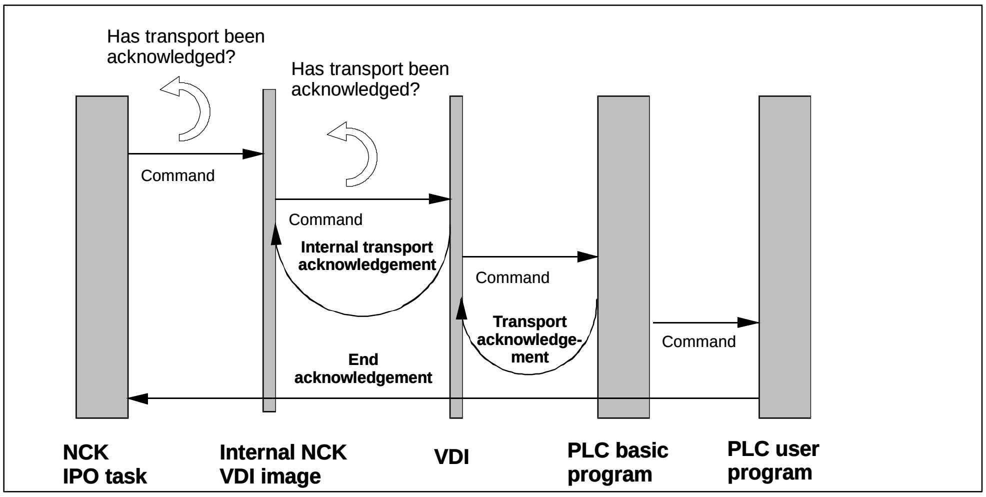


Bild 3-26 Transport and end acknowledgement

Acknowledgement of output commands

Acknowledgements from the basic PLC program and from the VDI are returned while the outputted command is being executed.

- The basic PLC program outputs the transport acknowledgement to the NCK once the command has been accepted.
- An internal transport acknowledgement is issued after the internal VDI image has been transmitted.

The PLC user program can only process one command at a time. It determines the command processing time. If the NCK provides the command faster than can be processed by the PLC user program, then the NCK is put into the waiting state.

The NCK can also output commands which do not originate from the part program over the interface. Included here are PI services that are asynchronously superim-posed over the part program processing.

Command termination

Depending on how bits 5–8 of MD 20310: TOOL_MANAGEMENT_MASK are set, the command output is considered completed at different points in time.

- If bit 5 (or bit 6 for the secondary spindle) of MD 20310 is set, the command output is completed when the internal transport acknowledgement and the transport acknowledgement have arrived. The command has been accepted by the basic PLC program.
- If bit 7 (or bit 8 for secondary spindle) of MD 20310: TOOL_MANAGE-MENT_MASK is set, this means that the command output is only completed when the end acknowledgement is received from the PLC.
- If the bits are not set, then the output of the command is considered as being completed when the NCK has output the command to the NCK-internal VDI image.

Notice

From the viewpoint of the tool-change command, the block change can take place as soon as the NCK has output the command.

Setting bit 19 in combination with bits 5–8 of MD 20310: TOOL_MANAGEMENT_MASK prevents blocks from changing before the required acknowledgements have been received.

3.16.3 Diagnostics for communication between NC and PLC

The NCK-PLC communication can be logged in a file as part of the tool change procedure.

Notice

The diagnostic data is saved when the Reset key is activated. In order for the data to be saved during program execution/without interrupting the program, software version 6 and higher also allows you to save data when Cancel is activated.

Requirements

- **Bit 13** must be set in machine data **MD 20310: TOOL_MANAGEMENT_MASK**.
- Free memory space must be available on the NC for saving the data. This applies both to the **SRAM** (passive file system) and to the **DRAM** – with approx. **4KB** each. The number of files in the file system must be below the maximum number of files.

Example of procedure

1. Start an NC program with the following history:
   No tool is present in spindle 1 (magazine no./location no. = 9998/3). Tool change is set with M06. A “milling” tool is present in magazine 2 at location 1 and has the internal T number 1.
   The following is programmed in the NC main program:
   ```
   T = milling tool
   ; Acknowledgement by PLC with FC 8 and Status 105 completed
   ; Acknowledgement by PLC with FC 8 and Status 1 completed (not shown)
   T = milling tool
   ; command with the same contents
   M06
   ; this command is not mentioned in the following recording
   ; Acknowledgement by PLC with FC 8 and Status 1 completed (not shown)
   TO M06
   ; Acknowledgement by PLC with FC 8 and Status 1
   M30
   ```
2. When the RESET key is activated, the recorded data which is stored in an internal circular buffer is copied to the file **_N_TCTRA’xx’_MPF,** with ’xx’= channel number 01, 02...; the file is created in the _N_MPF_DIR directory in the passive file system. In the current configuration, up to 25 communication procedures can be recorded. If more procedures are recorded in the circular buffer, the oldest data is overwritten. Up to 25 entries can also be made in the _N_TCTRAxx_MPF file. Any additional entries cause the file to be deleted and recording to start from the beginning again. This means that with longer diagnostic runs the only alternative is to save the current diagnostic log by pressing the Reset button once the program has finished.
3. Evaluation of the log file.
   In the file**_N_TCTRA01_MPF** a communication process is shown as follows:

- The command **from NC to the PLC**
  *T*00001 N:N10 CMD:00002
  NewTool: from M: 00002 P: 00001 to M: 09998 P: 00003
  TNo: 00001 Spindle : 00001
  OldTool: from M: 00000 P: 00000 to M: 00000 P: 00000
  Meaning:
  – T00001 = Number of communication process, in this case “1”
  – N:N10 = Block number in part program (if present), here N10
  – CMD:00002 = Command output by the NCK, in this case “2”
  – NewTool = The tool to be loaded

– OldTool = The tool to be removed (from the toolholder or a buffer location)
– TNo = The NCK-internal T number of the tool to be loaded at change
– Spindle = The spindle no. (toolholder no.) of the tool to be loaded at change
– M = Magazine number
– P = Magazine location number

i.e. a prepare command (CMD:00002) is output by the NCK in the above example. The new tool with T no.=1 is to be moved from location 2/1 to location 9998/3. There is no old tool. The magazine addresses for this are all zero.

- **Acknowledgement** of the NC command by the PLC
  T00002 N: ACK:00002 St: 00105
  NewTool: from M: 00002 P: 00001 to M: 09998 P: 00001
  OldTool: from M: 00000 P: 00000 to M: 00000 P: 00000
  Meaning:
  – ACK = Acknowledgement command from the PLC
  – St = Acknowledgement status from the PLC
- No output of commands with the same contents
  If it is set by the machine data that the NCK will not output successive commands with the same contents (dummy tool change, dummy tool preparation), then this is shown in the diagnostic log as follows:
  T00012 N:N20
  i.e., only the number and the block number are entered.
- **Result** of the above program (TO – M6 – M30)
  (contents of the log file):
  T00007N:N10 CMD:00005
  NewTool: from M: 00000 P: 00000 to M: 00000 P: 00000 TNo: 00000
  Spindle: 00001
  OldTool: from M: 09998 P: 00003 to M: 00002 P: 00001
  T00008 N: ACK:00005 St: 00001
  NewTool: from M: 00000 P: 00000 to M: 00000 P: 00000
  OldTool: from M: 09998 P: 00003 to M: 00002 P: 00001

Explanation:

- T00007 –> TO M6 results in command 00005
- No new tool is loaded for change; i.e. the addresses of the new tool are equal to zero; *TNo: 00000*
- There is one tool on the spindle with the address 9998/3. This tool is to be moved back into magazine 2/1.
- T00008 –> The PLC acknowledges the command with 5 and status = 1 and leaves the suggested motion tasks.

List of values and meanings for CMD and ACK

CMD	Explanation
1	A tool is transported from ... to ... . Load, unload, change, positioning
2	Tool change is to be prepared (setting MD 22550 = 1)
3	Tool change is to be carried out (setting MD 22550 = 1)
4	Tool change is to be prepared and carried out (setting MD 22550 = 0)
5	Tool change is to be prepared and carried out (setting MD 22550 = 1)

ACK	Explanation
1	Tool is/was transported. Load, unload, change, positioning FC 8 – Parameter TaskIdent = 1
2	Tool change is/was carried out (setting MD 22550 = 1) FC 8 – Parameter TaskIdent = 2
3	Tool change is/was executed (setting MD 22550 = 1) FC 8 – Parameter TaskIdent = 2
4	Tool is/was prepared (setting MD 22550 = 0) FC 8 – Parameter TaskIdent = 3
5	Tool change is/was prepared (setting MD 22550 = 1) FC 8 – Parameter TaskIdent = 2
7	Terminate canceled tool command DB10.DBX105.0=1
8	Tool was transported. If a tool is present at the source address, its data are transported to the target address. Otherwise, only the current magazine position is changed. If the tool transport is from a real magazine, the location to which the source address points is reserved. FC 8 – Parameter TaskIdent = 5
9	Tool was transported. If a tool is present at the source address, its data are transported to the target address. Otherwise, only the current magazine position is changed. FC 8 – Parameter TaskIdent = 4

3.16.4 Function blocks

Overview of function blocks

Block number	Meaning
FC 7	Transfer block for tool change with turret
FC 8	Transfer block for tool management, call at position and status changes
FC 22	Direction selection for shortest path

Transfer block FC 7, tool change with turret

For a description of the block, see
References: /FB1/ P3, Basic PLC Program

Transfer block FC 8

For a description of the block, see
References: /FB1/ P3, Basic PLC Program

Direction selection FC 22 TM_DIR

For a description of the block, see
References: /FB1/ P3, Basic PLC Program

Additional PLC services

In addition to the function blocks given above, there are further PLC services available for more complex requirements on the part of the PLC user program to influence tool management. These services are available in FB 2, FB 3 and FB 4, FB 7 (read and write variables or PI services). These FBs are described in Chapter 4 of the Basic PLC Program manual. The tool management PI services (program instances) are also described in Chapter 4 of the Basic PLC Program manual under the sections on FB 4 and FB 7. The tool management variables are described in the lists in the section on variables. (Please also refer to the Help section for the NC-Var selector.)

3.17 Shopfloor-oriented interface (ShopMill)

For detailed information, please see:

References: SINUMERIK 840D/810D
Operating/Programming ShopMill

3.18 Interface between Tool management HMI and WIZARD

The operator interface of the HMI tool management provides data in Ncdde variables that describe which object the operator is currently dealing with at the operator interface (e.g. the tool where the cursor is in a magazine-list view).

When these variables are written can be influenced by the settings in paratm.ini (see 4.4.2): either only when changing to Wizard screens or for ever toolmanagement status change.

TMHMI stands for Tool Management Human Interface

Name of the Ncdde variable TMHMICurData*MMCName*

With "*MMCNAME*" from mmc.ini, [GLOBAL], NcddeMmcName and NcddeMmc-Name may not take the standard value "_XXXX_" as this will otherwise be replaced by an arbitrary number.
The current TMHMI data are managed in these variables as a string, e.g. in the following form:

"curToolTNo=35;curToolIdent=Bohrer34;curToolDuplo=4;curMagNo=3;curMagPlaceNo=14;"
where "=" is used a separator between data name and data value
and ";" is used as a separator between data

Name of the N

Ncdde variable TMHMICurData*MMCNameDataName*

where "*MMCName*" from mmc.ini, [GLOBAL], NcddeMmcName; "DataName" (refer to the following list)

Notice

The Ncdde server does not allow multi-variable access to Ncdde variables. This means that write and read take place as separate access operations.

“DataName”

The following current data is available.

General data on TOA and channel:

– curTOANo	Current TOA number
– curChannelNo	Current channel number

Current tool in list views (the tool where the cursor is) and in tool-data displays for individual tools:

– curToolTNo	T number
– curToolIdent	Tool identifier
– curToolDuplo	Tool duplo number
– curToolType	Tool type
– curEdgeNo	Tool cutting-edge number, relative to the tool, <u>not</u> DNo
– curDLNo	Tool cutting-edge number, relative to the cutting edge
– curMagNo	Magazine number; “0”, if the current tool is neither located in the magazine nor intended for it
– curPlaceNo	Magazine location number; “0”, if the current tool is neither located on a location nor intended for it

Current magazine in magazine list views:

– curMagLiMagNo	Magazine number

Target magazine, target magazine locations when loading, unloading, relocating, positioning, searching for an empty location:

– targetMagNo	Magazine number
– targetPlaceNo	Magazin location number

Source magazine, source magazine locations when loading, unloading, relocating, positioning, searching for an empty location:

– sourceMagNo	Magazine number
– sourcePlaceNo	Magazin location number

Current tool in tool cabinet:

– curCabToolIdent	Tool identifier
– curCabToolDuplo	Tool duplo number
– curCabToolType	Tool type

Current tool in the tool catalog:

– curCatToolIdent	Tool identifier
– curCatToolDuplo	Tool duplo number
– curCatToolType	Tool type

Values that are currently unknown are shown as *“varname=;”* in the Ncdde variable or are not included there. The sequence of the data has not been specified.

As long a data word has not been set, its Ncdde variable is empty or not available.

The activity of the interface can be controlled by paramtm.ini in the section [General] (see Subsection 4.5.3).

Notice

The following setting must be the only line in paramtm.ini:

HMICurDataInterface = EnableAllTogetherWriteToNcdde := True,
EnableSingleWriteToNcdde := True,
WriteChangesWhenStateChanged := False

Installation and Start-Up

4

Start-up sequence for tool management

1. NC input of machine data (Section 4.1)
2. IPLC load the machine manufacturer PLC blocks (Section 4.2)
3. HMI Embedded – create magazine configuration (Section 4.3)
4. HMI Advanced – create magazine configuration (Section 4.4)
5. Additional settings (Section 4.5)
6. Panel operation (Section 4.6)

4.1 Input of the machine data

General machine data

Machine data for memory partitions, assignment of channels to TO units have to be set for tool management. Also, memory will be needed in the battery-backed RAM. When "memory-influencing" machine data is changed, i.e. at next Power ON, Restart or cold restart (reboot), this memory area is deleted and configured again. Therefore, data must be backed up prior to reset/cold restart.

Order for releasing memory using the machine data

Tool management option bit

MD 18080:	MM_TOOL_MANAGEMENT_MASK Activate the memory for tool management

Definition of number of magazines and magazine locations

MD 18084:	MM_NUM_TOOL_MAGAZINE Maximum number of magazines which NCK can manage (min. 3 magazines). Buffer location and loading magazine have to be added together!
MD 18086:	MM_NUM_MAGAZINE_LOCATION Number of magazine locations that NCK can manage. Add buffer locations and loading locations!

Definition of tools and tool edges

MD 18082:	MM_NUM_TOOL Number of tools to be managed by the NCK
MD 18100:	MM_NUM_CUTTING_EDGES_IN_TOA Number of cutting edges in NCK, tool offsets per TOA block
MD 18106:	MM_MAX_CUTTING_EDGE_PERTOOL Maximum number of cutting edges (D compensation) per tool (per T number)

Options for providing additional user data for magazines, magazine locations, tools and tool edges

MD 18090:	MM_NUM_CC_MAGAZINE_PARAM Number of additional magazine data $TC_MAPCx[n] generated
MD 18091:	MM_TYPE_CC_MAGAZINE_PARAM Type definition for magazine-oriented user data
MD 18092:	MM_NUM_CC_MAGLOC_PARAM Number of additional magazine location data $TC_MPPCx[n,m] generated
MD 18093:	MM_TYPE_CC_MAGLOC_PARAM Type definition for magazine location-oriented user data
MD 18094:	MM_NUM_CC_TDA_PARAM Number of additional tool-specific data per tool $TC_TPPCx[t] generated
MD 18095:	MM_TYPE_CC_TDA_PARAM Type definition for tool-oriented user data
MD 18096:	MM_NUM_CC_TOA_PARAM Number of additional data per tool edge $TC_DPCx[t,d] generated
MD 18097:	MM_TYPE_CC_TOA_PARAM Type definition for cutting edge-oriented user data
MD 18098:	MM_NUM_CC_MON_PARAM Number of additional monitoring data per tool edge $TC_MOPCx[t,d] generated
MD 18099:	MM_TYPE_CC_MON_PARAM Type definition for monitoring-oriented user data

Channelspecific machine data

Enabling of channel-specific functions for tool management

MD 20310:	TOOL_MANAGEMENT_MASK Channel-specific activation of tool management

Specification of spindle number for tool life monitoring

MD 20320:	TOOL_TIME_MONITOR_MASK Activation of tool life monitoring for the spindle specified here (toolholder number)

Tool change turret or spindle

MD 22550	TOOL_CHANGE_MODE New tool offset with M06 function
MD 22560	TOOL_CHANGE_M_MODE M06 function for tool change

Cutting edge selection after tool change

MD 20270	CUTTING_EDGE_DEFAULT Basic setting of tool cutting edge without program

Definition of tool with which tool offset is to be selected as a function of MD 20110 and MD 20112 during power-up and reset

MD 20122:	TOOL_RESET_NAME Definition for selection of tool length compensation

Definition of the active toolholder number

MD 20124:	TOOL_MANAGEMENT_TOOLHOLDER Definition of the active toolholder number

Assignment of TO units to channels

MD 28085:	MM_LINK_TOA_UNIT Assignment of a TO area to a channel (default = 1)

Definition of initial setting for control after boot, reset, end of part program in relation to G code, tool length compensation and transformation

MD 20110	RESET_MODE_MASK Definition of the control's basic setting. Relevant bit = 0: The current value remains valid.

Notice

In machine data 20310: TOOL_MANAGEMENT_MASK and 18080: MM_TOOL_MANAGEMENT_MASK, bits 0–3 must always be set to the same value.

4.2 Load the machine manufacturer PLC blocks

Overview

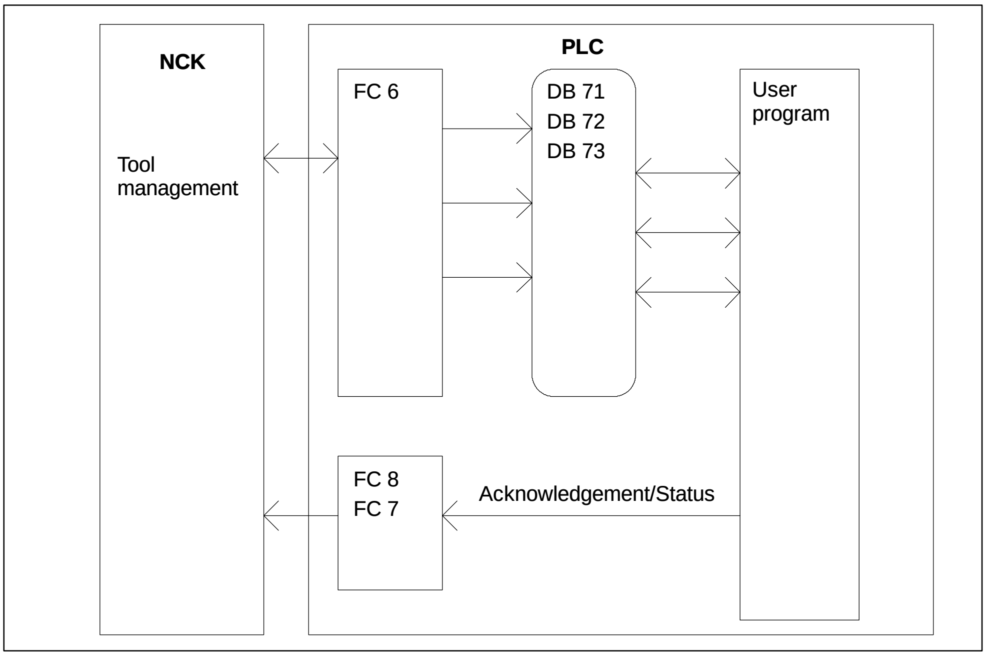


Bild 4-1 Starting Up the PLC Program

FC 6 supplies the tool management interfaces (data blocks DB 71–DB 73) with information for the new and old tool. The user must process this data from the active interface in the user program and ensure that the tools (old and new) are placed on the respectively associated positions (magazine, location). In order for the tool management (TOOLMAN) to always know where a tool is located, each time a tool changes location the new location must be transferred to the tool management via FC 7 or FC 8 acknowledgement status.

4.2.1 Create PLC data

When all magazines, buffers and loading points (for all channels / TO areas) are entered, the data must be passed to the PLC. Activate the **Create PLC data** softkey to download the data to the PLC (HMI Advanced only).

Notice

If new PLC data has been “generated”, data blocks DB 71–74 must be deleted and the PLC then cold restarted. The DBs are then set up for the new configuration.

Start-up of tool management function

Tool management in the PLC is set up by starting tool management in the HMI (create PLC data) and activating the NCK option “tool management”.

Before start-up of the PLC part of the tool management can be initiated, block FC 6 (part of the basic program) must be loaded in the PLC. The basic program calls this block; it does not need to be called in the user program as well.

FC 8 TM_TRANS (FC 7 with turret magazines), TM_TRANS (transfer block) and if required FC 22 TM_DIR (selection of direction) must be laoded and called by the user program.

When installation and startup is complete, the next time the PLC is booted the following data blocks are set up for the user (user interfaces for tool management) in addition to a data block for the tool management FCs. The lengths of the data blocks are derived from the start-up parameters in tool management (softkey Create PLC data) (HMI Advanced only).

Example of chain magazine

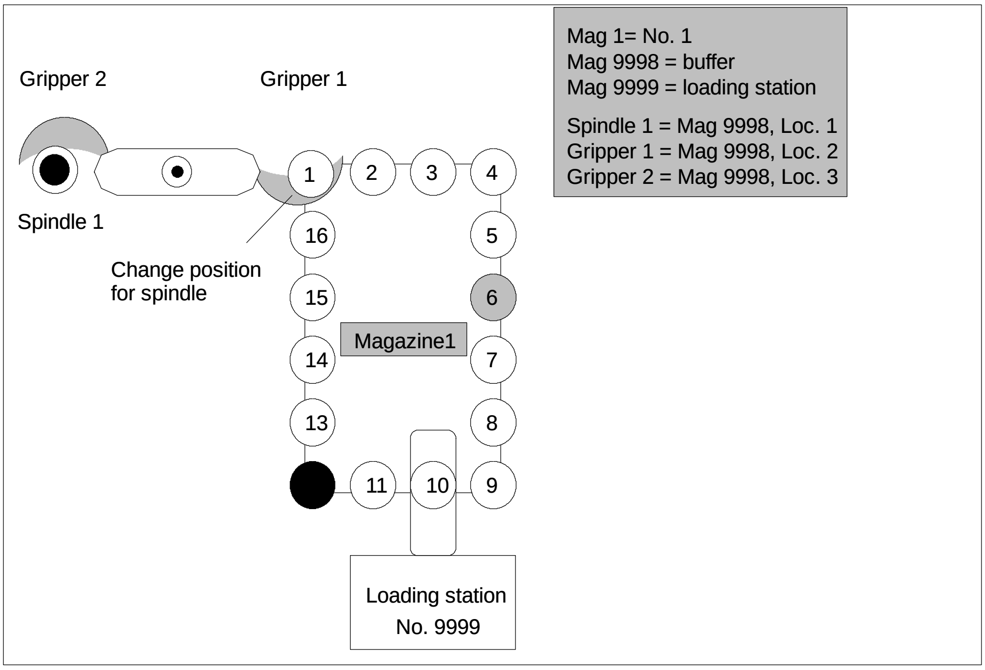


Bild 4-2 Example of a magazine with gripper and loading station

Tool “Drill120” is placed in location 6 and location 12 is reserved for the spindle tools to be exchanged.

Execution example for tool change

1. Part program contains T=“Drill120”
   Output to PLC:
   “PREPARE CHANGE” DBB(n+0) Bit2=1
   (bring new tool from Mag1, location 6 to Mag9998, location 1 and bring old tool from Mag9998, location 1 to Mag1, location 12).
2. Location 6 is moved to the point of change.
3. The tool is taken from location 6 and placed into gripper 1. “PREPARE CHANGE” DBB(n+0) Bit2 is reset to zero by the user program. The new position (9998, 2) of the new tool (“Drill 120”) is signaled via FC 8 with status 1. The old tool remains at position 9998, 1. FC 8 resets bit 0.0 in DB 72.
   The magazine is moved with location 12 to the change position for the old tool to be placed into it.
4. M06 is executed in the part program
   Output to PLC: “CHANGE” DBB(n+0) bit1=1
   No new tool positions are entered in the interface with output of the M06 command. If required, they can be later made by the user program at change of position.

5. The PLC user program carries out the tool change and brings the tool into the spindle. During this process, the old tool is removed from the spindle and placed into gripper 2. The new tool in gripper 1 is placed into the spindle. When the process is completed, FC 8 acknowledges with status 105 (position of new tool: 9998, 1; position of old tool 9998, 3).
6. The (old) tool is returned from gripper 2 to magazine location 12. This is acknowledged via FC8, status 1 (position of new tool: 9998, 1; position of old tool 1, 12). This represents the end of the tool change procedure. Bit 0.0 in DB 72 is reset by FC 8.

Notice

The timing of the tool change can be optimized by applying the following strategy for further processing in the part program:

Use status 1 with FC 8 in step 5 instead of status 105. The old tool is then returned to storage in step 6 with the asynchronous FC 8 transfer function (status 1, OldToolMag=9998, OldToolLoc=3, NewToolMag=1, NewToolLoc=12).

4.2.2 Description of the test blocks

Overview of test blocks

Block No.	Design	Meaning
FC 40	Subprogram	Preparation of the data on a change with gripper via asynchronous transfer
FC 41	Block to be called in OB 1	Global functions (job control, check commands, H decoder, ...)
FC 42	Subprogram	Supply of data for FC 8 if a task is active
DB 62	Data for active tasks Control parameters	
DB 63	Data for FC 22	
DB 64	Data for asynchronous transfer	

Test blocks for tool management

To test the tool management function, blocks FC 40, FC 41, FC 42 and data blocks DB 62, DB 63 and DB 64 must be loaded to the PLC. FC 41 (without parameters) must also be called in the organization block 1 (OB 1). The following overall procedure is implemented by integrating these blocks.

1. The tool management function is activated (acknowledgement of tasks) by programming H9001 in the first channel (and deactivated with H9000).
   The system can also be activated by setting data bit DB62.DBX 15.7. The initial setting when the PLC is rebooted is H9000.
   The other functions can only be used once the system has been activated via H9001.
2. The direction selection function (FC 22) can be activated with the machine control panel (MCP) key above the rapid traverse override key (i.e. the normal MCP connected via FC 19 or FC 25). Data must be written to data block DB 63 (e.g. via the variable status) before the function is activated.

Structure of data block DB 63:

Input parameters

DBW 0 = Magazine number
DBW 2 = Setpoint position
DBW 4 = Actual position
DBW 6 = Offset for special positioning

Output parameters

DBW 8 = Differential position (shortest path)
DBB 10 = Rotation in CW direction == 1
DBB 11 = Rotation in CCW direction == 1
DBB 12 = Position reached
DBB 13 = Error == 1

If an error (e.g. parameterizing error) occurs, the LED for the key lights up.

3. Every user interface (DB 71 to DB 73) is scanned for active status by block FC 41.
   If an interface is active, a transfer with new positions (usually target positions) and status information “1” (completed) is passed to the NCK immediately.
4. If H9003 is programmed in the first channel (equivalent to data block DB 62. DBX 15.6 set), the transfer operation described in paragraph 3 is only executed after operating the MCP key above the minus-direction key.
   This allows the the transfer values to be influenced via the status function. The function is deactivated via H9002 (default setting). The transfer values are provided in data block DB 62.

Input parameters:
DBB 0 = Task identifier (1, 2, 3)
DBB 1 = Task number

(make changes only in DBW 2 to DBW 10)

DBW 2 = Magazine for new tool
DBW 4 = Location for new tool
DBW 6 = Magazine for old tool
DBW 8 = Location for old tool
DBW 10 = Status information (see description of FC 8)

Output parameters:

DBW 12 = Error
If an error occurs, the LED for the key lights up.

The following functions are implemented for command acknowledgement in DB 71, DB 72, DB 73:
Load/unload, relocate:
The required target positions are acknowledged with status 1 via FC 8.
Positioning:
The required target position is acknowledged with status 5 via FC 8 because the tool remains in the magazine.
Prepare change (spindle interface):
"New tool" remains at the original location,
"Old tool" remains in the spindle.
Special treatment is implemented for T0 or empty spindle.
Acknowledgement is with status = 1 via FC 8.
Change (spindle interface):
"Old tool" is transferred to allocated magazine location,
"New tool" is loaded into the spindle.
Acknowledgement is with status = 1 via FC 8.
Special treatment is implemented for T0 or empty spindle.
Change (turret interface):
Acknowledgement is via FC 7.
Acknowledgement with DB62.DBX 15.4 = 1 is optional via FC 8 with status = 1.

5. Values other than zero can be set in DB62.DBW 20 and DB62.DBW 22. DB62.DBW 20 means the spindle number and DB62.DBW 22 the buffer number of a gripper assigned to the spindle.

   It is thus possible to automatically allow for a gripper *located between a spindle and a magazine in the acknowledgement*.

   The following sequence is implemented (only for spindle as change position, M06 setting as change command):

   The procedure for preparation is identical to "normal operation".
   The "New tool" remains in the magazine,
   the "Old tool" remains in the spindle.
   The "Old tool" must continue to machine.

   On the change command:
   "New tool" is loaded into the spindle.
   "Old tool" is placed into the gripper.

An asynchronous transfer is used to move the “Old tool” to the suggested magazine location.
A manual acknowledgement is required for this purpose.

6. Asynchronous transfer
   (changes in a tool location can be communicated without an NCK task)
   DB 64 can be used to communicate a change in position of a tool to the tool management function in the NCK.
   The position of the tool was changed by the PLC. Entries must be made in DB 64 (e.g. via variable status).
   The asynchronous transfer can then be started with DB64.DBX 14.0 = 1.
   The data in DB62.DBX 15.4 = 1 can be used to select the asynchronous transfer with location reservation.
   This corresponds to TaskIdent = 5.
   If value 0 is stored in the data specified above, TaskIdent = 4 is activated.
   Input parameters:
   DBB 1 = Associated NC channel number
   DBW 2 = Original magazine of tool
   DBW 4 = Original location of tool
   DBW 6 = Target magazine of tool
   DBW 8 = Target location of tool
   DBW 10 = Status information (see description of FC 8)
   Only status = 1 and status = 5 are permissible
   Output parameter:
   DBW 12 = Error

Notice

If incorrect values are communicated from the NCK, the following error signals causing PLC stop are output and either displayed via the HMI or entered in the diagnostics buffer of the PLC.

Alarm 400604:
In function 4 the stated magazine is not a turret.
Remedy: Machine data (tool change with M06 command).

4.2.3 Delete pending tasks

The communication initiated by the NC yet interrupted by the PLC job “Cancel pending job” (DB10.DBX105.0) can be terminated by the PLC during setup.

The function cancels pending tool management jobs from the NCK (compare NC switch-on). The NC tool management is reset in a defined manner.

This function enables direct intervention by the operator to e.g. take a tool out of the gripper where a change is just about to take place, or if there is no acknowledge from the PLC program.

Notice

Please ensure that the data consistency in the NC remains.

Supplementary condition

The “Delete active task” function can be activated only if the NC is in the “Channel not active” state.

4.3 HMI Embedded – create magazine configuration

Graphic support for start-up is not available for tool management with HMI Embedded. The start-up file for the magazine and PLC configuration must be created by the user. It then needs to be executed by the NCK once in order for it to be activated for the magazine configuration.

The start-up file can also be created using the HMI Advanced start-up tool and uploaded to the NCK.

There are several ways to create the start-up file:

- Input as part program at the HMI Embedded operator panel
- External creation on a PC with an ASCII editor without formatting.
- Downloading the example from the toolbox diskette and modifying it on HMI Embedded or PC.

HMI Embedded supports up to 4 real magazines.

4.3.1 Create start-up file

Structure of a start-up file

- Delete old data
- Define the type of search strategy
- Define real magazines
- Define buffer magazine
- Define load magazine
- Define locations for the real magazine
- Define locations for the buffer magazine
- Define spindle assignment (which buffer belongs to the spindle)
- Define the locations for the load magazine
- Define distances (offset) to magazines (which spindle, gripper, loading point belong to which magazine)

Part program

The start-up file is a part program e.g. %_N_MAGKONF_MPF. Two more sample programs are contained on the tool box CD.

Short description of the most important variables

Only the main variables for the configuration file are described here. For a more detailed description of the system variables, see Section 5.4 .

Magazine description data **$TC_MAP3**

$TC_MAP3[MagazineNo]=status of magazine

Default = 17 means: Active magazine, enabled for loading

Search strategy **$TC_MAMP2**

This screen is divided into a right and left byte. The right byte describes the tool search, the left byte describes the location search for the spindle tool. A value must be entered for both strategies (see also Section 3.3.1 and 5.4.7)

Location type **$TC_MPP1**

$TC_MPP1[MagazineNo, LocNo]= Type of location:

Default: value of corresponding location type

Location type **$TC_MPP2**

$TC_MPP2[MagazineNo, LocNo]=Type of location

Any values can be entered here. They must match the tools to be loaded at the location. Buffers and loading points have value 0.

Consider adjacent location **$TC_MPP3**

$TC_MPP3[MagazineNo, LocNo]= Consider adjacent location ON/OFF

Location status **$TC_MPP4**

$TC_MPP4[MagazineNo, LocNo]= Location status (bit mask)

Default=2 Location free

Location type index$TC_MPP5

$TC_MPP5[MagazineNo, LocNo]= Location type index

For $TC_MPP1[Magazine No., Location No.]=1 (location type is the magazine location), the location number is entered. For other location types, the type index is incremented:

Example with 2 grippers with location type 3

- the first gripper has location index 1
- the second gripper has location index 2

Distance between a change position, loading point and a zero point

Offsets (distances) to the magazine

$TC_MDP2[magazine no., buffer location no.]
Distances between buffer location and magazine

A value must be entered for each buffer, at least a zero. The value is not evaluated at this point, it is only for assignment.

$TC_MDP1[magazine no., loading point no.]
Distances between loading points and magazine

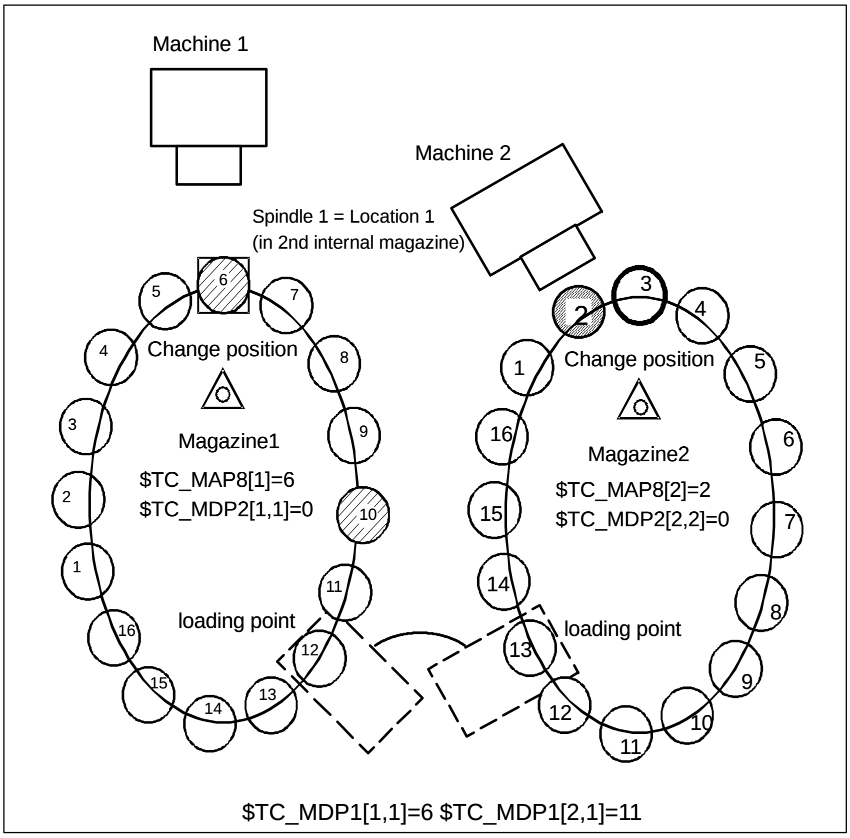


Bild 4-3 Change position, loading point, current position; magazine distance

The zero position is at the change point of the spindles, therefore the following applies: if location 1 is at the change point, the current magazine position = 1 = $TC_MAP8[x]

$TC_MDP**1**[1,1] = 6	**Distance** between Location 1 of the **loading point** and the **zero position** of the magazine
$TC_MDP1[2,1] = 11	Distance between the same location from the zero position of magazine 2
$TC_MDP**2**[1,1] =0	**Distance** of location 1 of the 2nd internal magazine (**spindle 1**) from the **zero position** of magazine 1
$TC_MDP2[2,2] = 0	Distance between the same location and the zero position of magazine 2

Assignment of magazine locations to spindles

$TC_MLSR [location no. of the buffer, location no. of the spindle in the buffer magazine]

This variable assigns buffers which have a link between a spindle and the magazines assigned to the spindle. This enables determination of which buffer, e.g. gripper, may carry out tool change to the spindle.

In Fig. 4-8 for example, gripper 2 in location 3 can change the tool in the spindle in location 1 ($TC_MLSR[3,1]).

Example of a start-up file

Plant configuration:

- 1 chain magazine with 50 locations
- 3 buffer locations
- 2 load points

```
%_N_MAGKONF_MPF
;$PATH=/_N_MPF_DIR
N10 ;
N20 ;
N30;
N40 ;
----------------------------------
N50 ; Magazine configuration: MMC100
----------------------------------
N60 ;
N70 ;
N80 ; Delete old data
N90 ;
N100 $TC_MAP1[0]=0
N110 $TC_DP1[0,0]=0
N120 ;
```

```
N130 ; Configuration
N140 ;
N160 $TC_MAMP2=4097               ; Type of search strategy
N170 ;
N180 ; Magazines
N190 ; Real magazine with number [1]
N200 $TC_MAP1[1]=1                ; Magazine type (1: Chain, 3: Turret,
                                  ; 5: Box)
N220 $TC_MAP3[1]=17               ; Magazine status
                                  ; (see also Planning Guide)
N230 $TC_MAP6[1]=1                ; Number of tiers in magazine
N240 $TC_MAP7[1]=50               ; Number of magazine locations
N250 ;
N260 ; Definition of buffer magazine (always number 9998)
N270 $TC_MAP1[9998]=7             ; Magazine type: 7: Buffer
N280 $TC_MAP3[9998]=17            ; Magazine status
N290 $TC_MAP6[9998]=1             ; Number of tiers
N300 $TC_MAP7[9998]=3             ; Number of locations
N310 ;
N320 ; Definition of loading magazine (always number 9999)
N330 $TC_MAP1[9999]=9             ; Magazine type: 9: Loading magazine
N340 $TC_MAP3[9999]=17            ; Magazine status
N350 $TC_MAP6[9999]=1             ; Number of tiers
N360 $TC_MAP7[9999]=2             ; Number of locations
N370 ;
N380 ; Locations of chain magazine
N390 ;
N400 $TC_MPP1[1,1]=1              ; Location type
N410 $TC_MPP2[1,1]=2              ; Location type
N420 $TC_MPP3[1,1]=1              ; Consider adjacent location ON
                                  ; (OFF would be 0)
N430 $TC_MPP4[1,1]=2              ; Location state
                                  ; (see also Planning Guide)
N440 $TC_MPP5[1,1]=1              ; Location type index
N450 ;
N460 $TC_MPP1[1,2]=1
N470 $TC_MPP2[1,2]=2
N480 $TC_MPP3[1,2]=1
N490 $TC_MPP4[1,2]=2
N500 $TC_MPP5[1,2]=2
N510 ;
N520 $TC_MPP1[1,3]=1
N530 $TC_MPP2[1,3]=2
N540 $TC_MPP3[1,3]=1
N550 $TC_MPP4[1,3]=2
N560 $TC_MPP5[1,3]=3
N570 ;
N580 $TC_MPP1[1,4]=1
N590 $TC_MPP2[1,4]=2
```

```
N600 $TC_MPP3[1,4]=1
N610 $TC_MPP4[1,4]=2
N620 $TC_MPP5[1,4]=4
N630 ;
N640 $TC_MPP1[1,5]=1
N650 $TC_MPP2[1,5]=2
N660 $TC_MPP3[1,5]=1
N670 $TC_MPP4[1,5]=2
N680 $TC_MPP5[1,5]=5
N690 ;
.......
......
N3160 $TC_MPP1[1,47]=1
N3170 $TC_MPP2[1,47]=2
N3180 $TC_MPP3[1,47]=1
N3190 $TC_MPP4[1,47]=2
N3200 $TC_MPP5[1,47]=47
N3210 ;
N3220 $TC_MPP1[1,48]=1
N3230 $TC_MPP2[1,48]=2
N3240 $TC_MPP3[1,48]=1
N3250 $TC_MPP4[1,48]=2
N3260 $TC_MPP5[1,48]=4
8N3270 ;
N3280 $TC_MPP1[1,49]=1
N3290 $TC_MPP2[1,49]=2
N3300 $TC_MPP3[1,49]=1
N3310 $TC_MPP4[1,49]=2
N3320 $TC_MPP5[1,49]=49
N3330 ;
N3340 $TC_MPP1[1,50]=1
N3350 $TC_MPP2[1,50]=2
N3360 $TC_MPP3[1,50]=1
N3370 $TC_MPP4[1,50]=2
N3380 $TC_MPP5[1,50]=50
N3390 ; locations in the buffer
N3400 $TC_MPP1[9998,1]=2          ; Location type (here spindle)
N3410 $TC_MPP2[9998,1]=0          ; Location type: as BUF is 0 here
N3420 $TC_MPP3[9998,1]=0          ; Consider adjacent location OFF
N3430 $TC_MPP4[9998,1]=2          ; Location state
N3440 $TC_MPP5[9998,1]=1          ; Location type index
N3450 ;
N3460 $TC_MPP1[9998,2]=3          ; Gripper 1
N3470 $TC_MPP2[9998,2]=0
N3480 $TC_MPP3[9998,2]=0
N3490 $TC_MPP4[9998,2]=2
N3500 $TC_MPP5[9998,2]=1
N3510 ;
N3520 $TC_MPP1[9998,3]=3          ; Gripper 2
```

```
N3530 $TC_MPP2[9998,3]=0
N3540 $TC_MPP3[9998,3]=0
N3550 $TC_MPP4[9998,3]=2
N3560 $TC_MPP5[9998,3]=2
N3870 ;
N3880 ; Spindle assignment        ; Spindle assignment
N3890 $TC_MLSR[2,1]=0             ; 1st gripper (location 2) belongs to
                                  ; spindle (location 1)
N3900 $TC_MLSR[3,1]=0             ; 2nd gripper (location 3) belongs to
                                  ; spindle (location 1)
N3920 ; Loading magazine locations
N3930 $TC_MPP1[9999,1]=7          ; Location type loading point
                                  ; (for spindle!)
N3940 $TC_MPP2[9999,1]=0          ; Location type (here always 0)
N3950 $TC_MPP3[9999,1]=0          ; Consider adjacent location OFF! N3960
$TC_MPP4[9999,1]=2                ; Location status: Free
iN3970 $TC_MPP5[9999,1]=1         ; Location type index
N3980 ;
N3990 $TC_MPP1[9999,2]=7
N4000 $TC_MPP2[9999,2]=0
N4010 $TC_MPP3[9999,2]=0
N4020 $TC_MPP4[9999,2]=2
N4030 $TC_MPP5[9999,2]=2
N4040 ;
N4650 ; Offsets (distances)       ; Distances to magazine
N4660 ;
N4670 $TC_MDP2[1,1]=0             ; Spindle
N4680 $TC_MDP2[1,2]=0             ; Gripper 1
N4690 $TC_MDP2[1,3]=0             ; Gripper 2
N4700 $TC_MDP1[1,1]=0             ; 1st loading point
N4710 $TC_MDP1[1,2]=25            ; 2nd loading point
                                  ; (distance 25 to actual position)
N4720 ;
N4730 ; End
N4740 ;
N4750 M30
```

Load and activate the start-up file

If the IBN file was created on an external PC, then it has to be transferred for control purposes to the directory _N_MPF_DIR.

In order to activate the IBN file in the NC, it must be started as a part program and be handled in the following way:

- Select the part program, e.g. _N_MAGKONF_MPF.MPF
- Execute the program with NC Start.

4.3.2 Create PLC data with HMI Embedded

The data for the initial settings is contained in DB 4 from data word 64 (see Section 9.5). This data must be described by the PLC user program. The number of magazines, loading points, spindles and turrets is determined from this data and used to automatically set up the tool management data blocks (DB 71 to DB 74). The start-up routine is part of the basic program.

4.4 HMI Advanced – create magazine configuration

4.4.1 Create configuration file

Real magazines

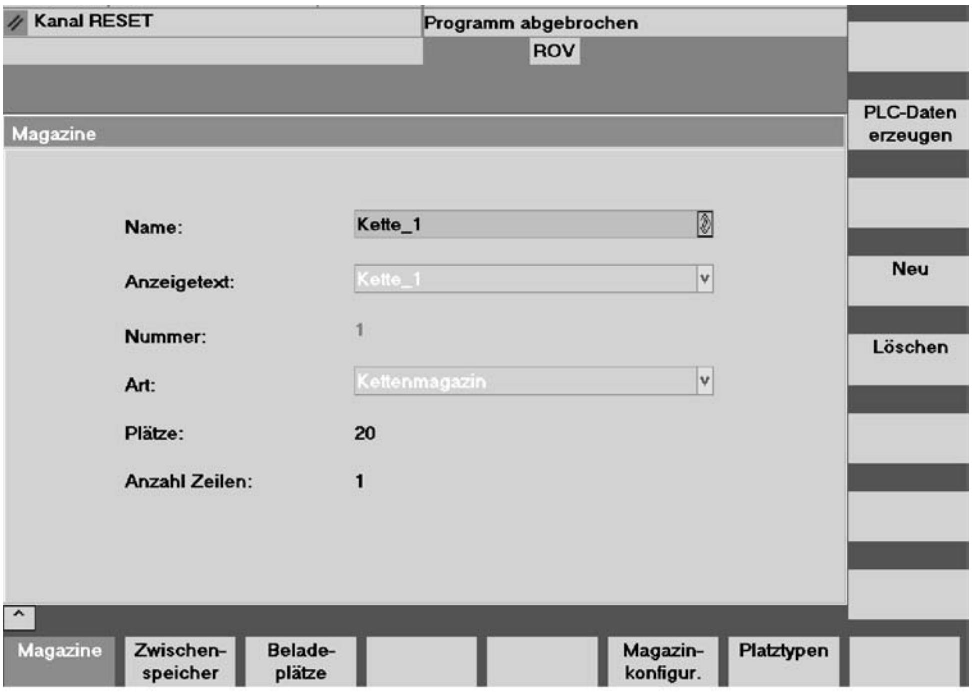


Bild 4-4 Start-up: Magazines

Magazines

In this screen, magazines are defined with the appropriate data or displayed with existing data.

Name	Enter or select the name of the real magazine (new).
Display text	Language-dependent name of the magazine (refer to Chapter 4.4.3)
Number	Display of the current magazine number
Type	Selection of a magazine type (chain magazine, turret, box-type magazine)
Locations	Enter or display number of magazine locations

Number of columns	The “number of columns” is required for considering adjacent location and is only relevant for box-type magazines.

Notice

Up to 32 magazines (including buffer and loading magazine) are possible, therefore a maximum of 30 real magazines.

Create a new magazine

1. Press softkey **New**. The magazine number is assigned by the system (in the order they were created).
2. Enter magazine name with up to 32 characters in length If it exists, the display text is immediately displayed from patm_xx.ini (see Section).
3. Select the magazine type:
   - Chain magazine
   - Revolver
   - Flat magazine
4. Enter number of magazine locations
5. For box magazines the “number of columns” must be entered too.
6. Accept the data with softkey **OK**

Notice

If the message “invalid value in magazine” appears, the number of locations and/or number of columns is incorrect. The “number of locations” value must be divisible by the “number of columns”.
Example:
20 locations cannot be divided into 3 columns, but 21 locations can be.

Delete magazine

1. Select the magazine name
2. Press the **Delete** softkey.
3. The magazine is deleted without a prompt.

Notice

A magazine can only be deleted if it is not assigned to any magazine configuration.

Create PLC data

When all magazines, buffers and loading points (for all channels / TO areas) are entered, the configuration data must be created for the PLC and downloaded to it.

To do this, press softkey **PLC data Creating** .

Notice

The next time the PLC is booted, the message “Deleting DB xx in PLC...” may appear. In this case, the specified DB is deleted via STEP 7. As an alternative a suitable PLC archive can also be loaded.

Buffer

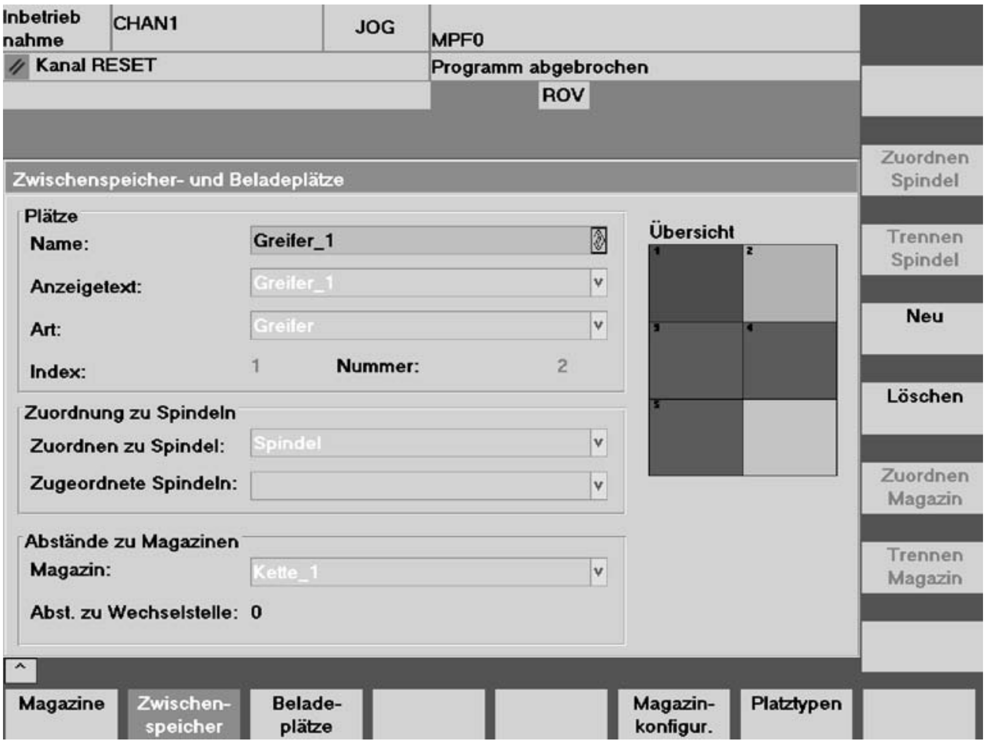


Bild 4-5 Start-up: Buffer

Buffer

In this screen buffer locations are defined (New) or existing ones are displayed.

Name	Enter or select name of the buffer.
Display text	Language-dependent name of the magazine If it exists, the display text is immediately displayed from patm_xx.ini (see Section 4.4.3).
Number	Display of the current magazine number
Type	Selection of buffer type (spindle/toolholder, gripper, transfer location, loader)
Index	The index counts the locations of a type.
Number	Display of consecutive internal number under which the location is to be addressed
Overview	The number of all buffer locations is displayed as a graph. In addition, the currently selected buffer location (No.) is highlighted. Each “type” is displayed in a different color.

Buffer locations are spindles, grippers, loaders and transfer locations.
All buffer locations are managed in an internal buffer magazine with the number 9998.

Create a buffer location

Notice

During input of the buffer, the sequence needs to be adhered to. The spindles must always be entered first. For each buffer location a number is assigned internally over which the buffer location is addressed.

The buffer “spindle” with index 1 and spindle_1 in the NC have a direct relationship to one another. This means that for example the “spindle” buffer with index 1 must also be the 1st spindle for the NC, index 2 = 2nd spindle, etc.

1. Press the **New** softkey.
2. Enter name: e.g. **Gripper_1**. If it exists, the display text is immediately displayed from patm_xx.ini (see Section 4.4.3).
3. Select the type: Transfer location, **gripper**, loader, spindle
4. The buffer location is created by pressing **OK**, the location number and the index are internally assigned and incremented.

Assign/remove spindle

Softkey **Assigning Spindle** assigns a buffer location (e.g. gripper) to a spindle. This allows the mechanical relation between the gripper and spindle to become known to the software.

The softkey **Remove Spindle** cancels an existing spindle assignment.

Assigning/removing buffer locations to/from the magazines

When a magazine is selected, the distance to the change position must be entered. At least 0 must be entered as otherwise the tool cannot be transported to this buffer location. The softkey **Assigning Magazine** assigns the buffer location to the magazine.

The softkey **Remove Spindle** cancels an existing assignment of a buffer location to a magazine.

Example

If for example tools are to be loaded from 2 magazines onto “Spindle_1”, the assignment must be made for both magazines.

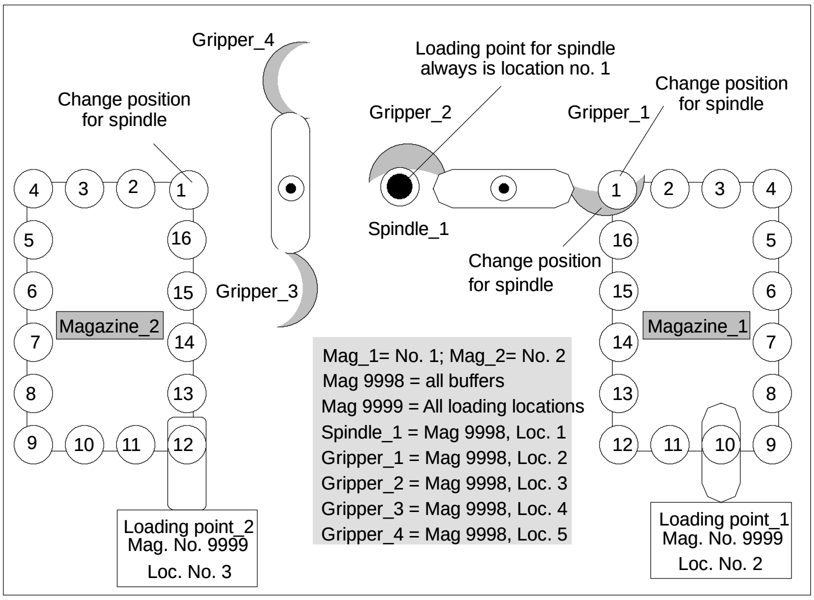


Bild 4-6 Example of a machine with buffer and loading magazine

No.	Name	Distance to change position
1	Spindle_1	Magazine_1, distance: 0 Magazine_2, distance: 0
2	Gripper_1	Magazine_1, distance: 0
3	Gripper_2	Magazine_1, distance: 0
4	Gripper_3	Magazine_2, distance: 0
5	Gripper_4	Magazine_2, distance: 0

Loading locations

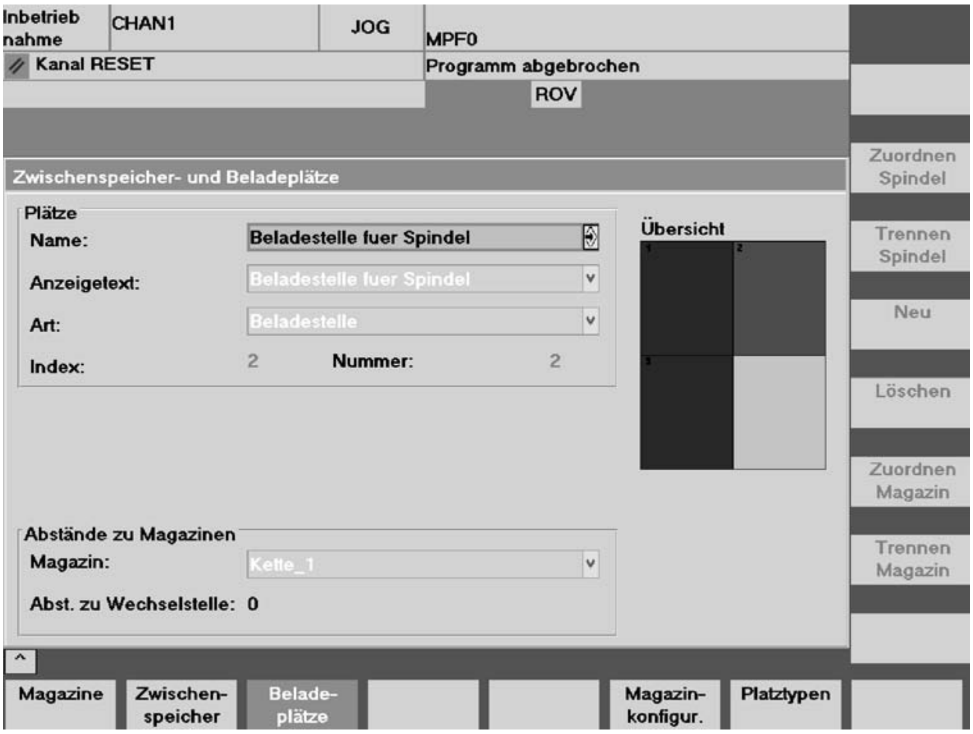


Bild 4-7 Start-up: Loading locations

Loading locations

In this screen loading locations are defined (New) or existing ones are displayed.

Name	Name of loading location (max. 32 characters).
Display text	Language-dependent name of the magazine. If it exists, the display text is immediately displayed from patm_xx.ini (see Section 4.4.3).
Number	Display of the current magazine number
Type	The options are loading point and loading station.
Index	The index counts the locations of a type.
Number	Display of consecutive internal number under which the location is to be addressed
Overview	The overview displays the number of all buffers graphically. The numbers for the selected loading point/station are highlighted in color.

Loading points are locations that are needed for loading the magazine. There are two types of loading points:

- Loading locations
- Loading stations

All loading locations are stored in an internal loading magazine with the number 9999.

Notice

Loading magazine 9999/1 is always automatically set for loading/unloading of the spindle (manual loading point).

Loading locations

Loading points are areas on the machine at which it is possible to directly load the magazine, i.e. the tool can be directly inserted in the magazine at this point. The magazine location to be loaded is moved to the loading point. For example, chain magazines have loading points.

The loading point is assigned to location type “7” ($TC_MPP1) in the magazine data.

Location type 7
If a tool is moved to this location from the magazine/toolholder, the NCK automatically removes the tool from this location when acknowledgement is received from the PLC.

Loading stations

A loading station is a location outside the magazine onto which the tool to be loaded is placed. The tool is then transported from that location to the magazine via a transport mechanism. Loading stations are generally used for box or chain magazines.

The loading station is assigned to location type “6” ($TC_MPP1) in the magazine data.

Location type 6
NCK does not differentiate between cases, i.e. if the tool was moved to this location, the tool remains there. It can only be removed (unloadd) from there though explicit operator action.

Input of loading locations

1. Press the **New** softkey.
2. Enter name: e.g. **Load_1**. If it exists, the display text is immediately displayed from patm_xx.ini (see Section).
3. Select the type: e.g. **Loading point**
4. The loading location is created by pressing **OK**. The location number and the index are internally assigned and incremented.

Assigning/removing loading locations to/from the magazines

When a magazine is selected, the distance to the change position must be entered (at least 0).

You can now establish an assignment by pressing the **Assigning Spindle** softkey. Distance 0 is usually used for the "manual loading point" (spindle loading point).

Example

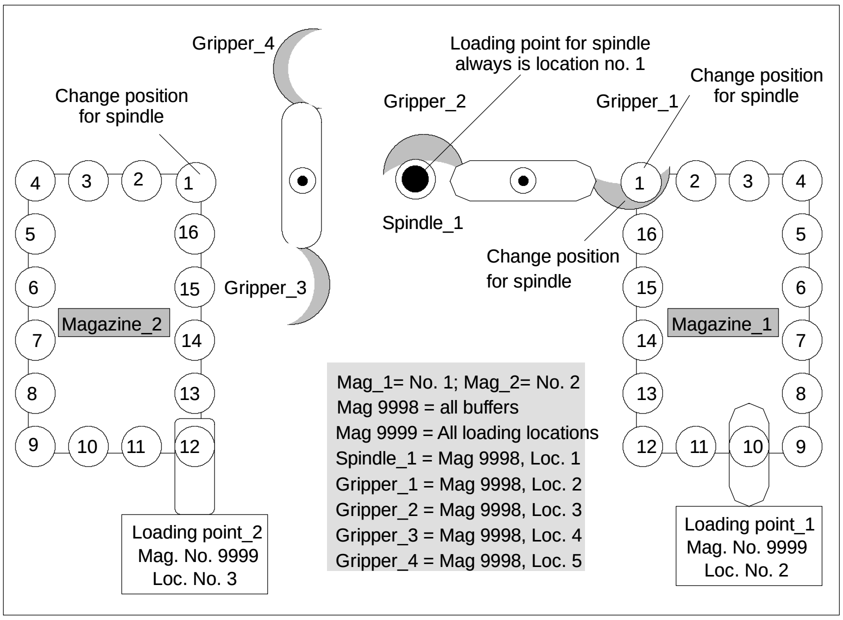


Bild 4-8 Example of a machine with 2 magazines and 3 loading locations

No.	Name	Distance to change position
1	Manual loading point (= spindle loading point)	Magazine_1, distance: 0 Magazine_2, distance: 0
2	Loading point_1	Magazine_1, distance: 9
3	Loading point_2	Magazine_2, distance: 11

Both magazines can be loaded via Spindle_1. Loading point_1 is only Magazine_1 and Loading point_2 is only assigned to Magazine_2.

Location types

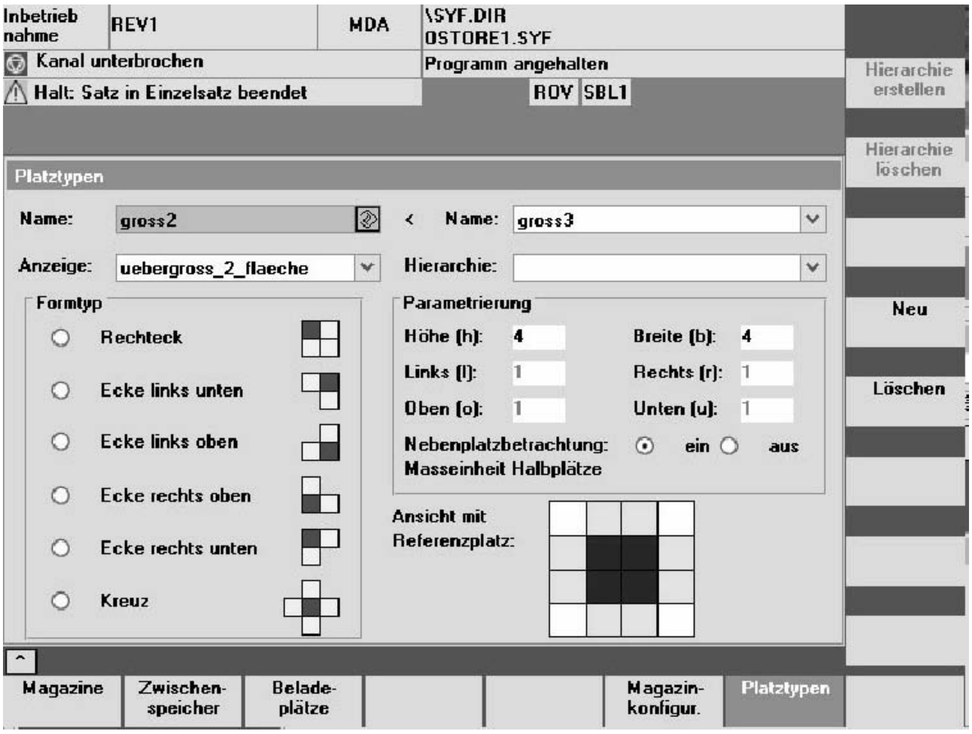


Bild 4-9 Start-up: Location types

In this screen location types are defined with the required data or existing ones are displayed.

Name	Name of the location type (max. 32 characters).
Display	Language-dependent name of the magazine. If it exists, the display text is immediately displayed from patm_xx.ini (see Section 4.4.3).
Hierarchy	To overcome the inflexible classification of magazine locations according to location type, locations can be arranged in ascending order, i.e. in a hierarchy (see Section “Hierarchy of location types”).

Form type	The position of the reference location (tool shank) is specified via the form type.
Parameter assignment	Definition of height and width as well as free half locations (left, right, top, bottom) (see examples).
Consider adjacent location	This information is stored magazine specifically (magazine configuration) and it relevant for the location search.

Assigning parameters for a location type

The number of half locations occupied by a tool in the magazine is defined when a location is parameterized. This corresponds to the tool size.
The four-digit number, e.g. 2 2 2 2, defines the half locations in the order left, right, top, bottom starting from a reference point. For setting the location type parameters, left plus right equals width and top plus bottom equals height.
The half locations that are not occupied are also defined using left, right, top and bottom (do not confuse this with the tool size!).

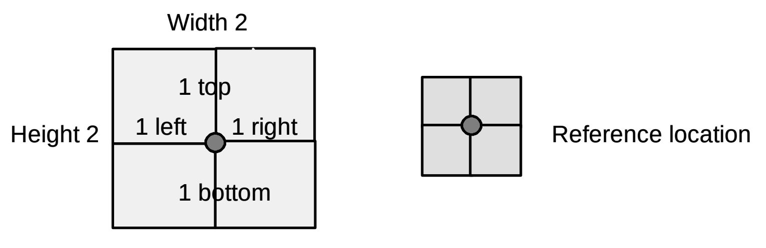
Reference location

The reference location is the physical location in the magazine. It is used as a reference point for specifying the tool size and is required for calculating the magazine assignment. The size of the reference location is always represented as tool size 1 1 1 1. (For parameter settings, refer to “Normal location type”)

Examples of parameter settings

Normal location type

A tool which occupies one magazine location exactly has tool size 1 1 1 1. This tool is referred to as a “normal sized tool”.



Left, right, top, bottom are the half locations of the tool size 1 1 1 1

Bild 4-10 Normal location type

Parameter settings for the location type:

Height (h):	2	Width (b):	2
Left (l):	0	Right (r):	0
Top (o):	0	Bottom (u):	0

Oversized location type for chain magazines

In a chain magazine, a tool with the size 2 2 1 1 occupies one half location on the right and one half location on the left in addition to the normal magazine location.

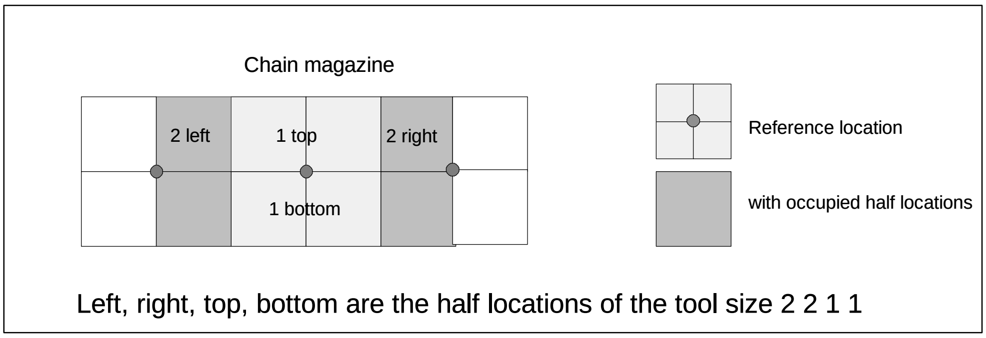


Bild 4-11 Oversized location type for chain magazines

Parameter settings for the location type:

Height (h):	2	Width (b):	4
Left (l):	0	Right (r):	0
Top (o):	0	Bottom (u):	0

Oversized location type for box magazines

In a box magazine, a tool with the size 2 2 2 2 occupies one half location in each direction in addition to the normal magazine location.

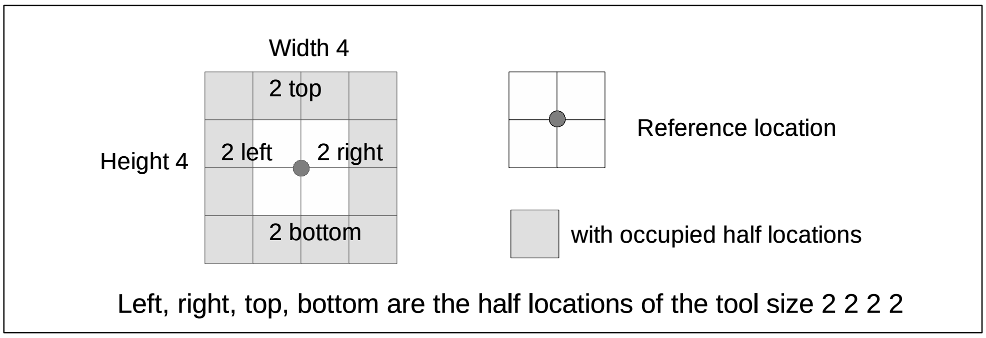


Bild 4-12 Oversized location type for box magazines

Parameter settings for the location type:

Height (h):	4	Width (b):	4
Left (l):	0	Right (r):	0
Top (o):	0	Bottom (u):	0

Oversized location type with free half locations for box magazine

In a box magazine, a tool with the size 2 2 2 2 occupies one half location in each direction in addition to the normal magazine location.
However, with this location type the half location in each corner is not used.

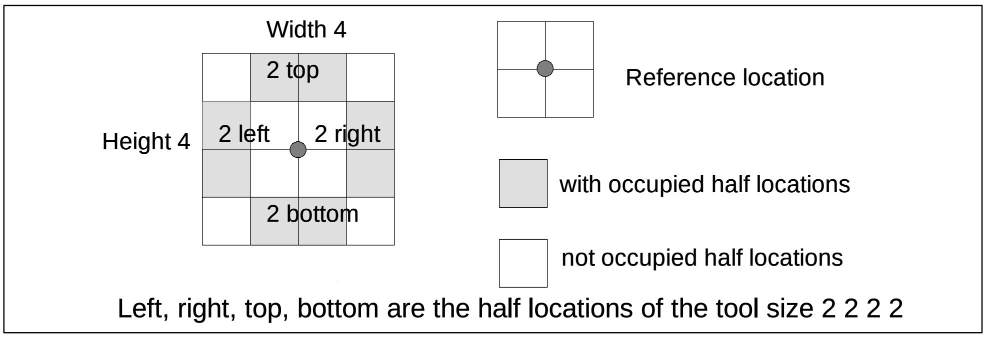


Bild 4-13 Oversized location type with free half locations for box magazine

Parameter settings for the location type
(see also Fig. 4-9, location type “oversized_2_box):

Height (h):	4	Width (b):	4
Left (l):	1	Right (r):	1
Top (o):	1	Bottom (u):	1

The half locations not occupied are defined by the parameter left, right, top, bottom as FREE.

Hierarchy of location types

To overcome the inflexible classification of magazine locations according to location type, locations can be arranged in ascending order, i.e. in a hierarchy. Several such hierarchies can be set up for TO units but a location type can only belong to one hierarchy. (For example, A < B and A < C or A < E and B < E are not permissible.)

This hierarchy ensure that a tool that only requires a “small” location type can also be placed in a “larger” location type if no “small” locations are free.

If a tool is to be inserted in the magazine, the location type decides which locations are available. If there is a hierarchy for this location, the locations are allocated in accordance with this hierarchy.

Let us assume, for example, that a tool of location type A is to be stored in the tool-holding magazine or a location search is to take place for a location of type A.
The following location hierarchy shall apply for this example: A < B < C.
First a check is performed to see whether there is a location with type A in the magazine to be searched.
If there is not, the search function will proceed to search for a location of type B or C.

Example 1:

Existing location types: A, B, C

Defined hierarchy: A < B
B <C

Therefore the entire hierarchy is as follows: A < B < C

Example 2:

Existing location types: A, B, C, D, E

Defined hierarchy: A < B
B < D
C < E

1st hierarchy A < B < D
2nd hierarchy C< E

Enter new location type

1. Press the **New** softkey.
2. Enter name (max. 32 characters). If it exists, the display text is immediately displayed from patm_xx.ini (see Section 4.4.3).
3. **Select form type**
4. **Enter height** and **width** of the location type in half locations. Depending on form type, left, right, top, bottom as required (unoccupied half locations).
5. Select consider adjacent location **ON** or **OFF**
6. In the view graphic, set the position of the reference location using the cursor keys
7. Press softkey **OK** to save

Delete location type

The softkey **Delete** deletes the selected location type. This is only possible if it has not been assigned to a magazine.

Create hierarchy

Select small location type (name left) (in our example location type A).
Select larger location type (name right) (in our example location type B).

Press the **Create hierarchy** softkey to create the hierarchy. The hierarchy name in the screen 4-9 corresponds to the name of the larger location type and is displayed in the hierarchy field (B in example).

Delete hierarchy

You can remove the hierarchy selected in the “Hierarchy” field by pressing the **Delete hierarchy** softkey.

Create magazine configuration

For each T0 unit, there is only **one** common magazine configuration for configuring the tool management. A magazine configuration can consist of one or more real magazines. As one T0 unit can be assigned to several channels, this magazine configuration is available to the associated channels simultaneously.

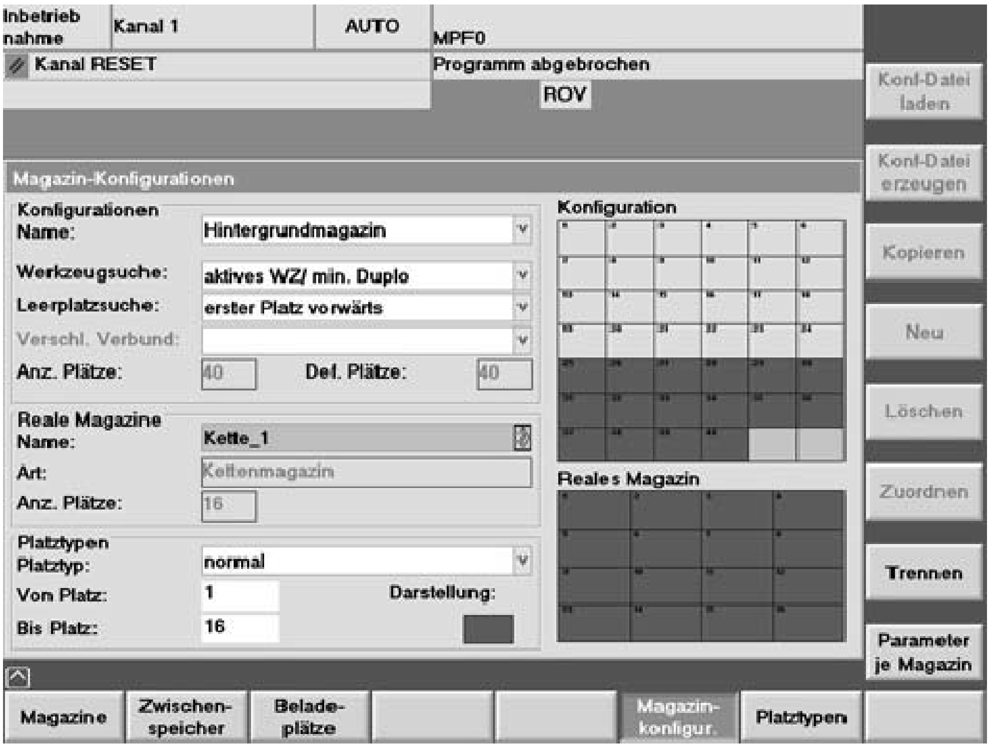


Bild 4-14 Magazine configuration

In this screen data required for the selected magazine is defined or existing data displayed.

<u>Configurations</u>

Magazine name Name of the magazine configuration (max. 32 characters).

Tool search	transfer from configuration (default value. Setting for $TC_MAP10=0, NCK uses values from $TC_MAMP2). active tool/min. duplo Shortest path active tool/min. $TC_TP10 Monitoring: min. actual value Monitoring: max. actual value Settings for the location coding can be variable (default) or fixed. You can exit the screen by pressing the softkey “Cancel” or “OK”.
Empty location search	First location forwards Current location forwards Last location backwards Current location backwards Symmetric current location For a detailed description, refer to Section 3.4
Wear group	Tool status remains unchanged Change “active” status for tools (for a detailed description, refer to section 3.1.6)
Number of locations	Entire number of locations in the configuration (all assigned magazines)
Def. locations	Entire number of locations in the configuration to which a location type was assigned

<u>Real magazines</u>

Name	Name of the selected real magazine
Type	Type of magazine
No. of Locations	Number of locations for the selected magazine

<u>Location types</u>

Location type	Name of the selected location type
From location	1st location to be defined
To location	Last location to be defined

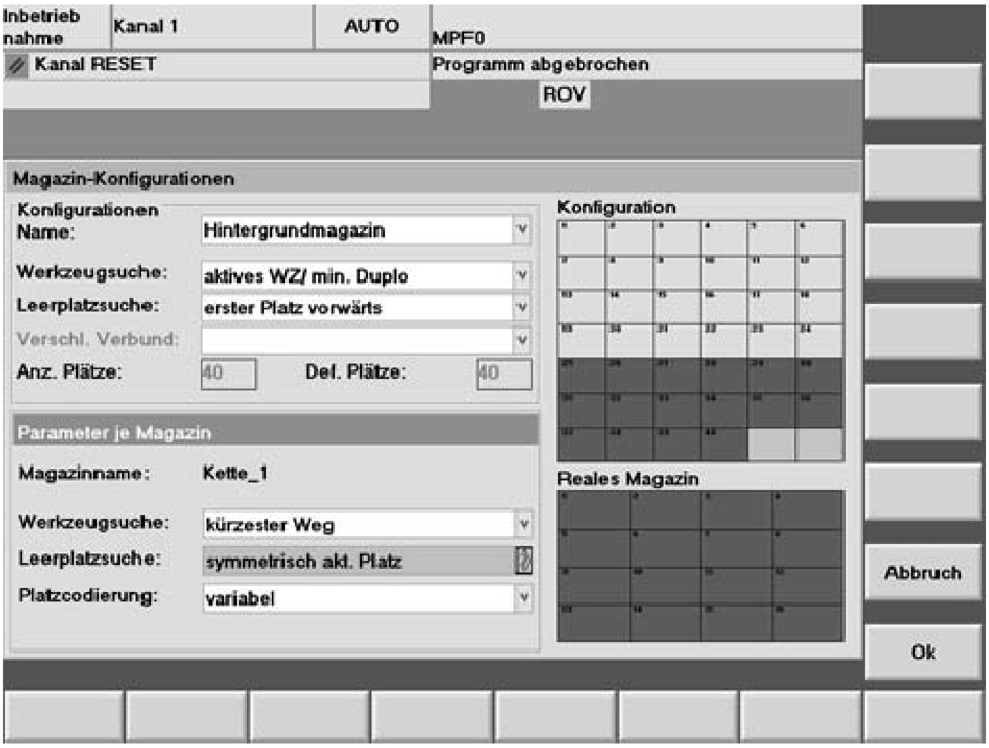


Bild 4-15 Set parameters for each magazine

The settings made in the previous screen (magazine configuration) become effective.

Create new magazine configuration (basic setting)

The screens “Configuration” and “Real magazine” show the location type assignments for the entire configuration/selected magazine.

1. Press the **New** softkey.
2. Enter name, e.g. Example_documentation (max. 32 characters)
3. Select selection menu for the tool search, location search and wear group and make the settings as appropriate.
4. Press softkey **OK** (create the magazine configuration)

Notice

The softkey “Assign” and “Remove” always refer to the selection field the cursor is positioned on:
– Magazine
– Location type

Assign/remove real magazines

Select real magazine and press softkey **Assigning**. This real magazine is then included in the magazine configuration.
After each assignment, the total number of magazine locations is accordingly updated in the configuration.
You can remove a magazine again from the magazine configuration by pressing the **Remove** softkey.

Assign/remove location types

Select location type. Enter magazine locations to which this location type is to be assigned.
Example: “From location:” 1, “To location:” 10.

Press the **Assigning** softkey. The defined locations are displayed with the color for this location type

You can remove the assignment made for a location type by pressing the **Remove** softkey.

Create configuration file

Press the **Create configuration file** softkey. An INI file is created which can later be uploaded to the NCK.

Copy configuration

Press the **Copy** softkey.

Enter a new name and confirm by pressing the **OK** softkey.

Delete magazine configuration

If the cursor is positioned on the selection field for the configuration, you can delete the selected configuration by pressing the **Delete** softkey.

Load magazine configuration

Activating the softkey **Load con- figuration file** opens the 4-16 “Load magazine configuration” screen.

Load magazine configuration

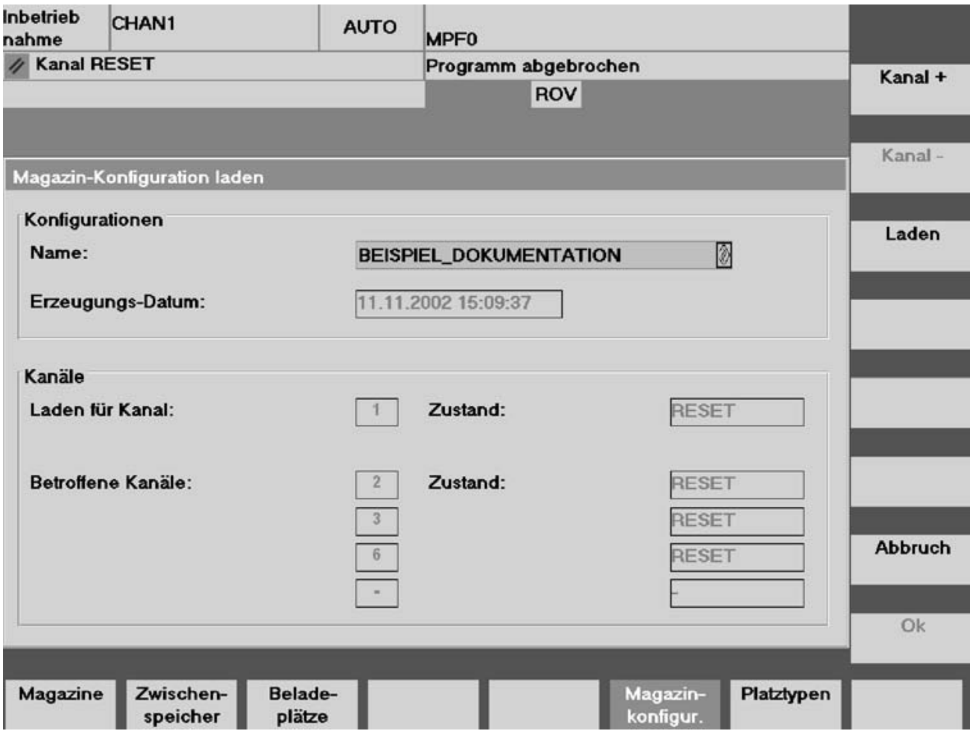


Bild 4-16 Start-up: Load configuration file

Here the previously created INI file is uploaded to the NCK via softkey **Load**. The procedure is channel-specific; only one configuration is possible for each TO unit.
(This means if TO unit 1 is assigned to channels 1, 2, 3 and 6, the configuration which was loaded in channel 1 is automatically valid for channels 2, 3 and 6 too.)

4.4.2 Adapt tool management operator interface for HMI Advanced

Activate the tool displays

Where the customer has made changes to the shipped INI files, the customized differences are saved in the **user** user directory. Only changes to the visual aspect of the operator interface are stored here, i.e. changes that can be made by settings in the HMI operator interface itself.

As a general rule, only entries that differ from the originals shoudl be stored in the parallel directories for mmc2.

Configuration

All of the data that describes the tool management operator interface is stored in the file **...\user\paramtm.ini**. You can edit the file by selecting it in /MMC/DOS-Shell and then opening it with the command edit ...\user\paramtm.ini. The file paramtm.ini can also be created on an external PC and copied into the directory ...\user.

Country-specific sections are parameterized in the “language\patm_xx.ini” file in the [BatchTools] section. “xx” here denotes the 2 letters for the country, e.g. gr for German, uk for English, sp for Spanish, nl for Dutch.

New functionality in the lists

The following additional functions have been implemented in the magazine, tool, work correction list and tool details:

- Parameter settings for bitmaps in the list
- Tool identifiers and duplo numbers can be changed in the lists
- New magazine list with several lines
- Job processing of tools
- Tool-status bit “Pre-warning limit” can be changed in displays of lists
- Tool status bit “Unload detection” and “Load detection” in displays of lists and details can be changed
- New tool types
  550 Steel profile
  700 Slotting saw
  711 Edge probe
  720 Oriented probe
  730 Stop
- If adapter data is set in the NCK, then the magazine list can be displayed either transformed or not transformed (softkey on the ETC bar). The setting is made in paramtm.ini, section [TMMODES], with the entry START_MAGLIST_TRANS-FORMED
- Suppression of status bits for tool cabinet, code carrier, SINCOM excerpt from paramtm.ini

;Tool status: If a tool is removed from the NCK and transferred to an external medium (tool cabinet, code carrier, SINCOM), then you can use the following screens to specify which tool status bits should be saved
;Code carrier: Since the standard conversion file wkonvert.txt (see Subsection 4.5.2) has entered 1 byte for the tool status and max. 92 were written to the code carrier up to now, CODECARRIER_TOOLSTATE_MASK receives the default value 92. If the value for CODECARRIER_TOOLSTATE_MASK is expanded, then the size of the dialog variables T9 in wkonvert.txt has to be adapted accordingly.

Structure of file paramtm.ini

Directory ...\user\paramtm.ini

Notice

Until now, the mmc2\paramtm.ini parameter settings for tool management only contained comments consisting of documentation of the individual entries. Almost the entire comment part had to be removed because new entries exceeded the critical file length limit of about 63KB.
A version of paramtm.ini with comments can now be found in mmc2\paramtm.txt.

Overshooting and other errors when reading the parameterization are logged as before in the file ...\user\paramini.out.

A comment can be inserted at the end of entries using a semicolon “;”.

Contents of INI files

```
[ACCESSLEVEL]
SKAVTIVTM=7       ; Activates TM in the PARAM application
SKMGLIST=7        ; Displays magazine list
SKTLLIST=7        ; Displays tool list
SKACLIST=7        ; Displays the tool offset list
SKTOOLLOAD=5      ; Loading tools
SKTOOLUNLOAD=5    ; Unloading tools
SKTOOLMOVE=7      ; Moving tools in the magazine
SKSETTINGS=4      ; Softkey settings
SKFILFCT=4        ; Softkey file functions
SKNXTCHAN=7       ; Enable softkey Next Channel
SKMAGCONF=4       ; Configuration of magazines
SKTOOLCAT=7       ; Tool catalog
SKTOOLCAB=7       ; Tool cabinet
SKSINCOMLD=7      ; Load tool from SINCOM (if code carrier was
                  ; installed)
SKCTORSINCOM=7    ; Code carrier functions or tool loading functions
                  ; from SINCOM (if code carrier was installed)
SKMGLREPR1=7      ; Display 1_MagList
SKMGLREPR2=5      ; Display 2_MagList
SKMGLREPR3=5      ; Display 3_MagList
SKNCTOOLDATA=7    ; Read tool data from NC or file management
                  ; (if ACTIVATE_EDGE_MANAGEMENT_IN_LISTS=True)
SKNCTOOLED=7      ; Cutting edge data
SKNCTOOLSUPV=7    ; Supervision data
SKNCTOOLDL=7      ; DL data
SKNCTOOLGRIND=7   ; Grinding parameters
SKNCDETAILS=7     ; Read tool data from NC (if ACTIVATE_EDGE_MANAGE-
                  ; MENT_IN_LISTS=True)
SKNCNEWTOOLED=6   ; Create new cutting edges in NC
```

```
SKNCDELTOOLED=6    ; Delete cutting edges in NC
SKNCDELTOOL=5      ; Delete tools in NC
SKTRAFO=7          ; Toggle transformed/not transformed view of edge
                   ; data
SKCHECKACTIVATE=6  ; D check and activation
SKMGBUFFER=7       ; Display the buffer
SKMGFIND=7         ; Softkey search and position
SKMGLISTPOS=7      ; Position
SKMGNEXT=7         ; Softkey next magazine
SKTLNEWTOOL=6      ; Create tools in NC
SKTLLREPR1=7       ; Display 1_ToolList
SKTLLREPR2=5       ; Selection of Display 2_ToolList
SKTLLREPR3=5       ; Display 3_ToolList
SKFINDPL1=7        ; Location search, user-defined 1
SKFINDPL2=7        ; Location search, user-defined 2
SKFINDPL3=7        ; Location search, user-defined 3
SKFINDPL4=7        ; Location search, user-defined 4
SKFINDPL=7         ; Location search
SKFINDPLACE=7      ; Location search, tool loading list
SKACTPLACE=7       ; Allows softkey current location
SKLDTOOLDAT=7      ; Softkey tool data in status loaded tools
SKCONFLOAD=4       ; Load a magazine configuration
SKACLREPR1=7       ; Display 1_ActList
SKACLREPR2=7       ; Display 2_ActList
SKACLREPR3=7       ; Display 3_ActList
SKDZERO=7          ; Softkey delete D numbers
SKDFIND=7          ; Softkey search for D numbers
SKBATCH=7          ; Softkey filter lists
SKBFILTER1=7       ; Softkey Filter1
SKBFILTER2=7       ; Softkey Filter2
SKBFILTER3=7       ; Softkey Filter3
SKBFILTER4=7       ; Softkey Filter4
SKBFILTER5=7       ; Softkey Filter5
SKBFILTER6=7       ; Softkey Filter6
SKBMAGFILTER=7     ; Softkey magazine selection (for filter)
SKBATREACT=7       ; Softkey batch function “Reactivate”
SKBATTOCABIN=7     ; Softkey batch function “In cabinet”
SKBATDELTOOL=7     ; Softkey batch function “Delete”
SKBATUNLOAD=7      ; Softkey batch function “Unload”
SKBFILTERACT=7     ; Softkey batch function “Update filter”
SKBATLOAD=7        ; Softkey batch function “Load”
SKBATLIST=7        ; Softkeys for controlling the job processing

ChangeToolTypeWithoutConfirmation=-1
ChangeToolSizeAndToolplace_spec=-1
READ_GUD_LUD=7
WRITE_ZOA=7
READ_SYSVAR=7
EDIT_VIEW=7
```

[DETAILS]

```
TOOLBASE_COL1=20
TOOLBASE_COL2=20
TOOLBASE_COL3=20
TOOLBASE_COL4=20
```

[DEFAULT SETTINGS]

For default settings for creating tools, see paramtm.ini or paramtm.txt in path mmc2

```
; Magazine list: Load, data input directly in the list:
; 0=The default settings must be confirmed with the “Tool data”
; screen due to missing input if they are needed.
; 1=The default settings become effective without confirmation
; (with the exception of tool identifier no.)
; 2=The default settings become effective without confirmation
; (including the tool identifier no.)
DEFAULT_WITHOUT_CONFIRM=0

; Half locations: From 1 to 7
TOOLSIZE_LEFT=1

; Half locations: From 1 to 7
TOOLSIZE_RIGHT=1

; Half locations: From 1 to 7
TOOLSIZE_UPPER=1

; Half locations: From 1 to 7
TOOLSIZE_DOWN=1

; Tool type, from 100 to 1000
TOOLTYPE=120

; Duplo number: From 1 to 32000
TOOLDUPLO=1

; Ident. no.: Max. length 27
TOOLIDENT=NEW

; Additive values, (default: 0):
; 1=Active tool
: 2=Permitted
; 4=Blocked
; 8=Measured
; 16=Warning limit reached
; 32=Being changed
; 64=Fixed location coding
; 128=Already in use
; 256=Tool in buffer
; 512=Blocked, not taken into account (because of PLC)
; 1024=Out (unload)
; 2048=In (loaded)
; 4096=Standard tool (constantly in the NCK)
; 8192=
; 16384=
TOOLSTATE=0

; Index of a defined location type
TOOLPLACESPEC=1

; 0=No monitoring (default)
```

```
; 1=Time monitoring
; 2=Workpiece count monitoring
TOOLMONITOR_MODE=0

; Tool search, number of replacement tool ($TC_TP10)
; 0 ... 32000
TOOLSEARCH_MODE=0
;!!!  Default setting of grinding-specific tool data at creation:
;!!!  If the machine operates with inch/mm conversion
;!!!  ($MN_CONVERT_SCALING_SYSTEM=1), the unit of length must be spe-
;!!!  cified!!!
; The following default values (TOOLGRIND..., if affected by the unit
; of length) are specified in relation to this basic unit of length:
; 0 = mm (default)
; 1 = Inch
TOOLGRIND_Default_Length_Unit=0

;    Spindle number       (wie $TC_TPG1)
TOOLGRINDspinNoDress=1

;    Chain rule    (as $TC_TPG2)
TOOLGRINDconnectPar=1050629
    ;1050629 binary: 0000 0000 0001 0000 0000 1000 0000 0101
    ;Bit0  =1 = Type
    ;Bit2  =1 = Geo-L1
    ;Bit11 =1 = Wear-L1
    ;Bit20 =1 = Base-L1

;   Minimum wheel radius (as $TC_TPG3)
TOOLGRINDminToolRadius=0

;   Minimum wheel width     (as $TC_TPG4)
TOOLGRINDminToolWide=0

;   Current width of grinding wheel (as $TC_TPG5)
TOOLGRINDactToolWide=0

;   Maximum grinding wheel speed         (as $TC_TPG6)
TOOLGRINDmaxRotSpeed=0

;   Maximum grinding wheel peripheral speed (as $TC_TPG7)
TOOLGRINDmaxTipSpeed=0

;   Inclination angle of inclined wheel (as $TC_TPG8)
TOOLGRINDinclAngle=0

;   Compensation parameter for grinding wheel peripheral speed
(as $TC_TPG9)
TOOLGRINDparamNrCCV=3
```

[TMMODES]

```
; 0=Do not delete tool automatically
; when it is unloaded (magazine list only). (default);
; 1=Delete tool automatically when it is unloaded (magazine list
; only)
DELETE_TOOL_ON_UNLOAD=0

; 0=Do not process cutting edge parameters outside the tool type
; (default)
; 1=Process cutting edge parameters outside the tool type
; (if not equal to 0)
EDGE_PARAMS_OUT_OF_TOOLTYPE=1

; 0=display: Tool size left, right, top, bottom (default)
; 1=display: left, right
SHOW_TOOLSIZE_ONLY_LEFT_RIGHT=0

; Tool size display:
; True=display (default)
; False=do not display
; is only used if SHOW_TOOLSIZE_ONLY_LEFT_RIGHT = 0
; (or default)
SHOW_TOOLSIZE_COMPONENTS=left:=True, right:=True, top:=True,
                           bottom:=True

; The function "Activate D check" refers to:
; -1=all magazines with distance relationship to spindle/toolholder
; (default)
; 1=current magazine only
DCHECK_ACTIVATE=-1

; The function "Activate D check" can be carried out automatically
; when the working offset list is opened
; False=function can only be activated via softkey (default)
; True=function is automatically carried out when the working offset
; list is opened
DCHECK_AUTO_ACTIVATE=False

; DCHECK_AUTO_ACTIVATE_MODE is only used if DCHECK_AUTO_ACTIVATE=True
; 0=the function "Activate D check" is carried out automatically if
; the working offset list is opened, even in the main screen (con-
; tains R Parameters softkey) (default)
; 1=The function "Activate D check" is carried out automatically if
; the working offset list is opened, but not in the main screen (con-
; tains R Parameters softkey). Same behavior as before implementation
; of DCHECK_AUTO_ACTIVATE_MODE (prior to HMI_ADV 06.03.19).

; Buffer location display:
; DB:   Original name from the magazine configuration from the
;       database, no language-specific texts
; DLL:  Name = text from the language -DLL + index; ; (default)
;       example: Spindle1, language-specific texts
NameOfBufferPlaceFrom=DB
```

```
; For display of the funcitons “Create tool edge” and “Delete tool
; edge” (only possible with multi-line display) in the main menu
; “Tool / magazine list”, the softkey “Tool details” can be replaced
; with the new softkey “Data management”.
; The “Tool details” softkey is saved with the same functionality
; behind the softkey “Data management”.
; False=“Tool details” remains active (default)
; True=“Data management” is activated
ACTIVATE_EDGE_MANAGEMENT_IN_LISTS=False

; If adapter data is activated in the NCK, the magazine list can
; either be displayed as a transforming or non-transforming list
; (softkey in the ETC menu). The type of display can be preset for
; MMC boot.
; False = Not transformed (default)
; True  = Transformed
START_MAGLIST_TRANSFORMED=False

; Softkey “Tool management” in the main menu “Parameters” allows a
; branch to the list set below.
; 0 = standard list depending on the respective NCK version / machine
; data (default)
; 1 = Magazine list
; 2 = Tool list
; 3 = Working offset list
START_LIST=0

; The setting inch/metric is considered for the code carrier
; !! CAUTION: The settings described here are only relevant if ma-
; chine data $MN_CONVERT_SCALING_SYSTEM=1 is set in the NC.
; The entry for DATABASE_LENGTH_UNIT is only analyzed if no unit was
; entered in the tool database, i.e. under normal circumstances once!
; If the NC is an older mode without the inch/metric conversion func-
; tionality or if $MN_CONVERT_SCALING_SYSTEM=0, the NC will operate
; without inch/metric conversions. Therefore no conversions are car-
; ried out in relation to the code carrier!!

; -1 =  No definition for inch/metric in relation to code carrier
;       default). The setting $MN_CONVERT_SCALING_SYSTEM=1 in the NC
;       means that inch/metric conversion is to be used for machin-
;       ing. Therefore, the user must define in which unit the data
;       is present on the code carrier or should be written to it. As
;       with setting -1 this does not take place, all softkeys for
;       code carrier activities are disabled.
; 0 =   mm. It is assumed that storage for the affected data was or
;       is to be in the unit “mm” on the code carrier. If “inch” is
;       set in the NCK, then all softkeys are disabled that start
;       code carrier functions.
; 1 =   inch. It is assumed that storage for the affected data was or
;       is to be in the unit “inch” on the code carrier. If “mm” is
;       set in the NCK, then all softkeys are disabled that start
;       code carrier functions.
```

```
DATABASE_LENGTH_UNIT=-1

The setting inch/metric is considered for the code carrier
; The setting inch/metric is considered for the code carrier
; !! CAUTION: The settings described here are only relevant if ma-
; chine data $MN_CONVERT_SCALING_SYSTEM=1 is set in the NC.
; If the NC is an older mode without the inch/metric conversion func-
; tionality or if $MN_CONVERT_SCALING_SYSTEM=0, the NC will operate
; without inch/metric conversions. Therefore no conversions are car-
; ried out in relation to the code carrier!!
; -1 =  inch/metric is ignored (default). The data exchange between
;       code carrier and NCK/MMC takes place without taking into ac-
;       count inch/metric. Behavior as up to now.
; 0 =   mm. All affected data is written to the code carrier as mm
;       values in future.
;       This is taken into account for future data transfer between
;       code carrier and NC.
; 1 =   inch. All affected data is written to the code carrier as
;       inch values in future.
;       This is taken into account for future data transfer between
;       code carrier and NC.
CODECARRIER_LENGTH_UNIT=-1

; Tool status: If a tool is removed from the NCK and transferred to
; an external medium (tool cabinet, code carrier, SINCOM), then you
; can use the following screens to specify which tool status bits are
; to be saved
; Code carrier: Since 1 byte was entered for the tool status in the
; standard conversion file wkonvert.txt and until now max. 92 has
; been written to the code carrier, CODECARRIER_TOOLSTATE_MASK is
; assigned the default value 92.
; If the value for CODECARRIER_TOOLSTATE_MASK is expanded, the size
; of the dialog variable T9 must be adapted accordingly in wkon-
; vert.txt.
; 1=Active Tool
; 2=Allowed
; 4=Disabled
; 8=Measured
; 16=Warning limit reached
; 32=In change
; 64=Fixed place coding
; 128=Was used
; 256=Tool in buffer
; 512=Disabled, ignored (because of PLC)
; 1024=Out (unload)
; 2048=in (Load)
; 4096=Regular tool (permanent in NCK)
; 8192=
; 16384=
; Default is 4828 (4+8+16+64+128+512+4096),
; For code carrier 92 (4+8+16+64)
```

```
CABIB_TOOLSTATE_MASK=4828
SINCOM_TOOLSTATE_MASK=4828
CODECARRIER_TOOLSTATE_MASK=92
; Tool search: processing $TC_TP10 in conjunction with the tool cabi-
; net/catalog
; 0 =   (default) If a tool is brought from the tool cabinet into the
;       NCK, then value “No. replacement tool ($TC_TP10)” is not
;       uploaded to the NCK. ($TC_TP10) is set in the NCK to 0.
; 1 =   The value “No. replacement tool ($TC_TP10)” is transferred
;       from the tool cabinet to the NCK and displayed in the Tool
;       catalog/cabinet screens.
TOOLSEARCH_TC_TP10_FROM_DB=0
```

[General]

```
; Settings for “Write current data for tool management operator
; interface to NCDDE variables, when there is a change to WIZARD
; pictures or the WIZARD softkeys were activated”:
; All settings must be set in one single line of the name parameter.
; An option is activated via the value “True” and deactivated by the
; value “False” or because the name parameter is missing in the line.
; “EnableAllTogetherWriteToNcdde := True”: All data in one single
; NCDDE variable
; “EnableSingleWriteToNcdde := True”: One separate NCDDE variable for
; each data
;      Both settings can be simultaneously active.
;      When none of the two options is active, the NCDDE variables
;      are not written to.
; “WriteChangesWhenStateChanged := True”: The data is written each
; time a softkey is activated; this does not apply to WIZARD
; softkeys.
;
HMICurDataInterface = EnableAllTogetherWriteToNcdde := True,
EnableSingleWriteToNcdde := True, WriteChangesWhenStateChanged :=
False
; Application of $MM_WRITE_TOA_FINE_LIMIT and
; $MM_USER_CLASS_WRITE_FINE to the geometrical data and basic values
; for the cutting edge data
UseFineLimitForToolGeoAndAdapt=False ;Default
;UseFineLimitForToolGeoAndAdapt=True
; 1: Read language-specific INI files (language\patm_xx.ini)
; (default)
; 0: Do not read
ReadLanguageIni=1
```

```
SearchPlaceMethod=NoInternalTool ; Don’t modify!

; Tool details forms: colors for mixed adapter transformed / untrans-
; formed display:
; Hex values, 8 characters per color(SSBBGGRR where SS=System,
; BB=Blue, GG=Green, RR=Red)
; 4 Values for:
; TransformedText, TransformedBackground, NotTransformedText, Not-
; TransformedBackground
; DetailsMixedTrafoColors=WinTxt,  li-brown,WinTxt  ,li-blue

DetailsMixedTrafoColors =80000008,00008080,80000008,00FFFF00

; Allow display and edit of all 3 len parameters L1 L2 L3 in cut edge
; geo, cut edge wear, SC, EC independently of tool type and indepen-
; dently of $SC_TOOL_LENGTH_CONST and $SC_TOOL_LENGTH_TYPE in tool
; management lists, detail forms, cabinet and catalog.

AllwaysAllowL1L2L3IO=False ;default

; AllwaysAllowL1L2L3IO=True
; Allow change of tool state bit 8 (least significant bit is bit 1)
; (TC_TP8_8 “ToolState Used, Tool was being used”) via GUI (list and
; details forms) for ncu tools.
; default: FALSE

AllowChangeOfTC_TP8_8 = False ;default

; AllowChangeOfTC_TP8_8 = True
```

[GeneralSettingsForMagAndToolList]

```
; If MagPlaceState_Lang_12345678 and ToolState_Lang_12345678 are not
; defined here or or are equal to “<Empty>”, then the
; language-specific values are shown in the magazine and tool list
; from pa_xx.dll.

; If values are set here and in the section “[General]” the entry
; “ReadLanguageIni” equals 1, then the text is searched in the files
; mmc2\language\patm_gr.ini, user\language\patm_gr.ini etc. in the
; same section as here. The name for the entry that is used in the
; language-specific file is the value of the entry from paramtm.ini.
; If an entry is found in the language-specific file, then this is
; used as the text.
; If no entry is found or ”...=<Empty> is found, then the value from
; the file paramtm.ini is used as the text.

; The 8 characters in MagPlaceState_Lang_12345678 and
; ToolState_Lang_12345678 correspond to the 8 states of magazine
; location and tool and are shown as values for the location and tool
; states in the magazine list and in the tool list.
; Example: ToolState_Lang_12345678=12345678_ToolState_Lang
ToolState_Lang_12345678=<Empty> ; use language-DLL

; MagPlaceState_Lang_12345678=12345678_MagPlaceState_Lang ; use
; patm_*.ini
```

```
; The new magazine location status bits can be displayed in each list
; screen. The parameters are set in the sections:
“[1_MagList]”, “[2_MagList]”, “[3_MagList]”
“[1_ToolList]”, “[2_ToolList]”, “[3_ToolList]”
“[1_ActList]”, “[2_ActList]”, “[3_ActList]”
; Entries are for example:
12=TC_MPP4_9,  1, TC_MPP4_9   ;PlaceStatus Left,
                              ;Reserved in left half location
13=TC_MPP4_10, 1, TC_MPP4_10  ;PlaceStatus Right,
                              ;Reserved in right half location
14=TC_MPP4_11, 1, TC_MPP4_11  ;PlaceStatus Top,
                              ;Reserved in upper half location
15=TC_MPP4_12, 1, TC_MPP4_12  ;PlaceStatus Bottom,
                              ;Reserved in lower half location
16=TC_MPP4_13, 1, TC_MPP4_13  ;PlaceStatus Bit 13 of 1 to 16
17=TC_MPP4_14, 1, TC_MPP4_14  ;PlaceStatus Bit 14 of 1 to 16
18=TC_MPP4_15, 1, TC_MPP4_15  ;PlaceStatus Bit 15 of 1 to 16
19=TC_MPP4_16, 1, TC_MPP4_16  ;PlaceStatus Bit 16 of 1 to 16

; Example: MagPlaceState_Lang_12345678=12345678_MagPlaceState_Lang
MagPlaceState_Lang_12345678=<Empty>

; For alphanumeric columns in a list:
; Width of a character in “twips”. The approximate column width is
; calculated by multiplying the values entered here by the number of
; characters from column parameterization
ColumnWidthTwipsPerAlphaCharacter=140

; For numerical columns in a list:
; Width of a character in “twips”. The approximate column width is
; calculated by multiplying the values entered here by the number of
; characters from column parameterization
ColumnWidthTwipsPerNumericCharacter=100
```

```
; Number of data elements in the magazine list or tool list at each
; internal data scan.
; Range: 1 to 27, default 18.
; The data-scanning rate has been improved as of version P4.3.8.
; There is no reaction to softkey operation as long as internal data
; scanning is taking place. This time should therefore not be longer
; than 1 second.
; The value from “NumLinesPerReq” is used for scanning the data in
; the background from a complete list once a list has been selected
; per softkey or following start-up.
; The number of visible lines in the list is used for getting the
; data when the displayed data are updated in the list following a
; data change or by scrolling and the number of visible lines in the
; list is less than NumLinesPerReq.
; If the data exchange between MMC and NC is too slow (for the NCU
; 810 D), then this value shall be lowered to 17 in order to reach a
; response time of about 1s when fetching the data from a complete
; list in the background.
; This setting applies for all lists where no individual settings are
; made.
; Individual settings are useful if the list contains a high number
; of columns.
; In this case the value should be lowered to about 10 or 5. Too many
; columns in a list is not advisable if the list is used frequently.
; This is because it would take too long to fetch the data for the
; entire list and the user would have to wait a long time for the
; display.
; It is necessary in order to set a single list to individual
; settings to add the line “NumLinesPerReq” to the appropriate list
; in the section (e.g. [2_ToolList])

Use 27 for NumLinesPerReq=27 from Version P4.3.8!

; Up until P4.3.8 the value 7 worked well.

; Width of the bitmap image for the current tool and the current
; tool-magazine location in the displayed lists. Unit: Number of
; characters; the width of a character is defined by
; “ColumnWidthTwipsPerAlphaCharacter” or
; “ColumnWidthTwipsPerNumericCharacter”. Smallest value: 1, largest
; value: 32, default value: 5. User-defined images can be used as
; well: If the file name is given without the path or is given with
; the MMC2 path, then the bitmap file is searched in the directories
; “user”, “oem”, “add_on” and “mmc2”. The first one found is then
; used. We advise not to generate user-defined bitmaps that are too
; large. the The ratio width to height should correspond
; approximately to the display in the lists so that the presentation
; given will not be distorted.
;
; Use WidthOfActBitmapsInCharacters = 7, if $SC_WEAR_TRANSFORM <> 0
; and G56-Reset-Value <> TOWSTD

WidthOfActBitmapsInCharacters = 5
```

```
WhichActChannelText=ChannelName,4

; default, show first 4 characters of channel name in lists actual
; tool indicator
; WhichActChannelText=ChannelNumber ;show channel number in lists
; actual tool indicator
; If you need to display more characters of channel name, please in-
; crease “WidthOfActBitmapsInCharacters”.
; For better readability modify bitmaps or use lpaat.bmp,
; lpaatd0.bmp, lpaatdl0.bmp, lpapt.bmp, lpaptd0.bmp, lpaptdl0.bmp,
; lpaap.bmp.
; Further you can define a extra column for bitmap exclusive display,
; which will not contain any other data.
; Example:
; [3_ActList]
; ShowActToolCol = 1
; column number where bitmap is displayed
; 1= NoData, 0, Activity
; column 1 is an empty column for display of channel activity;
; column with is 0 + “WidthOfActBitmapsInCharacters”;
; column header text is “Activity” or langauge dependent text;

; File name of the bitmap for the current tool / DNo / DL where D <>
; 0 and DL <> 0
ActToolBitmap = paat.bmp

; File name of the bitmap for the current tool / DNo / DL with D = 0
; in magazine list and tool list.
; Such cutting edges are not marked in the working offset list.
ActToolZeroDBitmap = paatd0.bmp

; File name of the bitmap for the current tool / DNo / DL with current DL = 0.
;
ActToolZeroDLBitmap = paatdl0.bmp

; as for ActToolBitmap for the programmed tool
ProgToolBitmap = papt.bmp
; as for ActToolZeroDBitmap for the programmed tool

ProgToolZeroDBitmap = paptd0.bmp

; as for ActToolZeroDBitmap for the programmed tool
ProgToolZeroDLBitmap = paptdl0.bmp

; File name of the bitmap for the current magazine location
ActPlaceBitmap = paap.bmp

; Display whether the current magazine is unassigned or disabled for
; loading/unloading tools
ShowMagFreeLocked = False
; ShowMagFreeLocked = True

; Name of the bitmap file to display whether the current magazine is
; free for loading/unloading tools
MagFreeBitmap = magfree.bmp
```

```
; Name of the bitmap file to display whether the current magazine is
; disabled for loading/unloading tools
MagLockBitmap = maglock.bmp

; File name of bitmap used in lists to show: G56-reset-value = TOWMCS
; = G56-current-value
G56ResetTOWMCSEqualCurrBitmap = pemcs.bmp

; File name of bitmap used in lists to show: G56-reset-value = TOWMCS
; = G56-current-value
G56ResetTOWWCSEqualCurrBitmap = pewcs.bmp

; File name of bitmap used in lists to show: G56-reset-value = TOWSTD
; = G56-current-value
G56ResetTOWSTDEqualCurrBitmap = pestd.bmp

; File name of bitmap used in lists to show: G56-reset-value = TOWMCS
; <> G56-current-value
G56ResetTOWMCSUnequalCurrBitmap = pumcs.bmp

; File name of bitmap used in lists to show: G56-reset-value = TOWMCS
; <> G56-current-value
G56ResetTOWWCSUnequalCurrBitmap = puwcs.bmp

; File name of bitmap used in lists to show: G56-reset-value = TOWSTD
; <> G56-current-value
G56ResetTOWSTDUnequalCurrBitmap = pustd.bmp

; In order to prevent excessive horizontal scrolling, cursor moves
; automatically to leftmost column in lists during the following ac-
; tions:
; Magazine lists: Softkey “Load” and softkeys “Start” and “Abort” in
; tool loading mode;
; Tool lists: Softkey “New tool”.
CursorMovesLeftmostBySomeActions=True  ;default
;CursorMovesLeftmostBySomeActions=False

; In magazine list forms change softkey “Next Mag” to softkey “Maga-
; zine Selection”, to activate vertical softkeys in an additional
; state for magazine selection.
; This helps to prevent users from excessive use of “next mag” soft-
; key if a lot of magazins ara available.
; You can define shortcut softkeys for up to 5 favorite magazines per
; TOA using section [ShortcutSoftKeysForMagSelect] in paramtm.ini and
; patm_??.ini.
MagListMagSelectSoftkey=NextMag   ;default
;MagListMagSelectSoftkey=SelectMag

; Magazine list: Display buffer initially
DisplayBufferInMagList = InitialVisible
```

```
                 ; Default behaviour as of version 6.4.1.
                 ; Whenever another magazine is visited in magazine
                 ; list:
                 ; automatically show its buffer too.
                 ; The buffer is hidden temporarily after the user
                 ; has pressed the softkey “Buffer”.
;DisplayBufferInMagList = InitialNotVisible
                 ; Previous behaviour prior to Version 6.4.1.
                 ; In magazine list:
                 ; At the beginning no buffer is shown.
                 ; After a magazine which doesn’t have a buffer was
                 ; visited, then no buffer is always shown for other
                 ; magazines too.
                 ; The buffer is shown temporarily when the user
                 ; presses the “Buffer” softkey.
```

[SoftKeysForMagAndToolList]

```
; Text for the softkeys of magazine lists and tool lists
; The locale mechanism is used.
; For an explanation of this mechanism refer to the section
; “[GeneralSettingsForMagAndToolList]”, entry
; “MagPlaceState_Lang_12345678” or “second “TC_TP2””

1_MagList=M1
2_MagList=M2
3_MagList=M3

1_ToolList=T1
2_ToolList=T2
3_ToolList=T3

1_ActList=A1
2_ActList=A2
3_ActList=A3
```

```
[FormTitles]

; List display titles for untransformed display of data with locale
; mechanism and transformation mechanism.
; With the transformed display, the substitute for the text code is
; taken from section “[TrafoFormTitles]”.
; If “ReadLanguageIni=1” applies in “[General]”, then the text from
; the locale file is read (e.g. language\patm_gr.ini), see section
; “[FormTitles]”.
; Whatever the case, the text parameters should be set for all lists
; of displays, both for the transformed as well as for the
; untransformed display.
; Pay particular attention to the working offset lists: Currently
; these lists are only output with transformed data. In this case the
; list display titles from “[TrafoFormTitles]” are used.

1_MagList=M1
2_MagList=M2
3_MagList=M3

1_ToolList=T1
2_ToolList=T2
3_ToolList=T3

1_ActList=A1
2_ActList=A2
3_ActList=A3


[TrafoFormTitles]

; List display titles for untransformed data display with locale
; mechanism and transformation mechanism.
; If “ReadLanguageIni=1” applies in “[General]”, then the text is
; read from the locale file; see section “[FormTitles]”.

M1=TM1
M2=TM2
M3=TM3

T1=TT1
T2=TT2
T3=TT3

A1=TA1
A2=TA2
A3=TA3

[SearchOfMagPlaces]

; Defines how the location search is to take place

; Softkey text for 1st location search, locale-dependent
1_SoftkeyText=EL1

; Half location; Left, right, top, bottom
1_ToolSizeLRTB=1,1,1,1
```

```
; Location type number for location search
1_PlaceTypeNo=1

; Softkey text for 2nd location search, locale-dependent
2_SoftkeyText=EL2

; Half location; Left, right, top, bottom
2_ToolSizeLRTB=1,2,1,1

; Location type number for location search
2_PlaceTypeNo=1

; Softkey text for 3rd location search, locale-dependent
3_SoftkeyText=EL3

; Half location; Left, right, top, bottom
3_ToolSizeLRTB=2,2,1,1

; Location type number for location search
3_PlaceTypeNo=1

; Softkey text for 3rd location search, locale-dependent
4_SoftkeyText=EL4

; Half location; Left, right, top, bottom
4_ToolSizeLRTB=1,2,1,1

; Location type number for location search
4_PlaceTypeNo=2

; Tool OEM Data and Tool Application Data
; Help for UserDataParamIO lines:
; Format of data in magazine list and tool list.

; This format is used only to display data and, in some cases, to
; input data in the HMI, magazine list and tool list. In the NC the
; data format of the OEM data is "float".

; In parameter lines "named parameters" are used. Names are separated
; from the value by ":=". Parameters are separated by ",".
; Blanks are allowed on the left and right of the parameter name,
; ":=", value and ",".

; Example for the syntax of lines in the OEM data format:
; "[ToolParams]"
;   "UserDataParamIO5=   Type:=Float, Res:=2,
;                        Min:=-9999, Max:=9999 ;Comment"
;   "UserDataParamIO6=   Typ:=Int, Min:=-99, Max:=99"
;   "UserDataParamIO7=   Type:=Int, Min:=0, Max:=1"
; "5":                   Number of the OEM data
;  "Type:=...":          Defines the type of OEM data
;                        Default: "Float"
```

```
;    "Float":             Floating point as used in the NC.
;                         The number of places after the decimal point
;                         depends on $MM_DISPLAY_RESOLUTION in mmc.ini
;                         and from "res:=..." in this parameter line.
;    "Int":               Integer (range: -999999999 to 999999999)
; "Res:=2":               Resolution, accuracy: Number of places after
;                         the comma if "Type:=Float" (floating-point
;                         number).
;                         Range from 0 to 6,
;                         Default is $MM_DISPLAY_RESOLUTION.
;                         If "Res:=..." is greater than
;                         $MM_DISPLAY_RESOLUTION, only the places
;                         $MM_DISPLAY_RESOLUTION are displayed.
;  "Min:=-9999":          For the entry: Minimum value
;  "Max:=-9999":          For the entry: Maximum value:
;  ";comment":            Comment

; Lines UserDataParamName:  Locale-specific mechanism
; Lines UserDataParamSize:  Locale-specific mechanism
; Lines UserDataParamSizex: Locale-specific mechanism
```

[ToolParams]

```
UserDataParamName1  = TC_TPC1
UserDataParamName2  = TC_TPC2
UserDataParamName3  = TC_TPC3
UserDataParamName4  = TC_TPC4
UserDataParamName5  = TC_TPC5
UserDataParamName6  = TC_TPC6
UserDataParamName7  = TC_TPC7
UserDataParamName8  = TC_TPC8
UserDataParamName9  = TC_TPC9
UserDataParamName10 = TC_TPC10

UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Type:=Float, Res:=2
; UserDataParamIO2 = Type:=Int

UserDataParamSize = TC_TPC_UNIT

UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_TPC1_UNIT
; ...
; UserDataParamSize10=TC_TPC10_UNIT

ApplDataParamName1  = TC_TPCS1
ApplDataParamName2  = TC_TPCS2
ApplDataParamName3  = TC_TPCS3
ApplDataParamName4  = TC_TPCS4
ApplDataParamName5  = TC_TPCS5
ApplDataParamName6  = TC_TPCS6
ApplDataParamName7  = TC_TPCS7
ApplDataParamName8  = TC_TPCS8
ApplDataParamName9  = TC_TPCS9
ApplDataParamName10 = TC_TPCS10
```

```
ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Type:=Float, Res:=2
; ApplDataParamIO2 = Type:=Int

ApplDataParamSize = TC_TPCS_UNIT

ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_TPCS1_UNIT
; ...
; ApplDataParamSize10=TC_TPCS10_UNIT

; Example for the syntax of lines in the OEM data format:
; "[ToolEdgeParams]"
;   "UserDataParamIO5=   Type:=Float, Res:=2,
;                        Min:=-9999, Max:=9999 "Comment"
;   "UserDataParamIO6=   Type:=Int, Min:=-99, Max:=99"
;  "UserDataParamIO7=    Type:=Int, Min:=0, Max:=1"
; "5":                   Number of the OEM data
;  "Type:=...":          Defines the type of OEM data
;                        Default: "Floatn"
;    "Float":            Floating point as used in the NC.
;                        The number of places after the decimal point
;                        depends on $MM_DISPLAY_RESOLUTION in mmc.ini
;                        and from "res:..." in this parameter line.
;    "Int":              Integer (range: -999999999 to 999999999)
;  "Res:=2":             Resolution, accuracy: Number of places after
;                        the comma if "Type:=Float" (floating-point
;                        number).
;                        Range from 0 to 6,
;                        Default is $MM_DISPLAY_RESOLUTION.
;                        If "Res:=..." is greater than
;                        $MM_DISPLAY_RESOLUTION, only the places
;                        $MM_DISPLAY_RESOLUTION are displayed.
;  "Min:=-9999":         For the entry: Minimum value
;  "Max:=-9999":         For the entry: Maximum value:
;  ";comment":           Comment;
; Lines EdgeParamName...:
; Example:
; EdgeParamNameLLen1=TC_DP3
;                        Defines the text where the cutting-edge
;                        parameters and additive offset parameters
;                        are displayed in the tool detail displays.

;                        The sequence of the EdgeParamName... lines
;                        corresponds to the sequence in the displays.
;                        The sequence in the displays cannot be
;                        influenced by any change in the sequence in
;                        paramtm.ini.
```

```
;                        We have used the character sequence “TP_DP
;                        ...” for the values of EdgeParamName...to
;                        illustrate the relationship between text and
;                        the associated NCK variables. Other
;                        sequences of characters can also be used
;                        here because this text reference only
;                        defines a parameter header and it is not
;                        defined which data are read out from the
;                        NCK.

;                        The value of EdgeParamName... is written
;                        directly into the display if [General]
;                        ReadLanguageIni =0 and in the untransformed
;                        presentation is active in tool detail
;                        display.

;                        If [General] ReadLanguageIni =1 and the
;                        untransformed presentation is active in the
;                        tool detail display, then the value of
;                        EdgeParamName... is used as the access code
;                        to read the text from the locale-specific
;                        INI file (language\patm_xx.ini, section
;                        [ToolEdgeParams], access code “TC_DP3” in
;                        this example).

;                        If the transformed presentation is active in
;                        the tool detail display, then the value of
;                        EdgeParamName... is used as the access code
;                        for the section
;                        [ToolEdgeParamsTrafoTextReplace] in
;                        paramtm.ini. The value found here is then
;                        used as the text or access code (depending
;                        on [General] ReadLanguageIni) to determine
;                        the output text for the detailed tool
;                        displays.

;                        If no entry is found in the INI files, then
;                        the text is taken from the Resource file
;                        language\pa_xx.dll.

; Lines UserDataParamName:  Locale-specific mechanism
; Lines UserDataParamSize:  Locale-specific mechanism
; Lines UserDataParamSizex: Locale-specific mechanism
; Lines EdgeParamName...:  untransformed / transformed mechanism and
                             locale mechanism
```

```
[ToolEdgeParams]
; Length 1
EdgeParamNameLLen1 = TC_DP3
; Length 2
EdgeParamNameLLen2 = TC_DP4
; Length 3
EdgeParamNameLLen3 = TC_DP5
; Radius l1
EdgeParamNameRLen1 = TC_DP8
; Radius l2
EdgeParamNameRLen2 = TC_DP9
; Radius r1
EdgeParamNameRRad1 = TC_DP6
; Radius r2
EdgeParamNameRRad2 = TC_DP7
; Angle1
EdgeParamNameAng1 = TC_DP10
; Angle2
EdgeParamNameAng2 = TC_DP11

UserDataParamName1  = TC_DPC1
UserDataParamName2  = TC_DPC2
UserDataParamName3  = TC_DPC3
UserDataParamName4  = TC_DPC4
UserDataParamName5  = TC_DPC5
UserDataParamName6  = TC_DPC6
UserDataParamName7  = TC_DPC7
UserDataParamName8  = TC_DPC8
UserDataParamName9  = TC_DPC9
UserDataParamName10 = TC_DPC10

UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Type:=Float, Res:=2
; UserDataParamIO2 = Type:=Int

UserDataParamSize = TC_DPC_UNIT

UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_DPC1_UNIT
; ...
; UserDataParamSize10=TC_DPC10_UNIT

ApplDataParamName1  = TC_DPCS1
ApplDataParamName2  = TC_DPCS2
ApplDataParamName3  = TC_DPCS3
ApplDataParamName4  = TC_DPCS4
ApplDataParamName5  = TC_DPCS5
ApplDataParamName6  = TC_DPCS6
ApplDataParamName7  = TC_DPCS7
ApplDataParamName8  = TC_DPCS8
ApplDataParamName9  = TC_DPCS9
ApplDataParamName10 = TC_DPCS10
```

```
ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Type:=Float, Res:=2
; ApplDataParamIO2 = Type:=Int

ApplDataParamSize = TC_DPCS_UNIT

ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_DPCS1_UNIT
; ...
; ApplDataParamSize10=TC_DPCS10_UNIT
```

[ToolEdgeParamsTrafoTextReplace]

```
; Transformed length 1
TC_DP3 = TTC_DP3
; Transformed length 2
TC_DP4 = TTC_DP4
; Transformed length 3
TC_DP5 = TTC_DP5
; Transformed radius l1
TC_DP8 = TTC_DP8
; Transformed radius l2
TC_DP9 = TTC_DP9
; Transformed radius r1
TC_DP6 = TTC_DP6
; Transformed radius r2
TC_DP7 = TTC_DP7
; Transformed angle1
TC_DP10 = TTC_DP10
; Transformed angle2
TC_DP11 = TTC_DP11
```

[CuttEdgeSupervisionOEM]

```
; [CuttEdgeSupervisionOEM] cutting edge monitoring OEM data and
; application data
; [MagazineOEM] magazine OEM data and application data
; [MagazineLocOEM] magazine location OEM data and application data
; Help for the lines “UserDataParamIO”:
; Data format for the data display in magazine list and tool list
; This format is only used for the display
; and
; in a number of instances as well for entering data.

; “Name parameters” are used in parameter lines.
; The names are separated from the parameter value by “: = ”.
; The parameters themselves are separated one each other by “,”;
; blanks are permitted to the left and to the right of
; parameter names, “:=”, parameter value and “,”.

; Example for the syntax of lines in the OEM data format:
; “[CuttEdgeSupervisionOEM]”
; “UserDataParamIO5=Min:=-9999, Max:=9999 ;comment”
; “5”:                     Number of the OEM data
; “Min:=-9999”:            For the entry: Minimum value
```

```
; "Max:=-9999":           For the entry: Maximum value:
; ";comment": Comment

; Lines UserDataParamName:  Locale-specific mechanism
; Lines UserDataParamSize:  Locale-specific mechanism
; Lines UserDataParamSizex:  Locale-specific mechanism

[CuttEdgeSupervisionOEM]
UserDataParamName1  = TC_MOPC1
UserDataParamName2  = TC_MOPC2
UserDataParamName3  = TC_MOPC3
UserDataParamName4  = TC_MOPC4
UserDataParamName5  = TC_MOPC5
UserDataParamName6  = TC_MOPC6
UserDataParamName7  = TC_MOPC7
UserDataParamName8  = TC_MOPC8
UserDataParamName9  = TC_MOPC9
UserDataParamName10 = TC_MOPC10

UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Min:=-4, Max:=6
; UserDataParamIO2 = Min:=-10, Max:=122

UserDataParamSize = TC_MOPC_UNIT

UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_MOPC1_UNIT
; ...
; UserDataParamSize10=TC_MOPC10_UNIT

ApplDataParamName1  = TC_MOPCS1
ApplDataParamName2  = TC_MOPCS2
ApplDataParamName3  = TC_MOPCS3
ApplDataParamName4  = TC_MOPCS4
ApplDataParamName5  = TC_MOPCS5
ApplDataParamName6  = TC_MOPCS6
ApplDataParamName7  = TC_MOPCS7
ApplDataParamName8  = TC_MOPCS8
ApplDataParamName9  = TC_MOPCS9
ApplDataParamName10 = TC_MOPCS10

ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Min:=-4, Max:=6
; ApplDataParamIO2 = Min:=-10, Max:=122

ApplDataParamSize = TC_MOPCS_UNIT

ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_MOPCS1_UNIT
; ...
; ApplDataParamSize10=TC_MOPCS10_UNIT
```

[Magazine0EM]

```
UserDataParamName1  = TC_MAPC1
UserDataParamName2  = TC_MAPC2
UserDataParamName3  = TC_MAPC3
UserDataParamName4  = TC_MAPC4
```

```
UserDataParamName5  = TC_MAPC5
UserDataParamName6  = TC_MAPC6
UserDataParamName7  = TC_MAPC7
UserDataParamName8  = TC_MAPC8
UserDataParamName9  = TC_MAPC9
UserDataParamName10 = TC_MAPC10

UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Min:=-22, Max:=24
; UserDataParamIO2 = Min:=-10, Max:=162

UserDataParamSize = TC_MAPC_UNIT

UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_MAPC1_UNIT
; ...
; UserDataParamSize10=TC_MAPC10_UNIT

ApplDataParamName1  = TC_MAPCS1
ApplDataParamName2  = TC_MAPCS2
ApplDataParamName3  = TC_MAPCS3
ApplDataParamName4  = TC_MAPCS4
ApplDataParamName5  = TC_MAPCS5
ApplDataParamName6  = TC_MAPCS6
ApplDataParamName7  = TC_MAPCS7
ApplDataParamName8  = TC_MAPCS8
ApplDataParamName9  = TC_MAPCS9
ApplDataParamName10 = TC_MAPCS10

ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Min:=-22, Max:=24
; ApplDataParamIO2 = Min:=-10, Max:=162

ApplDataParamSize = TC_MAPCS_UNIT

ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_MAPCS1_UNIT
; ...
; ApplDataParamSize10=TC_MAPCS10_UNIT
```

[MagazineLocOEM]

```
UserDataParamName1  = TC_MPPC1
UserDataParamName2  = TC_MPPC2
UserDataParamName3  = TC_MPPC3
UserDataParamName4  = TC_MPPC4
UserDataParamName5  = TC_MPPC5
UserDataParamName6  = TC_MPPC6
UserDataParamName7  = TC_MPPC7
UserDataParamName8  = TC_MPPC8
UserDataParamName9  = TC_MPPC9
UserDataParamName10 = TC_MPPC10

UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Min:=-42, Max:=62
; UserDataParamIO2 = Min:=-210, Max:=712

UserDataParamSize = TC_MPPC_UNIT
```

```
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_MPPC1_UNIT
; ...
; UserDataParamSize10=TC_MPPC10_UNIT

ApplDataParamName1  = TC_MPPCS1
ApplDataParamName2  = TC_MPPCS2
ApplDataParamName3  = TC_MPPCS3
ApplDataParamName4  = TC_MPPCS4
ApplDataParamName5  = TC_MPPCS5
ApplDataParamName6  = TC_MPPCS6
ApplDataParamName7  = TC_MPPCS7
ApplDataParamName8  = TC_MPPCS8
ApplDataParamName9  = TC_MPPCS9
ApplDataParamName10 = TC_MPPCS10

ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Min:=-42, Max:=62
; ApplDataParamIO2 = Min:=-210, Max:=712

ApplDataParamSize = TC_MPPCS_UNIT

ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_MPPCS1_UNIT
; ...
; ApplDataParamSize10=TC_MPPCS10_UNIT
```

Parameterization of the individual magazine, tool and working offset lists

```
; Changes in the following sections can determine which data is
; displayed in the individual magazine, tool and working offset
; lists:
; [1_MagList],      [2_MagList],     [3_MagList],
; [1_ToolList],     [2_ToolList],    [3_ToolList],
; [1_ActList],      [2_ActList],     [3_ActList].

; In these sections you can define the number of columns not
; displaced (i.e. always visible) by horizontal navigation
; (“scrolling”) (“NoOfFixedColumns=m”) as well as the number of
; individual columns (“1=...”, “2=...”,...).

; The column number (number in front of “=”) can take a value between
; 1 and 1000.
; The maximum number of columns in a list is about 90, whereby for
; 90 columns in a list however, the rate of display is slower and the
; user has to scroll horizontally in order to see all the columns
; that can be displayed. This means that this limit is not normally
; reached.

; Gaps are allowed between the numbers in the sequence of column
; numbers.

; If you want to deactivate a predefined column in mmc2\paramtm.ini,
; you can insert the corresponding entry with the value “<Empty>”.
```

```
; Specify "...=<EndOfList>" to define the end of the list. This
; increases the speed when reading the INI files once tool management
; has been started.

; Example for the syntax of a column-definition line:
; "2=TC_TP2,11,TC_TP2   ;WzIdent"
; "2":               Number of the entry,
; First "TC_TP2":    determines which NC data in the list of columns
;                    will be displayed.
;                    The sequence of characters TC_TP2 corresponds to
;                    an NCK variable, refer to the NC Programming
;                    Guide. The specified sequences of characters are
;                    described in paramtm.txt.

;                    New in Version P5:
;                    If "MultiLine=SINGLE" is in a list definition
;                    section of a magazine or tool list, then the
;                    cutting-edge number can be specified by adding
;                    "@Ee", whereby "e" denotes the cutting-edge
;                    number (range from 1 to maximum number of
;                    cutting edges per tool) for all cutting edge
;                    data.
;                    This applies to the following data:
;                    Cutting edge data              TC_DPp@Ee
;                    Cutting edge monitoring data   TC_MOPp@Ee
;                    OEM cutting edge data          TC_DPCp@Ee
;                    Freely assignable D No.        TC_DPCE@Ee
;                    Additive offset                TC_SCPz@Ee
;                    Setup compensation             TC_ECPz@Ee

;                    If "@Ee" is not specified in these columns, then
;                    the data for cutting edge 1 is used.
;                    This method of proceeding is compatible with the
;                    earlier versions of P5.

;                    In order to avoid confusion, you should give the
;                    cutting-edge number in the title text of the
;                    respective column for these cutting edges.

;                    "@Ee" may not be specified in magazine and tool
;                    lists with "Multitime=MULTI" or with working
;                    offset list. These values automatically display
;                    the data for the current cutting edges.
; "11":              Approximate width of the column in characters,
;                    relative to "[GeneralSettingForMagAndToolList]",
;                    entries "ColumnWidthTwipsPer AlphaCharacter" and
;                    "ColumnWidthTwipsPerNumericCharacter"
```

```
; Second "TC_TP2"    Column header text or code for text.
;                    If the entry in the section "[General]" is
;                    "ReadLanguageIni"="1", then the column header
;                    text is searched in the files
;                    mmc2\language\patm_gr.ini,
;                    user\language\patm_gr.ini etc. in the section
;                    "[ListColumnHeaderText]", entry "TC_TP2" (in
;                    this example).
;                    The character sequence "gr" in "patm_gr.ini"
;                    depends on the particular language (see mmc.ini,
;                    "[LANGUAGE]", entry "Language=...").
;                    If the entry is found in Language\patm_gr.ini
;                    then this is used as the column header text.
;                    If no entry is found or the text <Empty> is
;                    found, then the value from the file paramtm.ini
;                    is used as the column-header text.
;                    If transformed data are displayed, then the
;                    column-header text or its access code is
;                    "transformed", in that the corresponding
;                    assignment of the section
;                    [ListColumnHeaderTrafoTextReplace] is used for
;                    the replacing operation.
;                    (This way both the transformed/untransformed
;                    mechanism as well as the locale mechanism are
;                    used.)
; ";WzIdent":        ";" Introduces a comment; at the end of a
;                    parameter line you can also introduce a comment
;                    with "//".
```

[1_MagList]

```
MultiLine=SINGLE
NrOfFixedColumns=1
1= ToolInPlace,  3,   ToolInPlace
2= TC_MPP4_1,    1,   TC_MPP4_1
3= TC_MPP4_2, 1, TC_MPP4_2
4= TC_MPP4_3, 1, TC_MPP4_3
5= TC_MPP4_4,    1,   TC_MPP4_4
6= TC_MPP4_5,    1,   TC_MPP4_5
7= TC_MPP4_6,    1,   TC_MPP4_6
8= TC_MPP4_7,    1,   TC_MPP4_7
9= TC_MPP4_8,    1,   TC_MPP4_8
10= TC_TP2,      11,  TC_TP2
11= TC_TP1,      5,   TC_TP1
12= TC_MPP6,     5,   TC_MPP6
13= TC_TP3,      1,   TC_TP3
14= TC_TP4,      1,   TC_TP4
15= TC_TP5,      1,   TC_TP5
16= TC_TP6,      1,   TC_TP6
17= TC_TP7,      4,   TC_TP7
18= TC_TP8_1,    1,   TC_TP8_1
19= TC_TP8_2,    1,   TC_TP8_2
20= TC_TP8_3,    1,   TC_TP8_3
```

```
21= TC_TP8_4,    1,   TC_TP8_4
22= TC_TP8_5,    1,   TC_TP8_5
23= TC_TP8_6,    1,   TC_TP8_6
24= TC_TP8_7,    1,   TC_TP8_7
25= TC_TP8_8,    1,   TC_TP8_8
26= NoData,      1,   <automatic extend last column>
27= <EndOfList>
```

[2_MagList]

```
MultiLine=MULTI
NrOfFixedColumns=1
1= ToolInPlace,  3,   ToolInPlace
2= TC_TP2,       11,  TC_TP2
3= TC_TP1,       5,   TC_TP1
4= TC_MPP6,      5,   TC_MPP6
5= TC_TP7,       4,   TC_TP7
6= CuttEdgeNo,   1,   CuttEdgeNo
7= TC_DP1,       4,   TC_DP1
8= TC_DP3,       11,  TC_DP3
9= TC_DP6,       11,  TC_DP6
10= TC_TP3,      1,   TC_TP3
11= TC_TP4,      1,   TC_TP4
12= TC_TP5,      1,   TC_TP5
13= TC_TP6,      1,   TC_TP6
14= TC_MPP2,     4,   TC_MPP2
15= TC_ADPT1,    11,  TC_ADAPT1
16= TC_ADPT2,    11,  TC_ADAPT2
17= TC_ADPT3,    11,  TC_ADAPT3
18= TC_ADPT4,    4,   TC_ADAPT4
19= <EndOfList>
```

[3_MagList]

```
NrOfFixedColumns=1
1= ToolInPlace,  3,   ToolInPlace
2= TC_TP2,       11,  TC_TP2
3= TC_TP1,       5,   TC_MPP6
5= TC_TP9,       1,   TC_TP9
6= TC_MOP1,      7,   TC_MOP1
7= TC_MOP2,      7,   TC_MOP2
8= TC_MOP3,      7,   TC_MOP3
9= TC_MOP4,      7,   TC_MOP4
10= TC_MPP3,     1,   TC_MPP3
11= TC_MPP5,     2,   TC_MPP5
12= NoData,      1,   <automatic extend last column>
13= <EndOfList>
```

[1_ToolList]

```
NrOfFixedColumns=1
1= NO,            4,   NO
2= MagNo,         4,   MagNo
3= ToolInPlace,   3,   ToolInPlace
4= TC_TP2,        11,  TC_TP2
```

```
5= TC_TP1,        5,   TC_TP1
6= TC_MPP6,       5,   TC_MPP6
7= TC_TP3,        1,   TC_TP3
8= TC_TP4,        1,   TC_TP4
9= TC_TP5,        1,   TC_TP5
10= TC_TP6,       1,   TC_TP6
11= TC_TP8_1,     1,   TC_TP8_1
12= TC_TP8_2,     1,   TC_TP8_2
13= TC_TP8_3,     1    TC_TP8_3
14= TC_TP8_4,     1,   TC_TP8_4
15= TC_TP8_5,     1,   TC_TP8_5
16= TC_TP8_6,     1,   TC_TP8_6
17= TC_TP8_7,     1,   TC_TP8_7
18= TC_TP8_8,     1,   TC_TP8_8
19= TC_TP7,       4,   TC_TP7
20= NoData,       1,   <automatic extend last column>
21= <EndOfList>
```

[2_ToolList]

```
MultiLine=MULTI
NrOfFixedColumns=1
1= NO,           4,   NO
2= MagNo,        4,   MagNo
3= ToolInPlace,  3,   ToolInPlace
4= TC_TP2,       11,  TC_TP2
5= TC_TP1,       5,   TC_TP1
6= TC_MPP6,      5,   TC_MPP6
7= CuttEdgeNo,   1,   CuttEdgeNo
8= TC_DP1,       4,   TC_DP1
9= TC_DP3,       11,  TC_DP3
10= TC_DP4,      11,  TC_DP4
11= TC_DP5,      11,  TC_DP5
12= TC_DP6,      11,  TC_DP6
13= TC_MPP2,     4,   TC_MPP2
14= <EndOfList>
```

[3_ToolList]

```
NrOfFixedColumns=3
1= NO,           4,   NO
2= MagNo,        4,   MagNo
3= ToolInPlace,  3,   ToolInPlace
4= TC_TP2,       11,  TC_TP2
5= TC_TP1,       5,   TC_TP1
6= TC_MPP6,      5,   TC_MPP6
7= TC_TP3,       1,   TC_TP3
8= TC_TP4,       1,   TC_TP4
9= TC_TP5,       1,   TC_TP5
10= TC_TP6,      1,   TC_TP6
11= TC_DP3@E1,   11,  e1TC_DP3
12= TC_DP3@E2,   11,  e2TC_DP3
13= TC_DP3@E3,   11,  e3TC_DP3
14= TC_TP9,      1,   TC_TP9
15= TC_MOP1,     7,   TC_MOP1
```

```
16= TC_MOP2,     7,    TC_MOP2
17= TC_MOP3,     7,    TC_MOP3
18= TC_MOP4,     7,    TC_MOP4
19= <EndOfList>
```

[1_ActList]

```
MultiLine=SINGLE
NrOfFixedColumns=1
NumLinesPerReq = 11
1= NO,           4,    NO
2= TC_TP2,       11,   TC_TP2
3= TC_TP1,       5,    TC_TP1
4= TC_MPP6,      5,    TC_MPP6
5= CuttEdgeNo,   1,    CuttEdgeNo
6= TC_DPCE,      6,    TC_DPCE
7= MagNo,        4,    MagNo
8= ToolInPlace,  3,    ToolInPlace
9= TC_MPP2,      3,    TC_MPP2
10= TC_MPP5,     4,    TC_MPP5
11= TC_DP1,      11,   TC_DP3
13= TC_DP4,      11,   TC_DP4
14= TC_SCP13,    9,    TC_SCP13
15= TC_SCP14,    9,    TC_SCP14
16= TC_SCP23,    9,    TC_SCP23
17= TC_SCP24,    9,    TC_SCP24
18= TC_ADPT1,    11,   TC_ADAPT1
19= TC_ADPT2,    11,   TC_ADAPT2
20= TC_ADPT3,    11,   TC_ADAPT3
21= TC_ADPT4,    4,    TC_ADAPT4
22= TC_TP8_1,    1,    TC_TP8_1
23= TC_TP8_2,    1,    TC_TP8_2
24= TC_TP8_3,    1,    TC_TP8_3
25= TC_TP8_4,    1,    TC_TP8_4
26= TC_TP8_5,    1,    TC_TP8_5
27= TC_TP8_8,    1,    TC_TP8_8
28= <EndOfList>
```

[2_ActList]

```
MultiLine=MULTI
NrOfFixedColumns=0
NumLinesPerReq = 6
1= NO,           4,    NO
2= TC_TP2,       11,   TC_TP2
3= TC_TP1,       5,    TC_TP1
4= TC_MPP6,      5,    TC_MPP6
5= CuttEdgeNo,   1,    CuttEdgeNo
6= TC_DPCE,      6,    TC_DPCE
7= MagNo,        4,    MagNo
8= ToolInPlace,  3,    ToolInPlace
9= DLNO,         3,    DLNO
10= TC_SCP3,     9,    TC_SCP3
11= TC_SCP4,     9,    TC_SCP4
12= <Empty>
```

```
13= <Empty>
14= TC_DP1,      4,   TC_DP1
15= TC_DP2,      11,  TC_DP2
16= TC_DP3,      11,  TC_DP3
17= TC_DP4,      11,  TC_DP4
18= <Empty>
19= <Empty>
20= TC_ADPT1,    11,  TC_ADAPT1
21= TC_ADPT2,    11,  TC_ADAPT2
22= TC_ADPT3,    11,  TC_ADAPT3
23= TC_ADPT4,    4,   TC_ADAPT4
24= <EndOfList>
```

[3_ActList]

```
NrOfFixedColumns=3
NumLinesPerReq = 11
1= NO,           4,   NO
2= MagNo,        4,   MagNo
3= ToolInPlace,  3,   ToolInPlace
4= TC_TP2,       11,  TC_TP2
5= TC_TP1,       5,   TC_TP1
6= TC_DPCE,      6,   TC_DPCE
7= TC_TP3,       1,   TC_TP3
8= TC_TP4,       1,   TC_TP4
9= TC_TP5,       1,   TC_TP5
10= TC_TP6,      1,   TC_TP6
11= TC_TP9,      1,   TC_TP9
12= TC_MOP1,     7,   TC_MOP1
13= TC_MOP2,     7,   TC_MOP2
14= TC_MOP3,     7,   TC_MOP3
15= TC_MOP4,     7,   TC_MOP4
16= TC_MOP5,     7,   TC_MOP5
17= TC_MOP6,     7,   TC_MOP6
18= TC_MOP11,    7,   TC_MOP11
19= TC_MOP13,    7,   TC_MOP13
20= TC_MOP15,    7,   TC_MOP15
21= <EndOfList>
```

[ListColumnHeaderTrafoTextReplace]

```
; Example:
; TC_DP3 = TTC_DP3
;                    defines the replacement text for the transformed
;                    display of lists for the cutting edge parameters
;                    and the additive offset parameters in the column
;                    headers.
;                    The substitute code for the transformed
;                    presentation is searched in the code “TC_DP3”.

;                    If an entry is missing in this section, then the
;                    column header of the appropriate column “missed
;                    trafo text” (“no transformation text available”)
;                    is displayed.
```

```
TC_DP3       = TTC_DP3
e1TC_DP3     = Te1TC_DP3
e2TC_DP3     = Te2TC_DP3
e3TC_DP3     = Te3TC_DP3
TC_DP4       = TTC_DP4
TC_DP5       = TTC_DP5
TC_DP6       = TTC_DP6
TC_DP7       = TTC_DP7
TC_DP8       = TTC_DP8
TC_DP9       = TTC_DP9
TC_DP10      = TTC_DP10
TC_DP11      = TTC_DP11
TC_DP12      = TTC_DP12
TC_DP13      = TTC_DP13
TC_DP14      = TTC_DP14
TC_DP15      = TTC_DP15
TC_DP16      = TTC_DP16
TC_DP17      = TTC_DP17
TC_DP18      = TTC_DP18
TC_DP19      = TTC_DP19
TC_DP20      = TTC_DP20
TC_DP21      = TTC_DP21
TC_DP22      = TTC_DP22
TC_DP23      = TTC_DP23
TC_DP24      = TTC_DP24
TC_DP25      = TTC_DP25
TC_DPH       = TTC_DPH
TC_DPV       = TTC_DPV
TC_DPV3      = TTC_DPV3
TC_DPV4      = TTC_DPV4
TC_DPV5      = TTC_DPV5
TC_SCP3      = TTC_SCP3
TC_SCP4      = TTC_SCP4
TC_SCP5      = TTC_SCP5
TC_SCP6      = TTC_SCP6
TC_SCP7      = TTC_SCP7
TC_SCP8      = TTC_SCP8
TC_SCP9      = TTC_SCP9
TC_SCP10     = TTC_SCP10
TC_SCP11     = TTC_SCP11
TC_SCP13     = TTC_SCP13
TC_SCP14     = TTC_SCP14
TC_SCP15     = TTC_SCP15
TC_SCP23     = TTC_SCP23
TC_SCP24     = TTC_SCP24
TC_SCP25     = TTC_SCP25
TC_SCP33     = TTC_SCP33
TC_SCP34     = TTC_SCP34
TC_SCP35     = TTC_SCP35
e1TC_SCP13   = Te1TC_SCP13
e1TC_SCP14   = Te1TC_SCP14
e1TC_SCP15   = Te1TC_SCP15
e1TC_SCP23   = Te1TC_SCP23
```

```
e1TC_SCP24    = Te1TC_SCP24
e1TC_SCP25    = Te1TC_SCP25
e1TC_SCP33    = Te1TC_SCP33
e1TC_SCP34    = Te1TC_SCP34
e1TC_SCP35    = Te1TC_SCP35
e2TC_SCP13    = Te2TC_SCP13
e2TC_SCP14    = Te2TC_SCP14
e2TC_SCP15    = Te2TC_SCP15
e2TC_SCP23    = Te2TC_SCP23
e2TC_SCP24    = Te2TC_SCP24
e2TC_SCP25    = Te2TC_SCP25
e2TC_SCP33    = Te2TC_SCP33
e2TC_SCP34    = Te2TC_SCP34
e2TC_SCP35    = Te2TC_SCP35
e3TC_SCP13    = Te3TC_SCP13
e3TC_SCP14    = Te3TC_SCP14
e3TC_SCP15    = Te3TC_SCP15
e3TC_SCP23    = Te3TC_SCP23
e3TC_SCP24    = Te3TC_SCP24
e3TC_SCP25    = Te3TC_SCP25
e3TC_SCP33    = Te3TC_SCP33
e3TC_SCP34    = Te3TC_SCP34
e3TC_SCP35    = Te3TC_SCP35
TC_ECP3       = TTC_ECP3
TC_ECP4       = TTC_ECP4
TC_ECP5       = TTC_ECP5
TC_ECP6       = TTC_ECP6
TC_ECP7       = TTC_ECP7
TC_ECP8       = TTC_ECP8
TC_ECP9       = TTC_ECP9
TC_ECP10      = TTC_ECP10
TC_ECP11      = TTC_ECP11
TC_ECP13      = TTC_ECP13
TC_ECP14      = TTC_ECP14
TC_ECP15      = TTC_ECP15
TC_ECP23      = TTC_ECP23
TC_ECP24      = TTC_ECP24
TC_ECP25      = TTC_ECP25
TC_ECP33       TTC_ECP33
TC_ECP34      = TTC_ECP34
TC_ECP35      = TTC_ECP35
e1TC_ECP13    = Te1TC_ECP13
e1TC_ECP14    = Te1TC_ECP14
e1TC_ECP15    = Te1TC_ECP15
e1TC_ECP23    = Te1TC_ECP23
e1TC_ECP24    = Te1TC_ECP24
e1TC_ECP25    = Te1TC_ECP25
e1TC_ECP33    = Te1TC_ECP33
e1TC_ECP34    = Te1TC_ECP34
e1TC_ECP35    = Te1TC_ECP35
e2TC_ECP13    = Te2TC_ECP13
e2TC_ECP14    = Te2TC_ECP14
e2TC_ECP15    = Te2TC_ECP15
```

```
e2TC_ECP23    = Te2TC_ECP23
e2TC_ECP24    = Te2TC_ECP24
e2TC_ECP25    = Te2TC_ECP25
e2TC_ECP33    = Te2TC_ECP33
e2TC_ECP34    = Te2TC_ECP34
e2TC_ECP35    = Te2TC_ECP35
e3TC_ECP13    = Te3TC_ECP13
e3TC_ECP14    = Te3TC_ECP14
e3TC_ECP15    = Te3TC_ECP15
e3TC_ECP23    = Te3TC_ECP23
e3TC_ECP24    = Te3TC_ECP24
e3TC_ECP25    = Te3TC_ECP25
e3TC_ECP33    = Te3TC_ECP33
e3TC_ECP34    = Te3TC_ECP34
e3TC_ECP35    = Te3TC_ECP35
```

[BatchTools]

```
; Control of the job functions for the tools:
; load, unload or reactivate a number of tools
; Note: The tool filters only function if bit 4 (from 0 to ...) is
; set in ToolManagementMask.
; Max. 6 filters can be specified.
; The following can be specified for each filter:
; Softkey text, list header, search criteria, selection of the type
; of results list and additional data
; The file ...user\paramini.out contains error messages for the
; errors that were encountered when reading in the parameters.
; Search criteria:
; Permissible values in "_FindCondition":
; A maximum of 8 entries are permissible, separated by ",". They are
; ANDed.
; No data may occur more than once in the part conditions. Each part
; condition consists of three parts:
; 1. Datum for which the condition applies
; 2. Condition
  3. Comparison value
```

```
; The following data can be a filter criterion:
; Tool data:
; TC-TP1       Duplo number
; TC-TP2       Tool identifier
; TC-TP3       Tool size in half locations left
; TC-TP4       Tool size in half locations right
; TC-TP5       Tool size in half locations top
; TC-TP6       Tool size in half locations bottom
; TC-TP7       Tool location type
; TC-TP8       Tool status
; TC-TP9       Monitoring type
; TC-TP10      Replacement tool search
; TC-TP11      Tool information/replacement tool sequence
; A_TOOLMN     Magazine number
; A_TOOLMLN    Magazine location number
; P_TOOLND     Number of cutting edges
; Tool OEM data:
; "TC_TPC1" to "TC_TPC10",
; Tool OEM data must be activated on the NC and the numbers must be
; permissible on the NC.
; Tool cutting edge parameters:
; "TC_DP1" to "TC_DP25", "TC_DPH", "TC_DPV", "TC_DPV3", "TC_DPV4",
; "TC_DPV5"
; (the NCK setting applies instead of "25")
; Tool cutting edge OEM data:
; "TC_DPC1" to "TC_DP10"
; Tool cutting edge OEM data must be activated on the NC and the num-
; bers must be permissible on the NC.
; Tool cutting edge monitoring parameters:
; TC_MOP1      Prewarning limit for tool life
; TC_MOP2      Actual value for tool life
; TC_MOP3      Prewarning limit for workpiece count
; TC_MOP4      Actual value for workpiece count
; TC_MOP5      Prewarning limit for wear
; TC_MOP6      Remaining wear
; TC_MOP11     Setpoint for tool life
; TC_MOP13     Setpoint for tool workpiece count
; TC_MOP15     Setpoint for wear
; Tool cutting edge monitoring OEM data:
; "TC_MOPC1" to "TC_MOPC10"
; Tool cutting edge monitoring OEM data must be activated on the NC
; and the numbers must be permissible on the NC.
; If NckVersion >= 430000: User data
; Tool user data:
; "TC_TPCS1" to "TC_TPCS10"
; Tool user data must be activated on the NC and the numbers must be
; permissible on the NC.
; Tool cutting edge user data::
; "TC_DPCS1" to "TC_DPCS10"
; Tool cutting edge user data must be activated on the NC and the
; numbers must be permissible on the NC.
; Cutting edge monitoring user data:
; "TC_MOPCS1" to "TC_MOPCS10"
; Cutting edge monitoring user data must be activated on the NC and
; the numbers must be permissible on the NC.
```

```
; Condition:
; “==”          equal to
; “<”           smaller than
; “>”           greater than
; “<=”          smaller than or equal to
;               greater than or equal to
; &&            Bit-wise AND, only permissible for operands of type
;               WORD and DOUBLEWORD
; “==” is the only relational operator allowed for string operands
; Comparison value:
; String for TC_TP2 (tool data, tool identifier), max. 32 characters,
; no blanks before or after
;  0 ... 65535 for the other TC_TP data
; Double for all other data
; Max. one column with additional data can appear for each filter
; _FindResultAddColumnBtss:
; Additional data, OPI item acc. to OPI documentation
; (mmc2\btss_gr.hlp).
; Example 1: “/Tool/User/data[u#TOA#,c2,#TNO#](|“!d%.#RES#lf”)” tool
; OEM parameter 2, floating point representation, standard number of
; places after the decimal point
; Example 2: “/Tool/User/data[u#TOA#,c3,#TNO#](|“!l%ld”)” tool OEM
; parameter 3, integer representation
; Example 3: “/Tool/MagazineDescription/userData[u#TOA#,c#MAG#,1](|)”
; magazine OEM parameter 1
; The following placeholders are permissible: #TOA#, #TNO#, #MAG#,
; #RES#.
; #TOA#         TOA number (of the current channel)
; #TNO#         Internal T number (of the tool found)
; #MAG#         Magazine number (of the found tool)
; #RES#         Standard value for the number of places after the
;               decimal point
; Placeholders are substituted by the data for the current tool or by
; general settings.
; Max. 1 OPI item is permitted.
; “(|)” is entered in front of the result data to generate the data
; separation character “|”.
; OPI multiple variable accesses are generated internally from the
; OPI item.
; The OPI item must enclosed by “ ” especially when formatting infor-
; mation is contained in “ ”.
; The user setting the parameters is responsible for the correct syn-
; tax. The syntax is not checked by the OPI.
; General settings for all filters:
; This entry applies for HMI_ADV prior to software Version 6.3.
; With Version 6.3 and higher it is ignored.
; Colors for the Results list:
; A hex value consisting of 8 characters is assigned to each color.
; The hex value has the following syntax:
; SSBBGGRR where SS=System, BB=Blue, GG=Green, RR=Red
; The colors have to be specified for the following list elements:
; Non-selected text
; Non-selected background
; Cursor-selected text
```

```
; Cursor-selected background
; Job-selection and cursor-selected text
; Job-selection and cursor-selected background
; Job-selection and cursor-selected text
; Job-selection and cursor-selected background
; ;WinTxt, WinBa, HighLTxt,HighLiBa,TiBaTxt, green, ,TiBaTxt, blue-
; green
; ResultColors=80000008,80000005,8000000E,8000000D,80000009,0000FF00,
; 80000009,00FF8000
; ;experimental, LiteBlue for batch selected
; ;WinTxt, WinBa, HighLTxt,HighLiBa,WinTxt ,LiteBlue,HighLTxt,High-
; LiBa
; ResultColors=80000008,80000005,8000000E,8000000D,80000008,00FFFF00,
; 8000000E,8000000D
; ;experimental, LiteGreen for batch selected
; ;WinTxt, WinBa, HighLTxt,HighLiBa,WinTxt ,Litegree,HighLTxt,High-
; LiBa
; ResultColors=80000008,80000005,8000000E,8000000D,80000008,000FF000,
; 8000000E,8000000D
; WinTxt, WinBa, HighLTxt,HighLiBa,HighLTxt,HighLiBa,HighLTxt,High-
; LiBa
ResultColors=80000008,80000005,8000000E,8000000D,8000000E,8000000D,
8000000E,8000000D

; The user can replace the names of the bitmaps or the bitmaps them-
; selves with custom bitmaps. The custom bitmaps are stored in the
; “user” directory.
BatchFilterElBUnTUnBitmap  = pbfbutu.bmp
BatchFilterElBUnTSeBitmap  = pbfbuts.bmp
BatchFilterElBSeTUnBitmap  = pbfbstu.bmp
BatchFilterElBSeTSeBitmap  = pbfbsts.bmp
BatchRunElWaitingBitmap    = pbbwait.bmp
BatchRunElInWorkBitmap     = pbbwork.bmp
BatchRunElOKBitmap         = pbbok.bmp
BatchRunElErrorBitmap      = pbberr.bmp

; Width of a typical character
CharToGetColWidthPerCharacter= CharToGetColWidth

; language-dependent, see ...\language\patm_xx.ini
; Selection of the tool status bits which are displayed in the result
; list:
ResultToolStatusColumnsEnable=    1111100100110000

; Bits 1 to 16, bit 1 is the least-significant bit in the tool status
  and is positioned to the left in this character sequence
; Text in the header for the tool status column:
; If ResultToolStatusColumnsHeaderText and ResultToolStatusColumn-
; sListText equal “<Empty>”, the locale-specific values are contained
; in the batch list for pa_xx.dll.
ResultToolStatusColumnsHeaderText=<Empty> ;language-dependent/locale-
specific
; ResultToolStatusColumnsHeaderText=ToolStatusColHeaderText ;lan-
; guage-dependent
```

```
; Text in the data of the tool status column:
ResultToolStatusColumnsListText= <Empty> ;language-dependent/locale-
specific
; ResultToolStatusColumnsListText= ToolStatusColListText ;language-
; dependent
; Column width for tool identifier
ResultDisplayedNumberOfToolnameCharacters=18
TimeMSecBetweenBatchOrders=1000
; Definitions of individual filters:
1_FindSoftkeyText = F1SK
; Locale-specific; prewarning or disabled
1_FindResultHeadlineText = R1HL
; Locale-specific; prewarning limit reached or disabled
1_FindCondition = TC_TP8 && 20
; Prewarning bit set (bit 5 of bit 1 to 16 (2 to the power of
; (5-1)=16) + Disabled bit set (bit 3 (2 to the power of (3-1)=4)
1_FindResultAddColumnBtss = <empty>
1_FindResultAddColumnText = <empty> ; or R1AddCol ; locale-specific
1_FindResultAddColumnDisplayedNumberOfCharacters=0
1_FindLimitedToCurMagazine=true
; "True", "False"(default setting) ; limited to current magazine, if
; called via magazine list.
1_ResultListType =0
; 0 = Standardliste (default setting), 1 = loading list
1_ReactivatePositioningMode = 2
; Positioning during Reactivate
; 0: Do not position, 1: Ask the operator whether to position, 2:
; Always positioning (default setting)

2_FindSoftkeyText = F2SK ; "disabled"
2_FindResultHeadlineText = R2HL ; "Tools disabled"
2_FindCondition = TC_TP8 && 4
; Disabled=bit 3 (2 to the power of (3-1)=4)
2_FindResultAddColumnBtss = <empty>
2_FindResultAddColumnText = <empty>  ; or R2AddCol
2_FindResultAddColumnDisplayedNumberOfCharacters=0
2_FindLimitedToCurMagazine=False
; "True", "False"(default setting) ; limited to current magazine, if
; called via magazine list.
2_ResultListType =0
; 0 = Standardliste (default setting), 1 = loading list
2_ReactivatePositioningMode = 0
; 0: Do not position, 1: Ask the operator whether to position, 2:
; Always positioning (default setting)
```

```
3_FindSoftkeyText = F3SK ; “Load all”
3_FindResultHeadlineText = R3HL
; Unloading list for all loaded tools
3_FindCondition = A_TOOLMN > 0
; Magazine number of tool greater than 0
3_FindResultAddColumnBtss = <empty>
3_FindResultAddColumnText = <empty>  ; or R3AddCol
3_FindResultAddColumnDisplayedNumberOfCharacters=0
3_FindLimitedToCurMagazine=False
; “True”, “False”(default setting) ; limited to current magazine, if
; called via magazine list.
3_ResultListType =0
; 0 = Standardliste (default setting), 1 = loading list
3_ReactivatePositioningMode = 1
; Positioning during Reactivate
; 0: Do not position, 1: Ask the operator whether to position, 2:
; Always positioning (default setting)

4_FindSoftkeyText = F4SK ; “Unload all”
4_FindResultHeadlineText = R4HL
; Loading list for all unloaded tools
4_FindCondition = A_TOOLMN == 0
; Magazine number of tool equal to 0
4_FindResultAddColumnBtss = <empty>
4_FindResultAddColumnText = <empty>  ; or R4AddCol
4_FindResultAddColumnDisplayedNumberOfCharacters=0
4_FindLimitedToCurMagazine=False
; “True”, “False”(default setting) ; limited to current magazine, if
; called via magazine list.
4_ResultListType =0
; 0 = Standardliste (default setting), 1 = loading list
4_ReactivatePositioningMode = 1
; Positioning during Reactivate
; 0: Do not position, 1: Ask the operator whether to position, 2:
; Always positioning (default setting)

5_FindSoftkeyText = F5SK ; “Load identifier”
5_FindResultHeadlineText = R5HL
; “Load list for all tools with load identifier”
5_FindCondition = TC_TP8 && 2048
; (LoadIdentifier=bit12 (2 to the power of (12-1)=2048)
5_FindResultAddColumnBtss = <empty>
5_FindResultAddColumnText = <empty>  ; or R5AddCol
5_FindResultAddColumnDisplayedNumberOfCharacters=0
5_FindLimitedToCurMagazine=False
; “True”, “False”(default setting) ; limited to current magazine, if
; called via magazine list.
5_ResultListType =1
; 0 = Standardliste (default setting), 1 = loading list
```

```
6_FindSoftkeyText = F6SK ; "Unload identifier"
6_FindResultHeadlineText =  R6HL
; "Unload list for all tools with unload identifier"
6_FindCondition = TC_TP8 && 1024
; (UnloadIdentifier=bit11 (2 to the power of (11-1)=1024)
6_FindResultAddColumnBtss = <empty>
6_FindResultAddColumnText = <empty>  ; or R6AddCol
6_FindResultAddColumnDisplayedNumberOfCharacters=0
6_FindLimitedToCurMagazine=False
; "True", "False"(default setting) ; limited to current magazine, if
; called via magazine list.
6_ResultListType =0
; 0 = Standardliste (default setting), 1 = loading list
```

[ShortcutSoftkeysForMagSelect]

```
; Definition of shortcut softkeys for up to 5 favorite magazines per
; TOA, evaluated if section "[GeneralSettingsForMagAndToolList]"
; entry "MagListMagSelectSoftkey=SelectMag" is set.
; This helps to prevent users from excessive use of "Magazine +" and
; "Magazine -" softkeys, if a lot of magazines are available.
; You can define up to 5 shortcut softkeys for favorite magazines. It
; is possible to use a shortcut softkey for different magazines, if
; the magazines are in different TOAs.
; Syntax:         "magIdent = ShortKeyNummer, AutoReturn"
; Examples:       revolver15=3,NoAuto<<
;                 chain50   =1,Auto<<
; Meaning:        "magIdent": magazine ident like in $TC_MAP2 or in
;                 magazine configuration in application maintenance
;                 tool management.
;                 "ShortKeyNumber": Number of shortcut, value 1 to 5
;                 "AutoReturn": stay in magazine selection state or
;                 return automatically to magazine list state Values
;                 "NoAuto<<" and "Auto<<".
;                 magazine "revolver15" ($TC_MAP2) can be displayed
;                 by shortcut 3, you must use "<<" softkey explicitly
;                 to leave magazine select state.
;                 magazine "chain50" can be displayed by shortcut 1
;                 and there is an automatic return to magazine list
;                 state after pressing this shortcut softkey.
; To specify softkey text, use section "[ShortcutSoftKeysForMagSe-
; lect]" in language dependent ini files patm_??.ini.
;chain10    = 1, Auto<<
;turret20 = 2, Auto<<
;turret10 = 3, Auto<<
;chain20    = 5, NoAuto<<
;turret15 = 4, NoAuto<<
```

Configuring the tool management displays in the paramtm.ini file

The tool management operator interface can be customized with the file paramtm.ini.

The following custom settings can be made:

- Mofify the structure and layout of the lists
- Apply specific default values
- Protect or deactivate functions via access rights.

All the functions and possibilities of the tool management are listed in the paramtm.txt file on HMI Advanced.

At installation and start-up, the operator can decide which functions are required for a specific machine. Vaues and to some extent also functions can be preset to allow for convenient, user-friendly operation.

Examples for assigning parameters for access rights

Example 1

- The tool data is to be automatically deleted when unloaded (magazine list only).
- The tool list function is not used.
- The function tool catalog and cabinet is not used.

The parameters can be assigned as follows:

```
. . .
[TMMODES]
. . .
DELETE_TOOL_ON_UNLOAD=1  The tool data is automatically deleted at unloading
....
[ACESSLEVEL]
. . .
SKTLLIST=2               The tool list is only activated by the manufacturer
                         code, i.e. is disabled during normal operation.
SLTOOLCAB=2              Activation of the tool catalog and cabinet
                         SKTOOLCAT=2 is only via manufacturer code, there-
                         fore they are disabled during normal operation
```
....

Example 2

- The tool data is not deleted at unloading, but remain in the tool list (in NCK). The data can be used for loading tools.
- The function tool catalog and tool cabinet is not used.

The parameters can be assigned as follows:

```
. . .
[TMMODES]
. . .
DELETE_TOOL_ON_UNLOAD=0   The tool data is not automatically deleted at un-
                          loading
....
[ACCESSLEVEL]
. . .
SKTLLIST=7                The tool list can always be called.
SLTOOLCAB=2               Activation of the tool catalog and cabinet
                          SKTOOLCAT=2 is only via manufacturer code, there-
                          fore they are disabled for the user.
....
```

Example 3

The tool data is to be automatically deleted at unload in the magazine list.

The function tool catalog and tool cabinet is used.

The parameters can be assigned as follows:

```
. . .
[TMMODES]
. . .
DELETE_TOOL_ON_UNLOAD=1   The tool data is deleted at unloading
....
[ACESSLEVEL]
. . .
SKTLLIST=2                The tool list is only active via manufacturer pass-
                          word.
SLTOOLCAB=7               The tool catalog and tool cabinet
SKTOOLCAT=7               can be called (are not locked)
. . .
```

If access rights have been assigned for functions and the protection level is “less” than that assigned, then the softkey is not displayed in the operator interface and the function cannot be called.

This applies to all functions. If for example, the “Tool cabinet” function is barred from operation, then its softkeys are not displayed.

Parameter settings for bitmaps in the lists

The display of the active tool, the programmed tool and the current location in the magazine list can be freely parameterized in SW 5.2 and later, i.e. bitmaps can be inserted in the parameterizable columns of individual lists. These bitmaps can be customized to suit user requirements and are created in programs such as Paintbrush. This view is activated in paramtm.ini.

The bitmaps for the current tools are shown in red and those for the programmed tools are shown in green. The standard bitmaps described below reside in the directory “mmc2” (see /IAM/ Installation HMI Advanced, IM 4).

Standard bitmaps

Bitmap	Properties
Two arrow heads pointing to the right	TNo. <> 0; DNo./cutting edge no.<> 0 DLNo. <> 0
Arrow right	TNo. <> 0; DNo./cutting edge no.<> 0 DLNo. = 0
Arrow head pointing to left	TNo. <> 0; DNo./cutting edge no. = 0 DLNo. = 0
Dark green parallelepiped	Current location

User-defined bitmaps can be stored in the “user” directory. They can be displayed instead of the standard bitmaps in the lists.

Handling of lists

The columns of the lists in which the bitmaps are to be entered can be set for each list view. The width of the bitmaps is set in characters for the entire highlighted areas. The width of the column is automatically increased by the value set.

Bitmaps overwrite mutually when displayed in the same column and line. The highlighting at the top represents the current tool, the highlighting below represents the programmed tool and the highlighting at the bottom represents the current location. Hidden bitmaps are not displayed.

Notice

In multi-line magazine and tool lists, the marking is entered in the cutting edge line when the current/programmed DNo./cutting edge no.<> 0. The same applies in working offset lists for DLNo. <> 0 for the DL rows. Since only cutting edges can be displayed in the views of the working offset lists, the highlighting only appears if the current/programmed DNo./cutting edge no. <> 0.

The current magazine location is only highlighted in the magazine list views. Highlighting only appears in the normal magazine display and not in the buffer display.

Setting the bitmap parameters

By default, the bitmaps are not entered in paramtm.ini and are not displayed. If the bitmaps are to be displayed in the lists, you will have to make some changes to the parameter file. One entry is required for each bitmap.

Entries in paramtm.ini:

```
[GeneralSettingForMagAndToolList]

; Width of the bitmap display
; Unit: number of characters
WidthOfActBitmapsInCharacters=5

; Name of the bitmap for the current tool /DNo./DL,
; with D<>0 and DL<>0
ActToolBitmap=paat.bmp

; Name of the bitmap for the current tool/DNo.,
; if the current cutting edge D=0.
ActToolZeroDBitmap=paatd0.bmp

; Name of the bitmap for the current tool /DNo./DL,
; if the current DL=0.
ActToolZeroDLBitmap=paatdl0.bmp

; Name of the bitmap for the programmed tool /DNo./DL,
; with D<>0 and DL<>0
ProgToolBitmap = papt.bmp

; Name of the bitmap for the programmed tool/DNo.,
; if the current cutting edge D=0.
ProgToolZeroDBitmap = paptd0.bmp

; Name of the bitmap for the programmed tool /DNo./DL,
; if the current DL=0.
ProgToolZeroDLBitmap = paptdl0.bmp

; File name of the bitmap for the current magazine location
ActPlaceBitmap = paap.bmp

[1_MagList]
; Columns in which highlighting (Bitmaps) is to be displayed
```

```
ShowActToolCol=1
ShowProgToolCol=1
ShowActPlaceCol=1
```

Instructions for configuring the paramtm.ini file

Input of the softkey texts for the lists

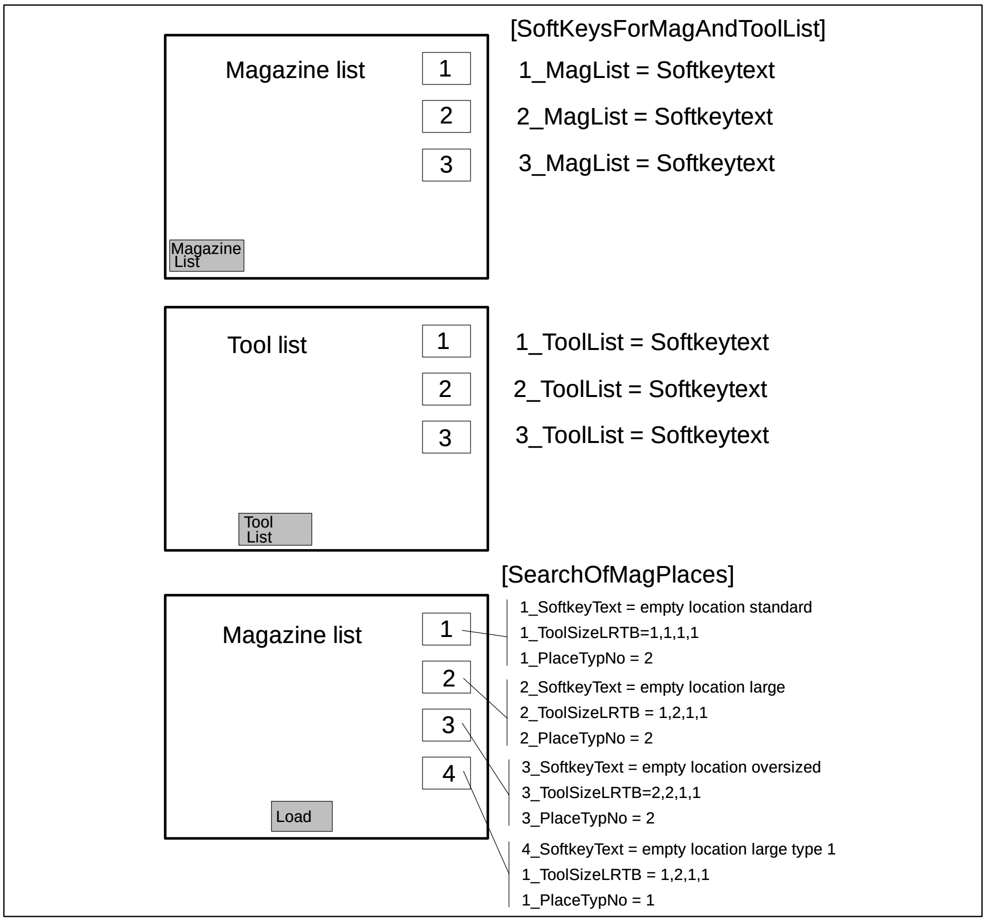


Bild 4-17 Softkeys

The displays stored behind softkeys 1 to 3 in the magazine and tool lists are defined in the file paramtm.ini. As the initial setting when tool management is selected, the displays appear that have been configured for [1_MagList] and [1_ToolList].

Displaying the displays

Hidden fields can be made visible by scrolling with the cursor keys.

The serial number defined by the input sequence during start-up is displayed in the location type box rather than the name of the location. The screen that is displayed under the 1st vertical softkey in the magazine list is specified after vocabulary word [1_MagList] in file paramtm.ini.

User data

The parameter name and the units can be defined for the displays of the tool and cutting-edge data. How many parameters are displayed depends on the MD and the number of defined parameters.

[ToolParams]	Tool user data
[ToolEdgeParams]	Cutting edge user data

Special characters

Special characters such as ü, ä, ö, ß are entered in ANSI code in order for them to be displayed in the screens.

Optional selection of magazines

Up to now, the softkey “Next magazine” could only control the display of the individual magazine lists.
If many magazines are present and there are important processes taking place in the magazines with high magazine numbers, this places a burden on the operator.

Therefore, the following new option was integrated:

Via an entry in the INI file the “Next magazine” softkey can be replaced with the softkey “Magazine selection”.

Next magazine ⇨ **Magazine selection**

Eight vertical softkeys for faster magazine navigation:
(Softkey 3 to 7 can only be used if the respective parameter was set in the INI file.)

Magazine +

Magazine –

Magazine shortcut 1

Magazine shortcut 2

Magazine shortcut 3

Magazine shortcut 4

Magazine shortcut 5

<<

The magazine list is displayed in this status.

Using the vertical softkeys 1 “Magazine +” and 2 “Magazine –” you can switch to the magazine with the next highest or next lowest magazine number within the TOA of the current channel of the operator panel. (When you are positioned on the last magazine and press “+” you jump to the first magazine; from the first to the last again with “–”.)

Using the five vertical softkeys 3 to 7, you can quickly jump to a specific magazine within the TOA of the current channel of the operator panel. The assignment to “Magazine-Ident” and the softkey text must be parameterized in the INI file. When you select a magazine via the vertical softkeys, the magazine list switches immediately to the new magazine.
Press the vertical softkey 8 “<<” to return to the standard magazine list state with the corresponding softkey assignment.

With the five softkey for rapid magazine selection, you can set an option in the INI file to return to the standard magazine list state automatically.
In this case, it is advisable to append the character sequence “”<<” to the magazine name in the softkey text.

The rapid selection keys support multiple assignment for use in different TOAs and for systems with N:M assignment between HMI_ADVs and NCUs.

The responsibility of assigning only magazines in different TOAs or different NCUs to the same softkey lies with the person setting the parameters.

Entries in paramtm.ini

```
[GeneralSettingsForMagAndToolList]
; In the magazine list forms change softkey “Next Mag” to softkey
; “Magazine Selection”
; to activate vertical softkeys in an additional state for magazine
; selection.
; This helps to prevent users from excessive use of “next mag” soft-
; key if a lot of magazins ara available.
; You can define shortcut softkeys for up to 5 favorite magazines for
; each TOA
; using section [ShortcutSoftKeysForMagSelect] in paramtm.ini and
; patm_??.ini.
MagListMagSelectSoftkey=NextMag   ;default
;MagListMagSelectSoftkey=SelectMag

[ShortcutSoftKeysForMagSelect]
; Definition of shortcut softkeys for up to 5 favorite magazines per
; TOA, evaluated if section “[GeneralSettingsForMagAndToolList]”
; entry “MagListMagSelectSoftkey=SelectMag” is set.
; This helps to prevent users from excessive use of “Magazine +” and
; “Magazine -” softkeys, if a lot of magazines are available.
; You can define up to 5 shortcut softkeys for favorite magazines.
; It is possible to use a shortcut softkey for different magazines,
; if the magazines are in different TOAs.
; Syntax: “magIdent = ShortKeyNumber, AutoReturn”
; Examples: revolver15=3,NoAuto<<
; chain50 =1,Auto<<
; Explanation: “magIdent”: magazine ident like in $TC_MAP2 or in mag-
; azine configuration in application maintenance tool management.
; “ShortKeyNumber”: Number of shortcut, value 1 to 5
; “AutoReturn”: stay in magazine selection state or return automati-
; cally to magazine list state
; Values “NoAuto<<” and “Auto<<”.
; magazine “revolver15” ($TC_MAP2) can be displayed by shortcut 3,
; you must use “<<” softkey explicitly to leave magazine select
; state.
; magazine “chain50” can be displayed by shortcut 1 and you return
; automatically to magazine list state after pressing this shortcut
; softkey.
; To specify the softkey text, use section
; “[ShortcutSoftKeysForMagSelect]” in language dependent ini files
; patm_??.ini.
; chain10 = 1, Auto<<
; turret20 = 2, Auto<<
> ;turret10 = 3, Auto<<
> ;chain20 = 5, NoAuto<<
> ;turret15 = 4, NoAuto<<
>
language\patm_*.ini:
[ShortcutSoftKeysForMagSelect]
```

```
; Softkey text of magazine selection shortcut softkeys.
; Syntax: magIdent=ShortcutSoftkeyText
; Explanation: “magIdent”: magazine ident like in $TC_MAP2 or in mag-
; azine configuration in application
; maintenance tool management.
; “ShortcutSoftkeyText”: Softkey text, use double blank to indicate
; wordwrap.
; turret10 = “1-Turret10 <<” // Softkey text
; turret20 = “2-Turret20 <<” // Softkey text
; chain10 = “3-Chain10 <<” // Softkey text
; Turret15 = “4-Turret15” // Softkey text
; Chain20 = “5-Chain20” // Softkey text
```

Display location status Adjacent locations

In the list displays (magazine list, tool list, working offset list) of the HMI Advanced tool management now also bits 8 to 16 of the magazine location status can be displayed, including the 4 bit “left, right, top, bottom half location reserved”.

The NC uses this data when “Adjacent location management” is activated.

The data largely correspond to the NC variable $TC_MPP4.

Until now, bits 1 to 8 of the magazine location status could be displayed in the lists; this expansion now makes it possible to display bits 8 to 16 as well.

The magazine location status bits 8 to 16 can be displayed using the HMI_ADV software; they cannot, however, be changed.

The HMI_ADV software contains a sample parameter assignment (paramtm.ini, paramtm.txt) where display of the additional magazine location status bits is prepared but not activated.

Parameter assignment

The parameter assignment of the status bit display as column in the individual list views for list displays 1 to 3 (magazine list 1 to 3, tool list 1 bis 3, working offset list 1 to 3) is accordingly expanded to include the nine status bits.

As was the case previously, the individual language-dependent letters for display in the HMI_ADV tool management list displays can also be parameterized in the INI file as an exception, e.g. if the machine operator wants to use different letters or there is no modified language DLL for the locale (see Section 4.4.3).

Restrictions

The NC only uses the data “left, right, top, bottom half location occupied/reserved” if “Adjacent location management” is activated. If “Adjacent location management” is not activated in the NC, display can still be activated for the associated magazine location status bits in the HMI_ADV tool management list-displays but the displayed values will always be “Bit not set”.

Settings in the INI files

paramtm.ini / paramtm.txt:

```
[GeneralSettingsForMagAndToolList]
MagPlaceState_Lang_12345678=<Empty> ; use language-DLL
; MagPlaceState_Lang_12345678=12345678_MagPlaceState_Lang ; use
; patm_*.ini
; The new magazine location status bits can be displayed in each list
; screen. The parameters are set in the sections:
“[1_MagList]”, “[2_MagList]”, “[3_MagList]”
“[1_ToolList]”, “[2_ToolList]”, “[3_ToolList]”
“[1_ActList]”, “[2_ActList]”, “[3_ActList]”
; Entries are for example:
12=TC_MPP4_9,  1, TC_MPP4_9   ;PlaceStatus Left,
                              ;Reserved in left half location
13=TC_MPP4_10, 1, TC_MPP4_10  ;PlaceStatus Right,
                              ;Reserved in right half location
14=TC_MPP4_11, 1, TC_MPP4_11  ;PlaceStatus Top,
                              ;Reserved in upper half location
15=TC_MPP4_12, 1, TC_MPP4_12  ;PlaceStatus Bottom,
                              ;Reserved in lower half location
16=TC_MPP4_13, 1, TC_MPP4_13  ;PlaceStatus Bit 13 of 1 to 16
17=TC_MPP4_14, 1, TC_MPP4_14  ;PlaceStatus Bit 14 of 1 to 16
18=TC_MPP4_15, 1, TC_MPP4_15  ;PlaceStatus Bit 15 of 1 to 16
19=TC_MPP4_16, 1, TC_MPP4_16  ;PlaceStatus Bit 16 of 1 to 16
```

patm_gr.ini:

```
[GeneralSettingsForMagAndToolList]
12345678_MagPlaceState_Lang=“123456789ABCDEFG”      ; // 16 exact
```

```
[ListColumnHeaderText]
TC_MPP4_9    = “P”      ;PlaceStatus Left,
                        ;Reserved in the left half loc. // 1
TC_MPP4_10   = “P”      ;PlaceStatus Right,
                        ;Reserved in the right half loc. // 1
TC_MPP4_11   = “P”      ;PlaceStatus Top,
                        ;Reserved in the top half loc. // 1
TC_MPP4_12   = “P”      ;PlaceStatus Bottom,
                        ;Reserved in the bottom half loc. // 1
TC_MPP4_13   = “P”      ;PlaceStatus undefined,
                        ;(Wear group disabled) // 1
TC_MPP4_14   = “P”      ;PlaceStatus Bit14 from 1 to 16 // 1
TC_MPP4_15   = “P”      ;PlaceStatus Bit15 from 1 to 16 // 1
TC_MPP4_16   = “P”      ;PlaceStatus Bit16 from 1 to 16 // 1
```

Coding of location status and tool status

Location statuses		Tool statuses	
G	Disabled location	G	Disabled tool
F	Free location	F	Released tool
Z	Reserved for tool in buffer	A	Active tool
B	Reserved for tool to be loaded	M	Measured tool
L	Left half location occupied	V	Prewarning limit reached
R	Right half location occupied	W	Tool is being changed
O	Upper half location occupied	P	Fixed location coded tool
U	Lower half location occupied	E	Tool has been in use
l	Left half location reserved	R	Unloading marking
r	Right half location reserved	B	Loading marking
o	Upper half location reserved	S	Master tool
u	Lower half location reserved		

4.4.3 Language-dependence for user-defined name

Language-dependent name for magazine location type

Function

The magazine location types (= location types) and their identifiers/names are entered by the user via the tool management start-up tool (IW) in the *Location type* screen. This is why the assigned names are contained in the tool management database and not in a language DLL. In previous versions, this meant that they were not available in different languages.

The new functionality allows the user to create the location type names in different languages/locales.

You can achieve this by entering name texts in the tool management INI files for the location types configured in the database.

In future there will be two names for each location type:

- The standard name which is used internally (tool database) and
- an associated language-specific name which is displayed on the operator interface.

If the user does not assign a language-specific name, the standard name from the database is displayed.

The location type "*standard*" is contained in the original database shipped with the tool management. The following special handling applies for this location type:

- The default setting for all patm_xx.ini files contained in the scope of supply includes an entry in [Placetype_VISName] for the location type "*standard*" (see next section).
- The language-specific text from patm_xx.ini is displayed in the screen *Location types* of the tool management start-up tool (IW) for the location type "*standard*" even in the selection box *Name*.

Entries in the language-specific INI files

Users must make the entries described here themselves. They are not written to the to the INI files by HMI_ADV.

Exception: Standard="Location type standard".

The language-specific INI files are called *patm_xx.ini* and can be found under ../hmi_adv/language. The user-defined files *patm_xx.ini* are found under ../user/language.

File: patm_xx.ini
Section: [Placetype_VISName]
Entry: Standard name="Language-specific text"

Example: [Placetype_VISName]
Standard="Location type standard".
SmallPlaceType="small"
. . .

For newly entered texts from the INI files to be activated in the display, you need to change the language setting or start HMI_ADV again.

Display location type names in the HMI tool management screens

The language-specific names of the location types are displayed in all the tool management screens and the tool management start-up screens. If there are no entries in the corresponding INI files, the standard name from the tool database is displayed.

Affected screens/functions:

Tool management:	tool details Tool new Tool catalog Tool cabinet Empty location search
Tool management start-up:	magazine configuration Location types

Tool catalog/tool cabinet

The standard name of the assigned location type is maintained and internally processed for each tool in the tool catalog/cabinet.

The language-specific name is displayed in the tool catalog/cabinet screens for the location type. If there is no language-specific name, the standard name from the tool database is used instead.

Code carrier

The name of the magazine location type which is written via the dialog variable T8 to the code carrier chip always corresponds to the standard name of the magazine location type from the tool database.

Alternatively, the name of the magazine location type can be written to the code carrier via dialog variable T12.

T8 and T12 are also accepted. When the chip is read in, a cross check is performed to associate magazine location type name and number.

Language-dependent name for buffer

Function

The buffer locations and their identifiers/names are entered by the user via the tool management start-up tool (IW) in the *Buffer* screen. This is why the names are specified in the tool management database and not in a language DLL. In previous versions, this meant that they were not available in different languages.

Until now two options were available for displaying the buffer locations in the tool management (not tool management start-up):

- Display names from the tool database (*paramtm.ini, [TMMODES] NameOfBufferPlaceFrom=DB*, see next section)
- Display the type of buffer location from the language DLL plus the associated index. I.e., for example, *Spindle1, Spindle2* or *Gripper2 etc.* (*NameOfBufferPlaceFrom=DLL*)

The new functionality allows the user to create the buffer location names in different languages/locales.

For this display it is necessary to select *display name from the database* (*NameOfBufferPlace From=DB*).

The user must enter a corresponding name text in the language-specific tool management INI files for each buffer location configured in the database.

From now on there are two names for each buffer location: The standard name which is used for internal processing (tool database) and an associated language-specific name which is displayed on the operator interface.

If the user does not assign a language-specific name, the standard name from the database is displayed (as was the case up to now).

The tool management startup tool does not evaluate the entry *NameOfBufferPlaceFrom* and always uses the name from the tool database and the language-specific INI files.

Entries in the language-specific INI files

Users must make the entries described here themselves. They are not written to the to the INI files by HMI_ADV.

The language-specific INI files are called *patm_xx.ini* and can be found under ../hmi_adv/language. The user-defined files *patm_xx.ini* are found under ../user/ language.

The INI files *paramtm.ini* and the associated description file *paramtm.txt* reside under ../hmi_adv. The user-defined file paramtm.ini resides under ../user. As it is the default setting, there is no need to make an entry in the user-defined paramtm.ini.

File	paramtm.ini
Section:	[TMMODES]
Entry:	NameOfBufferPlaceFrom=DB (default setting)

and

File:	patm_xx.ini
Section:	[BufferPlace_VISName]
Entry:	Standard name="Language-specific text"

Example: [BufferPlace_VISName]
Spindle1="Main spindle"
Gripper1="1st gripper"
. . .

For newly entered texts from the INI files to be activated in the display, you need to change the language setting or start HMI_ADV again.

Display buffer names in the HMI tool management screens

The language-specific names of the buffer locations are displayed in all the relevant tool management screens and tool management start-up screens. If there are no entries in the corresponding INI files, the standard name from the tool database is displayed.

<u>Affected screens/functions:</u>

Tool management:	Magazine list, with display of buffer
Tool management start-up:	buffer

Language-dependent name for loading locations

Function

The loading locations and their identifiers/names are entered by the user via the tool management start-up tool (IW) in the *Loading locations* screen. This is why the names are specified in the tool management database and not in a language DLL. In previous versions, this meant that they were not available in different languages.

The new functionality allows the user to create the loading locations names in different languages/locales.

You can achieve this by entering name texts in the tool management INI files for the loading locations configured in the database.

In future there will be two names for each loading location:

- The standard name which is used internally (tool database) and
- an associated language-specific name which is displayed on the operator interface.

If the user does not assign a language-specific name, the standard name from the tool database is displayed.

This also applies to the first location in the loading magazine which is automatically assigned:

The tool management start-up tool (IW) automatically creates an entry in the tool database for the first location in the loading magazine. This occurs at first access to the loading locations screen with an original database. This location must always exist, therefore it is not possible to delete it.

It is assigned the internal standard name *"FirstLoadingPoint"* with the following characteristics:

- The default setting for all patm_xx.ini files contained in the scope of supply includes an entry in [LoadLocation_VISName] for the location type "FirstLoadingPoint" (see next section).
- In the loading locations screen for the tool management start-up tool (IW) the language-specific text from patm_xx.ini is also displayed for the first loading location in the *Name* selection box.

Already existing databases in systems that have already been operating for some time:

In older HMI versions, this 1st loading point was called *"Loading point for spindle"* or *"Loading point manual"* (in the language set at this point in time).

If it is detected at the first start of tool management or associated start-up tool (IW) with the functionality described here, then the existing name in the database can be replaced with *"FirstLoadingPoint"*.

Entries in the language-specific INI files

Users must make the entries described here themselves. They are not written to the to the INI files by HMI_ADV.

<u>Exception:</u> FirstLoadingPoint="Loading point manual"

The language-specific INI files are called *patm_xx.ini* and can be found under ../hmi_adv/language. The user-defined files *patm_xx.ini* are found under ../user/language.

File: patm_xx.ini
Section: [LoadLocation_VISName]
Entry: Standard name="Language-specific text"

Example: [LoadLocation _VISName]
FirstLoadingPoint="Loading point manual"
Loading station1="Main loading station"
. . .

For newly entered texts from the INI files to be activated in the display, you need to change the language setting or start HMI_ADV again.

Display loading locations in the HMI tool management screens

The language-specific names of the loading locations are displayed in all the relevant tool management screens and tool management start-up screens. If there are no entries in the corresponding INI files, the standard name from the tool database is displayed.

Affected screens/functions:

Tool management: Load
Unload
Empty location search
Positioning

Tool management start-up: Loading locations

Language-dependent name for magazines

Function

The magazine identifiers/names are configured by the user in the *Magazines* screen via the tool management start-up tool (IW). This is why the assigned names are contained in the tool management database and after loading a magazine configuration also in the NCK but not in a language DLL. In previous versions, this meant that they were not available in different languages.

The new functionality allows the user to create the magazine names in different languages/locales.

You can achieve this by entering name texts in the tool management INI files for the magazines configured in the tool database.

In future there will be two names for each magazine:

– The standard name (also known by the NCK) which is used for functional operation and

– an associated language-specific name which is displayed on the operator interface.

If the user does not assign a language-specific name, the standard name from the NCK is displayed in the tool management; and the standard name from the tool database is displayed in the startup tool.

Entries in the language-specific INI files

Users must make the entries described here themselves. They are not written to the to the INI files by HMI_ADV.

The language-specific INI files reside under ../hmi_adv/language.

File: patm_xx.ini
Section: [Magazine_VISName]
Entry: Standard name=“Language-specific text”

Example:

```
[Magazine_VISName]
Chain1=“Chain magazine 1”
Turret1=“Turret 1”
. . .
```

For newly entered texts from the INI files to be activated in the display, you need to change the language setting or start HMI_ADV again.

Display magazine names in the HMI tool management screens

The language-specific names of the magazines are displayed in all the tool management screens and the tool management start-up screens. If there are no entries are the corresponding INI files, the standard names from the NCK (in the tool management) or from the tool database (in the startup tool) are displayed.

Affected screens/functions:

Tool management:	Magazine list Magazine selection
Tool management start-up:	Magazine Buffer Loading locations Magazine configuration

4.4.4 Job processing of tools

The settings for batch processing of tools are contained in paramtm.ini and language\patm_xx.ini in the section [BatchTools].

Notice

The filter only works if bit 4 is set in MD 18080: TOOL_MANAGEMENT_MASK.
Up to 6 search filters can be defined.
The following can be specified for each filter: Softkey text, results title, search criteria, selection of the type of results list and additional data.
The file ...user\paramini.out contains error messages occuring when reading in the parameter assignment.

Settings that are valid for one filter at a time

Search criteria

The search criteria are set in “n_FindCondition” (with n=1 to 6) for each filter.

A maximum of 8 part conditions are permissible, separated by “,”. They are ANDed.

No data may occur more than once in the part conditions.

Each part condition consists of three parts:

1. Datum for which the condition applies
2. Condition
3. Comparison value

Example

```
1_FindCondition = TC_TP8 && 20, A_TOOLMN > 0
```

Filter 1 filters tools in the NC which fulfill the following condition:

((prewarning bit set (bit5 of bit1 to 16 (2 to the power of (5–1)=16))))

OR

(disabled bit set (bit3 (2 to the power of (3–1)=4)))

)

AND

((magazine number > 0 means "loaded tool"))

Filter criterion

The following data in the NC can be a filter criterion:

Tool data

TC_TP1	Duplo number
TC_TP2	Tool identifier
TC_TP3	Tool size in half locations left
TC_TP4	Tool size in half locations right
TC_TP5	Tool size in half locations top
TC_TP6	Tool size in half locations bottom
TC_TP7	Tool location type
TC_TP8	Tool status
TC_TP9	Monitoring type
TC_TP10	Replacement tool search
TC_TP11	Tool information/replacement tool sequence
A_TOOLMN	Magazine number
A_TOOLMLN	Magazine location number
P_TOOLND	Number of cutting edges

Tool cutting edge parameters

TC_DP1 to TC_DP24

Tool cutting edge monitoring parameters

TC_MOP1	Prewarning limit for tool life
TC_MOP2	Actual value for tool life
TC_MOP3	Prewarning limit for workpiece count
TC_MOP4	Actual value for workpiece count
TC_MOP5	Prewarning limit for wear
TC_MOP6	Remaining wear
TC_MOP11	Setpoint for tool life
TC_MOP13	Setpoint for tool workpiece count
TC_MOP15	Setpoint for wear

Tool OEM data

TC_TPC1 to TC_TPC10
The tool cutting edge OEM data must be activated on the NC and the numbers must be permissible on the NC.

Tool cutting edge OEM data

TC_DTPC1 to TC_DTPC10
The tool cutting edge OEM data must be activated on the NC and the numbers must be permissible on the NC.

Tool cutting edge monitoring OEM data:

TC_MOPC1 to TC_MOPC10
The tool cutting edge monitoring OEM data must be activated on the NC and the numbers must be permissible on the NC.

Conditions

The following conditions can be used:

==	equal to
<	less than
>	greater than
<=	smaller than or equal to
>=	greater than or equal to
&&	bit-wise AND between screen and data; only permissible for operands of type WORD and DOUBLEWORD; the individual result bits are ORed. If this condition is applied to the tool status, you can locate tools with specific set (AND) tool status bits. If several set bits are queried at the same time, just one set bit is sufficient for the tool to appear in the hit list.

For string operands (e.g. tool identifier) "==" is the only permissible relational operator.

Comparison value

The following value ranges apply:

String	with TC_TP2, maximum of 32 characters, no blanks before or after
0 ... 65535	for other TC_TP data
Double	for all other data

Additional data

Max. one column with additional data can appear for each filter in the results list.

There are three settings for each filter:

- `n_FindResultAddColumnText`
  Header text for column or reference to the header text with language-specific settings/locales.
- `n_FindResultAddColumnDisplayedNumberOfCharacters`
  Column width in characters
- `n_FindResultAddColumnBtss`
  OPI item acc. to OPI documentation (mmc2\btss_gr.hlp).

Example

Paramtm.ini, [BatchTools]:

1_FindResultAddColumnText=<empty>
; No additional column

1_FindResultAddColumnText=R1AddCol
; Language-specific

1_FindResultAddColumnDisplayedNumberOfCharacters=8

1_FindResultAddColumnBtss="/Tool/User/data[u#TOA#,c3,#TNO#] (l"!l%ld")"

language\patm_gr.ini, [BatchTools]:

R1AddCol="Additional data search 1"

`n_FindResultAddColumnBtss:`

Additional data, OPI item acc. to OPI documentation (mmc2\btss_gr.hlp).

Example 1

`"/Tool/User/data[u#TOA#,c2,#TNO#](l"!d%.#RES#lf")"`

Tool OEM parameter2, floating point representation, standard number of decimal places

Example 2

`"/Tool/User/data[u#TOA#,c3,#TNO#](l"!l%ld")"`

Tool OEM parameter3, integer representation

Example 3

`"/Tool/MagazineDescription/userData[u#TOA#,c#MAG,1](l)"`

Magazine OEM parameter 1

Placeholder

The following placeholders are permissible:

#TOA#	TOA number (of the current channel)
#TNO#	Internal T number (of the tool found)
#MAG#	Magazine number (of the found tool)
#RES#	Standard value for the number of places after the decimal point

Placeholders are substituted by the data for the current tool or by general settings.

Max. 1 OPI item is permitted.

OPI multiple variable accesses are generated internally from the OPI item.

“(|)” is entered in front of the result data to generate the data separation character “|”.

The OPI item must enclosed by “ ” especially when formatting information (e.g. “!l%ld”) is contained in “ ”.

The syntax is not checked by the operator interface software. The person setting the parameters is responsible for the correct syntax.

Notice

If errors are made in the parameter settings, the result list can no longer be displayed or the secondary faults can occur.

Filter name and softkey text

You can set one name and one softkey text in each language for each filter.

There are two settings for each filter:

- `n_FindResultHeadlineText`
  Text for filter name or reference
- `n_FindSoftkeyText`
  Softkey text for the filter or reference (a double blank in the text defines a new line)

Example

```
Paramtm.ini, [BatchTools]:

    1_FindResultHeaderText = R1HL

    1_FindSoftkeyText = F1SK

language\patm_gr.ini, [BatchTools]:

    R1HL = “prewarning limit reached or disabled”

    F1SK = “prewarning or disabled”
```

Result list type

You can select the result list type for each filter. determines which job functions are available for each softkey. There is one setting for each filter:

– n_ResultListType
 Result list type, value range:
 0: Standard list (default) for unload, delete, into the cabinet Reactivate
 1: Load list for loading, reactivate

Example

```
Paramtm.ini, [BatchTools]:
   1_ResultListType = 0       ;0 = standard list
```

Filter restricted to one magazine

For each filter you can select whether it is restricted to a specific magazine. This should be visible in the filter name.

There is one setting for each filter:

– n_FindLimitedToCurMagazine
 Filter restricted to a specific magazine, value range:
 TRUE: hit list restricted to current magazine
 FALSE: (default) hit list is not restricted to the current magazine

Example

```
Paramtm.ini, [BatchTools]:
   1_FindLimitedToCurMagazine=true
   ;“True”, “False”; can be restricted to current magazine
```

Positioning during Reactivate

For each filter you can select whether the job function can be selected, whether with job function “Reactivate” the tool is to be positioned in a loading point.

There is one setting for each filter:

– n_ReactivatePositioningMode
 Position at reactivate, value range:
 0: do not position
 1: ask the operator whether to position for each complete job
 2: (default) always position

Example

```
Paramtm.ini, [BatchTools]:
   1_ReactivatePositioningMode=2   ;always
```

Parameter for PI TSEARCH

Notice

Siemens reserves the right to withdraw support in future versions.

The parameters for the PI TSEARCH used for filtering can be specified for each filter (see PI documentation `pi_gr.hlp`).

This setting is very sensitive to errors. It does not support insertion of blanks, the number of places must be strictly adhered to; the character string must be contained in “ ”.

There is one setting for each filter:

– `n_FindPiSearchPar`
  8 parameters for PI SEARCH

  #Mag# can be used as placeholder for the magazine setting. Constant, five-digit magazine numbers can also be specified for the from/to magazine range instead of the placeholder.

  If the value in the 8th parameter is set to “2”, filter criteria for cutting edge specific data will also be used correctly for multi-point cutting tools (from NCK version NCK.P6_43 and NCK.P5_20.4).

Example

```
Paramtm.ini, [BatchTools]:
1_FindPiSearchPar=“#Mag#,-0001,#Mag#,-0001,00000,00001,1,2”
```

General settings that apply for all filters at the same time

Colors for the results list

The colors for the results list are customizable. When setting colors, please avoid color combinations that are difficult to read or too bright.

The parameter settings consist of eight colors separated by comma. A hex value consisting of 8 characters is assigned to each color. The hex value has the following syntax:

SSBBGGRR mit *SS*=System, *BB*=Blue, *GG*=Green, *RR*=Red.

The colors have to be specified for the following list elements:

- Non-selected text
- Non-selected background
- Cursor-selected text
- Cursor-selected background
- Job-selection and cursor-selected text
- Job-selection and cursor-selected background
- Job-selection and cursor-selected text
- Job-selection and cursor-selected background

Examples for colors:

- 80000008 Windows text
- 80000005 Windows background
- 8000000E Highlighted Windows text
- 8000000D Highlighted Windows background
- 80000009 Windows active window header text
- 80000002 Windows active window header background
- 00FFFF00 Light blue
- 0000FF00 Green
- 00FF8000 Blue-green

Examples for setting colors, see [BatchTools], “General settings which apply to all filters” in paramtm.txt.

Bitmaps for the status display of the individual job elements

The user can replace the names of the bitmaps or the bitmaps themselves with custom bitmaps. The custom bitmaps are stored in the “user” directory.

Examples for setting bitmaps, see [BatchTools], “General settings which apply to all filters” in paramtm.txt.

Example

```
BatchFilterElBUnTUnBitmap=   pbfbutu.bmp
BatchFilterElBUnTSeBitmap=   pbfbuts.bmp
BatchFilterElBSeTUnBitmap=   pbfbstu.bmp
BatchFilterElBSeTSeBitmap=   pbfbsts.bmp
BatchRunElWaitingBitmap  =   pbbwait.bmp
BatchRunElInWorkBitmap   =   pbbwork.bmp
BatchRunElOKBitmap       =   pbbok.bmp
BatchRunElErrorBitmap    =   pbberr.bmp
```

Width of a typical character

For each language you can specify a character whose width is used as the basis to calculate column widths from a specified number of characters. A wide character should be entered here, in Europe typically an “X” or “A”.

Example

```
Paramtm.ini, [BatchTools]:
CharToGetColWidthPerCharacter=CharToGetColWidth ;language-specific
language\patm_gr.ini, [BatchTools]:
CharToGetColWidth=“A”
```

Column width for tool identifier

You can set the column width for the tool identifier as as a rule the full number of 32 characters is not used.

Example

```
Paramtm.ini, [BatchTools]:
ResultDisplayedNumberOfToolnameCharacters=18
```

Tool status bits

You can set which tool status bits are to be displayed in the results list. Language-specific letters can also be specified in bits for header and list lines.

Example

`Paramtm.ini, [BatchTools]:`

`ResultToolStatusColumnsEnable=    1111100100110000`
1: display, 0: do not display. Bit 1 to 16,
bit 1 is the least-significant bit in the tool status and is positioned to the left in this character sequence

`ResultToolStatusColumnsHeaderText=<Empty>`
Text in the header for the tool status column, language-specific

`ResultToolStatusColumnsListText=<Empty>`
Text in the data for the tool status column, language-specific

`ToolStatusColHeaderText="123456789ABCDEFG"`
Header; Bits 1 to 16, bit 1 is the least-significant bit in the tool status and is positioned to the left in this character sequence

`ToolStatusColListText="123456789ABCDEFG"`
Data; Bits 1 to 16, bit 1 is the least-significant bit in the tool status and is positioned to the left in this character sequence

User authorizations

User authorizations for the associated softkeys can be set in `paramtm.ini`, section `[ACCESSLEVEL]`, entries "SKB...".

Example

```
[ACCESSLEVEL]
SKBATCH=7            ; Softkey filter lists
SKFILTER1=7          ; Softkey Filter1
SKFILTER2=7          ; Softkey Filter2
SKFILTER3=7          ; Softkey Filter3
SKFILTER4=7          ; Softkey Filter4
SKFILTER5=7          ; Softkey Filter5
SKFILTER6=7          ; Softkey Filter6
SKBMAGFILTER=7       ; Softkey magazine selection
SKBATREACT=7         ; Softkey batch function "Reactivate"
SKBATTOCABIN=7       ; Softkey batch function "In cabinet"
```

```
SKBATDELTOOL=         ; Softkey batch function “Delete”
SKBATUNLOAD=7         ; Softkey batch function “Unload”
SKBFILTERACT=7        ; Softkey batch function “Update”
SKBATLOAD=7           ; Softkey batch function “Load”
SKBATLIST=7           ; Softkeys for controlling the job processing
```

4.4.5 Grinding tools and tool-specific grinding data

The HMI Advanced tool management is modified so that the “tool-specific grinding data” of grinding tools is displayed and can be edited.

This data is exchanged with the NC via the OPI block TG; it largely corresponds to the NC variables $TC_TPG1 to $TC_TPG9 (see Section 5.3.2).

For more information on softkey extension in the tool details main screen, tool details cutting edge data screen and tool detail monitoring data screen, see:

References: /BAD/ Operator’s Guide HMI Advanced, Edition 11.02

Setting parameters for the default values

The parameters for the default values for the tool-specific grinding data when creating tools are set in the “paramtm.ini file, section `[DEFAULT_SETTINGS]`.

Description

paramtm.ini:

[DEFAULT_SETTINGS]

```
;!!!  Default setting of grinding-specific tool data at creation:
;!!!  If the machine operates with inch/mm conversion ($MN_CON-
;!!!  VERT_SCALING_SYSTEM=1), the unit of length must be specified!!!
; The following default values (TOOLGRIND..., if affected by the unit
; of length) are specified in relation to this basic unit of length:
; 0 = mm (default)
; 1 = Inch
TOOLGRIND_Default_Length_Unit=0

;    Spindle number       (as $TC_TPG1)
TOOLGRINDspinNoDress=1

;    Chain rule    (as $TC_TPG2)
TOOLGRINDconnectPar=1050629
```

```
    ;1050629 binary: 0000 0000 0001 0000 0000 1000 0000 0101
    ;Bit0  =1 = Type
    ;Bit2  =1 = Geo-L1
    ;Bit11 =1 = Wear-L1
    ;Bit20 =1 = Base-L1
;   Minimum wheel radius   (as $TC_TPG3)
TOOLGRINDminToolRadius=0
;   Minimum wheel width    (as $TC_TPG4)
TOOLGRINDminToolWide=0
;   Current width of grinding wheel     (as $TC_TPG5)
TOOLGRINDactToolWide=0
;   Maximum grinding wheel speed        (as $TC_TPG6)
TOOLGRINDmaxRotSpeed=0
;   Maximum grinding wheel peripheral speed (as $TC_TPG7)
TOOLGRINDmaxTipSpeed=0
;   Inclination angle of inclined wheel     (as $TC_TPG8)
TOOLGRINDinclAngle=0
;   Compensation parameter for grinding wheel peripheral speed
;                                           (as $TC_TPG9)
TOOLGRINDparamNrCCV=3
```

Notice

The HMI Advanced function “Change tool type” used up to now is not modified. Therefore, with grinding tools too, when the tool type is changed, most tool data is set by the HMI to “0”.
The grinding-specific tool data is not set to “0”; instead it is processed by the NCK.

4.4.6 Inch/metric setting

Now the tool database (tool cabinet, tool catalog) and code carrier will support machining in inch or mm measurements.

The entries DATABASE_LENGTH_UNIT and CODECARRIER_LENGTH_UNIT in the paramtm.ini file in the section [TMMODES] (see Section 4.4.2) allows the operator to set the behavior for length units for the tool cabinet, tool catalog and code carrier.

Tool database

NCK with inch/mm data conversion

[TMMODES]

!! CAUTION: The settings described here are only relevant if machine data *$MN_CONVERT_SCALING_SYSTEM=1* is set in the NCK. The entry for DATABASE_LENGTH_UNIT is only evaluated if no unit is entered yet in the tool database. This means once in a normal case scenario!

-1 = **No setting for inch/metric** in the tool database (default). The setting $MN_CONVERT_SCALING_SYSTEM=1 in the NCK means that inch/ metric conversion is to be used for machining. Therefore, the user **must** specify **which unit applies for the data present in the tool database**. This is not the case with setting -1, as a result all softkeys are disabled for database activities.

0 = **mm.** The first time a tool database is **opened** by a tool management differentiating between inch/mm, it is ascertained that **mm** applies for the relevant data in the **tool database**.
An entry is added to the database to indicate that the measurement unit for the relevant tool data in the database is mm. This is taken into account for future data transfer between database and NCK.

1 = **inch.** The first time a tool database is **opened** by a tool management differentiating between inch/mm, it is ascertained that **inch** applies for the relevant data in the **tool database.**
An entry is added to the database to indicate that the measurement unit for the relevant tool data in the database is inch. This is taken into account for future data transfer between database and NCK.

DATABASE_LENGTH_UNIT=-1

If DATABASE_LENGTH_UNIT is assigned the setting 0 or 1, the following occurs: The first time the new tool management is started with inch/ metric differentiation, the database receives a new entry which defines the unit of measurement for the database for future use.

Tabelle 4-1 **TM_Info**

Name	Type	Size
InfoKey_Name	Text	255
InfoKey_Index	Integer	2
Info_String	Text	255
Info_Num	Double	8

Contents of table 4-1 for data in mm or inch:

InfoKey_Name	InfoKey_Index	Info_String	Info_Num
“BasicLenUnit”	0	“mm”	0
“BasicLenUnit”	0	“inch”	1

The **unit entered in the database is valid for future processing**. From now on, the data will be written to and read out from the tool cabinet in the specified unit.

Tool management active

During operation with active tool management you can switch between inch/mm units of measurement as follows:

1. Via softkey e.g. in the machine. Then run the tool management. When selected again the data is displayed in all screens in the new unit.
2. Switchover by changing the machine data $MN_SCALING_SYSTEM_IS_METRIC and Power ON reset or machine control panel reset in start-up. Then run the tool management. When selected again the data is displayed in all screens in the new unit.
3. Switchover by changing the machine data $MN_SCALING_SYSTEM_IS_METRIC and Power ON reset. During the reset process switchover is according to tool management.
   - Tool details screens: the focus returns to the associated list display
   - List displays: The data is updated accordingly. If necessary (due to reset) a screen change takes place
   - Catalog/cabinet displays with affected data: The display screen is exited with <Cancel>
4. Switchover by changing the machine data $MN_SCALING_SYSTEM_IS_METRIC. Then call tool management and Power ON reset in the already selected tool management. This corresponds to point 3.
5. Switchover by changing the machine data $MN_SCALING_SYSTEM_IS_METRIC. Then call tool management and machine control panel reset in the already selected tool management.
   - Tool details screens and list displays: As the individual values are immediately written after input and are constantly updated in these displays, the data is immediately displayed in the new unit.
   - Catalog/cabinet displays with affected data: As the entered data is only written in full to the database when a softkey with “Save function” (e.g. <OK>, <New cutting edge>) is activated, the new unit only becomes active after this type of action.

Code carrier

For setting inch/metric for code carriers, see Section 4.5.2.

4.5 Further settings

4.5.1 Display machine data with HMI Embedded

With software Version 4 and higher, HMI Embedded allows settings to be made via display machine data and access rights to be assigned for specific functions (see Section 2.9).
For a more detailed description of this machien data please refer to Section 8.1.1.

Additional user parameters

If additional user data was created via NCK machine data (user parameters for cutting edge and/or tool data), this data is displayed in additional screens.

The data is administered but not evaluated by the tool management.

User texts

The HMI Embedded software is shipped with an application diskette which allows users to set parameters for the custom data.

The table shows which texts can be parameterized. They are stored in the pa.txt file. Custom text can be entered under “User text”.

User cutting edge data, texts

Name of the text	Custom text		
T_EDGE_TEXT_1TT_EDGE_TEXT_1”	T_TM_OEM_CUT_TM_OEM_CUT	47”	72
. . .	. . .		
. . .	. . .		
T_10T_TM_OEM_CUTT_EDGE_TEXT_10”	“T_TM_OEM_CUTT_EDGE_TEXT_10”	47	72

Custom tool data, texts

Name of the text	Custom text		
T_TM_OEM_TOOL_TEXT_1	“T_TM_OEM_TOOL _TEXT_1”	47	72
. . .	. . .		
. . .	. . .		
T_TM_OEM_TOOL _TEXT_10	“T_TM_OEM_TOOL_TEXT_10”	47	72

Fine offsets for cutting edge data

With software version 5.2 and higher, you can set whether WRITE_TOA_FINE_LIMIT and USER_CLASS_WRITE_FINE apply to the cutting edge data via MD 9449: WRITE_TOA_LIMIT_MASK.

The bits set in MD 9449 determine whether the display machine data WRITE_TOA_FINE_LIMIT and USER_CLASS_WRITE_FINE apply to the cutting edge data type or not. If the bits are not set, FINE_LIMIT is not used.

Bit assignment for MD 9449

The bits are assigned as follows for MD 9449: WRITE_TOA_LIMIT_MASK.

Bit	Application	Default value
Bit 0	Cutting edge data (offsets), wear data	1
Bit 1	SC data (local offsets and the wear values for these)	1
Bit 2	EC data (local offsets and the setup values for these)	1

7 is the default value for MD 9449. The FINE_LIMIT is applied to all data types.

Compatibility of fine offsets for HMI Advanced and Embedded

Up to SW 5.2, the machine data WRITE_TOA_FINE_LIMIT and USER_CLASS_WRITE_FINE were used for the geometry, basic and wear parameters in the tool management for the MMC 103.

In the tool management for HMI Embedded, this machine data is active only on the wear parameters of the cutting edges. As of SW 5.2, the two MDs are active only on the wear parameters of the cutting edges in the tool management for HMI Advanced.

Changing the compatibility of the fine offsets

An entry in the paramtm.ini can restore the old response of the tool management for HMI Advanced. The MD WRITE_TOA_FINE_LIMIT and MD USER_CLASS_WRITE_FINE is again applied to the geometry, basic and wear parameters.

Entry in paramtm.ini:

```
[General]
; Application of
; $MM_WRITE_TOA_FINE_LIMIT and $MM_USER_CLASS_WRITE_FINE
; on geometry values and basic values of cutting edge data
UseFineLimitForToolGeoAndAdapt=False            ;default
;UseFineLimitForToolGeoAndAdapt=True            ;
```

4.5.2 Start-up of code carrier

Notice

From SW 6.3 code carrier systems can only be operated via SinTDC and no longer directly.
WToolIdSys = SinTDC
See also /FBTC/ SINUMERIK Tool Data Communication SinTDC.

For the code carrier system see also Section 3.13 and the Description of Functions of the individual tool identification systems.

A code carrier system is connected to the HMI e.g. via an RS-232 (V.24) interface.

If the machine has its own code carrier system (tool identification system), then this system must also be started up separately.
This is carried out by running a setup program for the code carrier system and making settings in the associated INI files (see corresponding Description of Functions for the tool identification system).

Description of the code carrier files

In order for the code carrier system to be activated from the tool management, it must be entered in the file ...\user\mmc.ini (prior to SW4 ...\mmc2\mmc.ini).

Settings for inch/metric units of measurement and validation of tool status bits can be made in the file ...\user\paramtm.ini).

Notice

None of the INI files in the “mmc2” directory may be modified.

INI file

An INI file is associated with every manufacturer-specific server (“exe file”) Manufacturer-specific settings are made in this INI file via the code carrier system. The parameters it contains are described in the documentation from the code carrier manufacturer or the respective Description of Functions for the tool identification system.

Adapting file “mmc.ini”

The connected code carrier system (e.g. WToolIdSys=Ballu) is activated in the file ...\user\mmc.ini.

Notice

From SW 6.3 code carrier systems can only be operated via SinTDC and no longer directly.
WToolIdSys = SinTDC
See also /FBTC/ SINUMERIK Tool Data Communication SinTDC.

```
– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –
[ToolMgmt]
– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –
WToolIdSys=0 ; or Ballu
; Identifier for code carrier system
; 0          means :"No code carrier active"
; Specify manufacturer name (only first 5 characters!)
; Ballu      means: Code carrier from manufacturer Balluff is active

WToolIdSysKonv=wkonvert.txt
; Name of the conversion file used for the code-carrier format.
; The file resides in directory ...\add_on or ...\user.

[TIS]
; Tool Identification System
; EOT for code-carrier data
TIS_EOT=0x2F2F
; The end identifier for data has to be entered on the code carrier
; here.
```

Adapt the file paramtm.ini

The excerpts printed below are found in the paramtm.ini. In newer software versions, the paramtm.ini comments have been summarized in paramtm.txt.

As a general rule, the desired entries should be made in ...\user\paramtm.ini so that they are retained during the next HMI software update.

Settings for inch/metric

If the paramtm.ini or paramtm.txt of the software version that has been installed contains one of the following (variant 1 or variant 2), then the function can be used by making an entry in ...\user\paramtm.ini as described below. Otherwise the default setting applies.

Variant 1

```
[TMMODES]
. . .
; The setting inch/metric is considered for the code carrier
; -1 =  inch/metric is ignored (default). The data exchange between
;       code carrier and NCK/MMC takes place without taking into ac-
;       count inch/metric. Behavior as up to now.
; 0 =   mm. It is assumed that storage for the affected data was or
;       is to be in the unit “mm” on the code carrier. If “inch” is
;       set in the NCK, then all softkeys are disabled that start
;       code carrier functions.
; 1 =   inch. It is assumed that storage for the affected data was or
;       is to be in the unit “inch” on the code carrier. If “mm” is
;       set in the NCK, then all softkeys are disabled that start
;       code carrier functions.
DATABASE_LENGTH_UNIT=-1
```

Variant 2

```
[TMMODES]
. . .
; The setting inch/metric is considered for the code carrier
; !! CAUTION: The settings described here are only relevant if ma-
; chine data $MN_CONVERT_SCALING_SYSTEM=1 is set in the NC.
; If the NC is an older mode without the inch/metric conversion func-
; tionality or if $MN_CONVERT_SCALING_SYSTEM=0, the NC will operate
; without inch/metric conversions. Therefore no conversions are car-
; ried out in relation to the code carrier!!

; -1 =  No definition for inch/metric in relation to code carrier
;       default). The setting $MN_CONVERT_SCALING_SYSTEM=1 in the NC
;       means that inch/metric conversion is to be used for machin-
;       ing. Therefore, the user must define in which unit the data
;       is present on the code carrier or should be written to it. As
;       with setting -1 this does not take place, all softkeys for
;       code carrier activities are disabled.
; 0 =   mm. All affected data is written to the code carrier as mm
;       values in future.
;       This is taken into account for future data transfer between
;       code carrier and NC.
; 1 =   inch. All affected data is written to the code carrier as
;       inch values in future.
;       This is taken into account for future data transfer between
;       code carrier and NC.
CODECARRIER_LENGTH_UNIT=-1
```

Setting for tool status

If the paramtm.ini or paramtm.txt of the software version that has been installed contains one of the following entries, then the function can be used by making an entry in ...\user\paramtm.ini as described below. Otherwise the default setting applies.

```
; Tool status: If a tool is removed from the NCK and transferred to
; an external medium (tool cabinet, code carrier, SINCOM), then you
; can use the following screens to specify which tool status bits are
; to be saved
; Code carrier: Since 1 byte was entered for the tool status in the
; standard conversion file wkonvert.txt and until now max. 92 has
; been written to the code carrier, CODECARRIER_TOOLSTATE_MASK is
; assigned the default value 92.
; If the value for CODECARRIER_TOOLSTATE_MASK is expanded, the size
; of the dialog variable T9 must be adapted accordingly in wkon-
; vert.txt.
; 1=Active Tool
; 2=Allowed
; 4=Disabled
; 8=Measured
; 16=Warning limit reached
; 32=In change
; 64=Fixed place coding
; 128=Was used
; 256=Tool in buffer
; 512=Disabled, ignored (because of PLC)
; 1024=Out (unload)
; 2048=in (Load)
; 4096=Regular tool (permanent in NCK)
; 8192=
; 16384=
; Default is 4828 (4+8+16+64+128+512+4096),
; Default for codecarrier 92 (4+8+16+64)
. . .
CODECARRIER_TOOLSTATE_MASK=92
```

Structure of description file

Description file

All data on the code carrier are stored in a particular order. This is defined during commissioning of the code carrier system. A conversion rule in the form of a description file is provided so that the tool management can read or write this data flow. This description file consists of correctly defined tool and cutting edge dialog data. Only this dialog data can actually be processed by the tool management. All the other data on the code carrier must not be assigned to any dialog variables as otherwise it will not be processed. An OEM application would, however, also be able to access this data.

The description file can be created as an ASCII file using a standard editor. The file name must be entered in mmc.ini with WToolIdSysKonv = **wkonvert.txt.**

Notice

Minimum requirements for wkonvert.txt:
Name
Location type
Subtype
Tool size (the part of the size can be omitted which is hidden via paramtm.ini *)
Number of cutting edges (if cutting edge data available)

* see paramtm.ini
SHOW_TOOLSIZE_ONLY_LEFT_RIGHT=0
SHOW_TOOLSIZE_COMPONENTS=left:=True, right:=True, top:=True, bottom:=True

---

Tool dialog data

The tool dialog data is defined as follows:

Dialog variable	Data type	Description	Assignment $TC...
T1	String	Tool name, max. 32 characters	$TC_TP2
T2	Integer	Duplo number	$TC_TP1
T3	Integer	Number of cutting edges	$P_TOOLND[tnr] tnr=tool number
T4	Integer	Tool size left in half locations	$TC_TP3
T5	Integer	Tool size right in half locations	$TC_TP4
T6	Integer	Tool size upper in half locations	$TC_TP5
T7	Integer	Tool size down in half locations	$TC_TP6

Dialog variable	Data type	Description	Assignment $TC...
T8	String	Magazine location type	$TC_TP7*
T9	Integer	Tool status	$TC_TP8
T10	Integer	Type of tool monitoring	$TC_TP9
T11	Integer	Type of tool search	$TC_TP10
T12	Integer	Magazine location type Previously, only the name of the magazine location type could be stored as a string via dialog variable T8. The assignment between location type number and location type name and vice versa is made in the tool management database.	$TC_TP7

* The character string which is stored there is an HMI internal location type which is assigned the value in $TC_TP7. This text is defined via the tool management start-up and stored in the database.

Notice

If chips are to be exchanged between several machines, the following rule applies if T12 is used:
The location type names must be present in the same sequence on all these machines (with the same location type numbers).

Notice

If changes are made to the conversion file, old code carriers can no longer be read!

Notice

Code carrier chip/SINTDC:

If the user has defined the dialog variable T11 in the conversion file wkonvert.txt for “Type of tool search, $TC_TP10”, the value from the NCK is written to the chip and written back to the NCK at read in. If T11 is missing, the value is 0 in $TC_TP10 after the chip is read in.

Cutting edge dialog data

Dialog variable	Data type	Description	Assignment $TC...
C1	Integer	Subtype	$TC_DP1
C4	Integer	Cutting edge position	$TC_DP2
		Geometry tool length compensation	
C5	Double	Length 1	$TC_DP3
C6	Double	Length 2	$TC_DP4
C7	Double	Length 3	$TC_DP5
		Geometry tool radius compensation	
C8	Double	Length 1	$TC_DP8
C9	Double	Length 2	$TC_DP9
C10	Double	Radius 1	$TC_DP6
C11	Double	Radius 2	$TC_DP7
C12	Double	Angle 1	$TC_DP10
C13	Double	Angle 2	$TC_DP11
		Wear tool length compensation	
C14	Double	Length 1	$TC_DP12
C15	Double	Length 2	$TC_DP13
C16	Double	Length 3	$TC_DP14
		Wear tool radius compensation	
C17	Double	Length 1	$TC_DP17
C18	Double	Length 2	$TC_DP18
C19	Double	Radius 1	$TC_DP15
C20	Double	Radius 2	$TC_DP16
C21	Double	Angle 1	$TC_DP19
C22	Double	Angle 2	$TC_DP20
		Base dimension/adapter dimension tool length compensation	
C23	Double	Basic length 1	$TC_DP21
C24	Double	Basic length 2	$TC_DP22
C25	Double	Basic length 3	$TC_DP23
C26	Double	Undercut angle	$TC_DP24
C27	Integer	Reverse insert	$TC_DP25
C28		Cutting edge number for addressing variables	–
C29 *	Integer	Downtime in minutes	$TC_MOP2

Dialog variable	Data type	Description	Assignment $TC...
C30 *	Integer	Prewarning limit for tool life in minutes	$TC_MOP1
C31	Integer	Number of pieces still to be produced	$TC_MOP4
C32	Integer	Prewarning limit for number of pieces still to be produced	$TC_MOP3
C33	Double	Set downtime in minutes	$TC_MOP11
C34	Integer	Unit quantity setpoint	$TC_MOP13
C35	Double	Prewarning limit for wear	$TC_MOP5
C36	Double	Wear	$TC_MOP6
C37	Double	Setpoint wear	$TC_MOP15
C38 *	Double	Downtime in minutes	$TC_MOP2
C39 *	Double	Prewarning limit for tool life in minutes	$TC_MOP1
C40 *	Double	Quantity	$TC_MOP4
C41	Double	Pre-warning limit for count	$TC_MOP3
C42	Double	Unit quantity setpoint	$TC_MOP13

The dialog variables C2 and C3 are managed only internally.

* see following note

Notice

C38 and C39 can only be used as alternatives for C29 and C30.

C40, C41 and C42 can only be used as alternatives for C31, C32 and C34.

User tool parameters and the new monitoring parameters are now used for code carriers. The following new dialog variables are available for the file “wkonvert.txt”:
A1 – A10: User tool data (see $TC_TPCx[t])
U1 – U10: User cutting edge data (see $TC_DPCx[t,d])
S1 – S10: User monitoring data (see $TC_MOPCx[t,d])

Data type “Double” is defined for the dialog variables A, U and S.

Data Types

The following data types are defined for dialog variables:

- Integer: value range –32768 to 32767
- Double: Floating point double precision
- String: Character sequence of ASCII characters

Keywords

The assignment of code-carrier data to dialog data is made using the code-carrier description file. The description file can be created and edited as an ASCII file using a standard editor. The code-carrier files is structured as lines whereby each line is prefixed by one of the following **keywords**:

Inverted comma

The **'** (single quotation mark) marks the beginning of a comment. The characters that follow are skipped.

Example:

```
' This is a comment
```

Notice

This format for the beginning of a comment is used only in the description file for code carriers. Otherwise, the beginning of a comment is introduced by a semicolon (;).

Datalen

DATALEN=CONST | VARIABLE 0x<delimiter>

The following data have a constant (**CONST**) or a variable (**VARIABLE**) data length. Data with variable length are terminated with 0x<delimiter>.

Example:

```
DATALEN=VARIABLE 0x0A        ' variable data length, delimiter LF
```

DEFINE_KEYWORD

DEFINE_KEYWORD=<keyword> <value><keyword> := any user keyword for indicating a new data section on the code carrier

<value> := “<string>” or 0x<hexvalue>

Definition of the keyword <keyword> with the value <value>

Example:

```
DEFINE_KEYWORD=DATA _OEM      "OEM"
DEFINE_KEYWORD=DATA_SIN840D   0x840D
```

<keyword>

A keyword defined by **DEFINE_KEYWORD** that identifies a new data section on the code carrier. The item in the code carrier description file following **<keyword>** must contain the value <value> defined by **DEFINE_KEYWORD**.

Item

Item<n>**=<**line**>**

<n> := Consecutive number of code carrier date, ascending from 1 without gaps
<line> := <(max.) length in bytes> <code carrier data format> <dialog variable>
<code carrier data format> :
<dialog variable> : Assignment of code carrier to dialog datum

If just before **Item<n>** a user keyword **keyword** was defined, then <dialog variable> has the value **<keyword>**

Conversion rule for code carrier data <n>

Example:

```
Item1    32    ASCII    T3       ’ Relocate tool identifier to/from tool
                                 ’ dialog datum 3
```

BItem

BItem<n>**=<**line**>**

<n> := Consecutive number of code carrier date within **Block**<i>, ascending from 1 without gaps

<line> := analog **Item**<n>

Conversion rule for code carrier data <n> within a block. If tool dialog data T<n> is assigned to the code-carrier data, then the first value in the code-carrier data in the block is assigned to the dialog data.

Example:

```
BItem1   1     BCD      C1, T2   ’ Relocate subtype to/from cutting edges
                                 ’ dialog datum 1 and tool dialog datum 2
                                 ’ (1st value of block is relevant for T2)
```

Group

Block<n> <repeat rule>

<n> := Consecutive number of block, ascending from 1 without gaps
<repeat rule> := * Item<n> | **CONTIGUOUS** BItem1

A block of data **BItem**<n> follows (up to the keyword **End_Block**<n>), which is stored on the code carrier according to the <repetition rule>.

Notice

In the case of **Block**<n> * **Item**<n>, Item <n> must be defined before Block<n>.

Example:

```
Block1 * Item6              ' Repeat Block1 according to the value of Item6
Block1 CONTIGUOUS BItem1    ' Read Block1 repeatedly until the count vari-
                            ' able BItem1 no longer returns a value increm-
                            ' ented by 1.
                            ' Write Block1 as many times as defined by the
                            ' value of the dialog variable assigned to BI-
                            ' tem1.
End_Block<n>
```

End_Block

End identifier for a data block defined with **Block**<n>

Code carrier data formats

The following code-carrier data formats are supported:
(comp. <code carrier data format> for **Item** / **BItem**)

Data format	Explanation
ASCII	ASCII character set
INT	16-bit integer (Intel format) • Value range –32768 <= INT <= +32767
BCD	• Binary-coded decimal number (if necessary, with sign and decimal point) • Non-relevant decades are preassigned the value 0, left-justified

Assignment between code carrier data and dialog data

The conversion rule for **Item<n>** or **BItem<n>** also contains the assignment to none/one/several dialog variables, if necessary with a conversion that is explained in detail in this section.

The general conversion specification for **Item<n>** or **BItem<n>** is:

(B)Item<n>**=<**line**>**

<n>	:=	Consecutive number of code carrier date, ascending without gaps
<line>	:=	<(max.) length in bytes> <code carrier data format> <dialog variable>

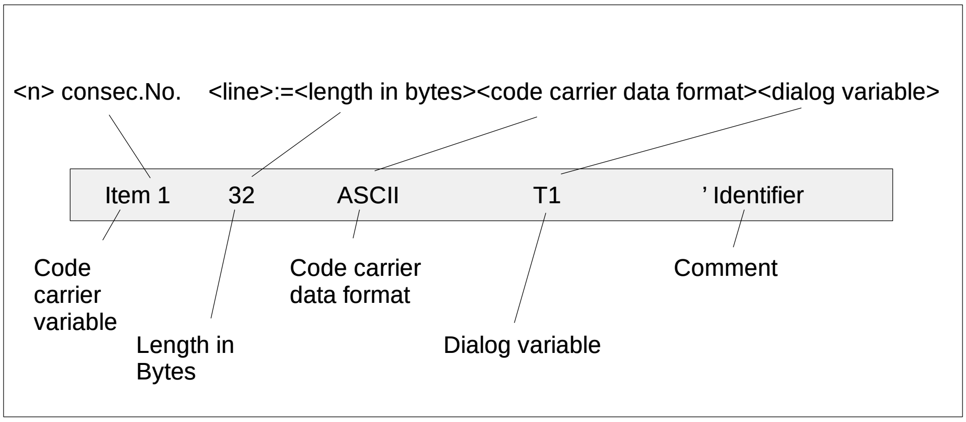


Bild 4-18 Conversion specification

Dialog variable

<**Dialog variable**> := <dvar1>[=(<uv>)] [, <dvar2>[=(<uv>)] [,<dvar3>
[&<dvar4>]=(<uv>)] [, <dvarN>[=(<uv>)]

<dvar> := T<index> | C<index> | –
T = tool data,
C = cutting edge data,
index = index within tool/cutting edges dialog data
– = no assignment to a dialog variable

<dvar1>&<dvar2>=<uv> : Conversion specification applies for <dvar1> and <dvar2>

uv := <arithm. Op1> [<arithm. Op2>] .. [<arithm. OpN>]
arithm. Op := +<const> | –<const> | *<const> | /<const>

Example:

```
T2=(*10), T3=(/100 + 10)
```

Or

uv := <replacement1> [<replacement2>] .. [<replacementN>]
replacement := <const1> [, <const2>] ..
[, <constN>] **^** <constM>
or
<const1..const2> ^ <const3>
const1 = lower limit value,
const2 = upper limit value

Notice

When converting the dialog variable to the code carrier variable on writing, if there are several left operands, the right operand is converted in the first left operands!

Example:

```
T2=(20..29 ^ 120 40,50 ^ 130)
```

The code carrier variable with the value 25 is converted to dialog variable T2 with the value 120 (read). Dialog variable T2 with value 120 is converted to code carrier variable with the value 20 (write).

Or

uv := <Tetn>
Tetn := nth tetrad in byte sequence
Byte1, = Tet1 and Tet2
Byte2, = Tet3 and Tet4

Allocation of the tetrads of code carrier variables (in BCD format) to dialog variables.

Example:

```
T5=(Tet1), T6=(Tet2), T7=(Tet3), T8=(Tet4)
```

If the code carrier has the value 0x1234 for example, dialog variable T5 is assigned the value 1, dialog variable T8 the value 4.

Or

uv := <compare>
compare := **<** <const> [INVSIGN] | **<=** <const> | **=** <const> | **>** <const> **>=** <const>

Assignment of code carrier variable to a dialog variable according to the comparison result

INVSIGN

- Reading: Invert leading sign of the dialog variable,
- Writing: Invert leading sign of the code carrier variable

Example:

```
C1=(<0 INVSIGN) , C2=(>=0)
```

Read:

A negative code carrier variable value corresponds to dialog variable C1, a positive value to dialog variable C2; dialog variable C1 is converted to a positive value.

Write:

Dialog variable C1 is multiplied by (–1). If the value is less than 0, then the code carrier variable is given the value from C1, otherwise the value from C2.

Notice

Conversion specifications are only evaluated for dialog variables of data type “integer”.

Conversion example / structure of a data string

Tabelle 4-2 Conversion file: wkonvert.txt

Code carrier variable	Length in Bytes	Format	Dialog variable
Item1	10	ASCII	T1 Identifier, $TC_TP2
Item2	2	BCD	T2 Duplo, $TC_TP1
Item3	2	BCD	T4 = (Tet1), T5 = (Tet2), T6 = (Tet3), T7 = (Tet4) Tool size: left, right, top, bottom, $TC_TP3, 4, 5, 6
Item4	10	ASCII	T8 location type, text for $TC_TP7

Tabelle 4-2 Conversion file: wkonvert.txt

Code carrier variable	Length in Bytes	Format	Dialog variable
Item5	1	BCD	T3 number of cutting edges, $P_TOOLND[tnr], tnr = tool number
Item6	4	BCD	A1 Tool OEM1, $TC_TPC1
Item7	4	BCD	A2 Tool OEM2, $TC_TPC2
Item8	2	BCD	C1 Subtype, type, $TC_DP1
Item9	4	BCD	C5 Geometry length1, $TC_DP3
Item10	4	BCD	C10 Geometry radius1, $TC_DP6
Item11	4	BCD	C14 Wear length 1, $TC_DP12

This conversion file can be used to read in the following data string/generate this string when writing:

626F687265725F312020000111116E6F726D616C2020202001D00010E3D000000
50205B00002E3B0000003B000E4562F2F

If working with SINTDC, this string corresponds to the interface between HMI Advanced and SINTDC.

Dividing this string into individual values

String	Value	Entry wkonvert.txt	Data types for dialog variables
626F687265725F312020	“Drill_1”	10 ASCII T1	T1 String identifier, $TC_TP2
0001	1	2 BCD T2	T2 Integer duplo, $TC_TP1
1111	1,1,1,1	2 BCD T4=(Tet1), T5=(Tet2), T6=(Tet3), T7=(Tet4))	T4 Integer tool size: Left T5 Integer tool size: Right T6 Integer tool size: top T7 Integer tool size: bottom
6E6F726D616C20202020	“Normal”	10 ASCII T8	T8 String location type, text for $TC_TP7
01	1	1 BCD T3	T3 Integer no. cutting edges, $P_TOOLND[tnr] tnr=tool number
D00010E3	–10.300	4 BCD A1	A1 Double tool OEM1, TC_TPC1
D0000005	–5	4 BCD A2	A2 Double tool OEM2, TC_TPC2
0205	205	2 BCD C1	C1 Integer subtype, $TC_DP1
B00002E3	2.3	4 BCD C5	C5 Double geometry length1, $TC_DP
B0000003	3	4 BCD C10	C10 Double geometry radius1, $TC_DP6

String	Value	Entry wkonvert.txt	Data types for dialog variables
B000E456	0.456	4 BCD C14	C14 Double wear length1, $TC_DP12
2F2F	End identifier (according to TIS_EOT=0x2F2F, mmc.ini)		

Example of a description file

Name of the description/conversion file

The file name must be entered in ...\user\mmc.ini at WToolIdSysKonv = **wkonvert.txt.**

The name of the file is e.g. **wkonvert.txt**

Code carrier variable	Length (bytes)	Data format	Interactive Variable	Comment
Item1	32	ASCII	T1	’ Identifier
Item2	3	BCD	T2	’ Duplo
Item3	2	BCD	T4=(Tet1),T5=(Tet2),T6=(Tet3), T7=(Tet4)	
’Tool size Left, right, top, bottom				
Item4	32	ASCII	T8	’ Location type
Item5	1	BCD	T9	’ Status
Item6	1	BCD	T3	’ No. Tool noses
Item7	1	BCD	T10	’ Type of tool monitoring
Item8	1	BCD	T11	’ Type of tool search
’ User tool data				
Item9	4	BCD	A1	’ Tool OEM1
Item10	4	BCD	A2	’ Tool OEM2
’Cutting edge data Block1 * Item6				
Bitem1	2	BCD	C1	’ Subtype, type
Bitem2	1	BCD	C4	’ Cutting edge position
’ Tool length compensation				
Bitem3	4	BCD	C5	’ Length 1
BItem4	4	BCD	C6	’ Length 2
BItem5	4	BCD	C7	’ Length 3
’Tool radius compensation				

Code carrier variable	Length (bytes)	Data format	Interactive Variable	Comment
BItem6	4	BCD	C8	’ Length 1
BItem7	4	BCD	C9	’ Length 2
BItem8	4	BCD	C10	’ Radius 1
BItem9	4	BCD	C11	’ Radius 2
BItem10	4	BCD	C12	’ Angle 1
BItem11	4	BCD	C13	’ Angle 2
’ Wear length compensation				
BItem12	4	BCD	C14	’ Length 1
BItem13	4	BCD	C15	’ Length 2
BItem14	4	BCD	C16	’ Length3
’Wear radius compensation				
BItem15	4	BCD	C17	’ Length 1
BItem16	4	BCD	C18	’ Length 2
BItem17	4	BCD	C19	’ Radius 1
BItem18	4	BCD	C20	’ Radius 2
BItem19	4	BCD	C21	’ Angle 1
BItem20	4	BCD	C22	’ Angle 2
’Basic dimension length compensation				
BItem21	4	BCD	C23	’ Basic length 1
BItem22	4	BCD	C24	’ Basic length 2
BItem23	4	BCD	C25	’ Basic length 3
BItem24	4	BCD	C26	’ Tool clearance angle
BItem25	1	BCD	C27	’ Reverse insert
BItem26	2	BCD	C29	’ Downtime in minutes
BItem27	2	BCD	C30	’ Prewarning limit for tool life
BItem28	2	BCD	C31	’ Workpiece count
BItem29	2	BCD	C32	’ Prewarning limit for tool life
’ User cutting edge data				
BItem30	4	BCD	U1	’ Cutting edge OEM1
BItem31	4	BCD	U2	’ Cutting edge OEM2
’User monitoring data				
BItem32	4	BCD	S1	’ Monitoring OEM1
BItem10	4	BCD	S2	’ Monitoring OEM2

4.6 Start-up of operator panel OP030

The tool management of the OP030 supports the loading/unloading detection of the OEM application SINTDI.
The full scope of functions is available with no restrictions in NCK version 3.7 and later.

Display machine data

MM_TM_SINTDI
The default setting 0 means that the SINTDI identifier function is deactivated.
A numerical value higher than 0 identifiers the parameter from which the load/unload identifiers are read or written.

MM_TM_NUM_MAG

Default setting = 0
A numerical value > 0 identifiers the magazine to be displayed first; if an illegal value is set, the first real magazine to be found is displayed.

MM_TM_UNLOAD_AND_DELETE
The default setting 0 means that tools are not erased from the TO memory when they are deleted, but only from the tool list.

MM_TM_LOAD_TOOL_NEW
The default setting 1 means that tools are marked immediately for loading in the tool list.

MM_TM_TOOL_STATE_DEF_VAL
The default setting is 2. Other values:

2	Enabled
4	Disabled
8	Measured
64	Fixed-location-coded

These values can also be combined (except for enable and disabled), i.e. 72 means measure and fixed-location-coded. Each change applies to newly created tools until the tool status is changed.

The display-device data can be changed in the file bd_op030.tea prior to the flash routine. A corresponding prompt is output during installation.

The differentiation is made during installation between system software and the screen kit. Screen kit here means the OEM variant or the development kit.

Notes

5 Programming

5.1 Overview of OPI and system variables

All the data required for the purpose of data management (e.g. to define a magazine or load a tool...) is stored in the NCK. This data can be read and written via part programs using system variables and via the PLC using FB 2 and FB 3. When configuring the machine, the user (machine manufacturer) must determine the most efficient method of reading and writing tool management data, i.e. in the PLC, the NC or in an ASUB.

Read and write access can generally be made to system variables.

When language commands are used, it may be necessary to program the “STOPRE” command.

The $TC variables do not generate a preprocessing stop.

Tool identifiers can consist of the following characters:
a...z
A...Z
0...9
+ – _ . ,
The names are case-sensitive, i.e. uppercase and lowercase characters are considered different characters.

Notice

Additional information on OPI variables can be found in the Help file for the NC variables selector.

Overview

Fig. 5-1 displays an overview of all cutting edge, tool and magazine data ($TC_...) when tool management is active.

Comment:

The sequence of system variables shown in the diagram corresponds to the OPI numbering sequence.

Notice

System variables are available for OEM Siemens data. However, they are not described here because they are not meaningful at present.

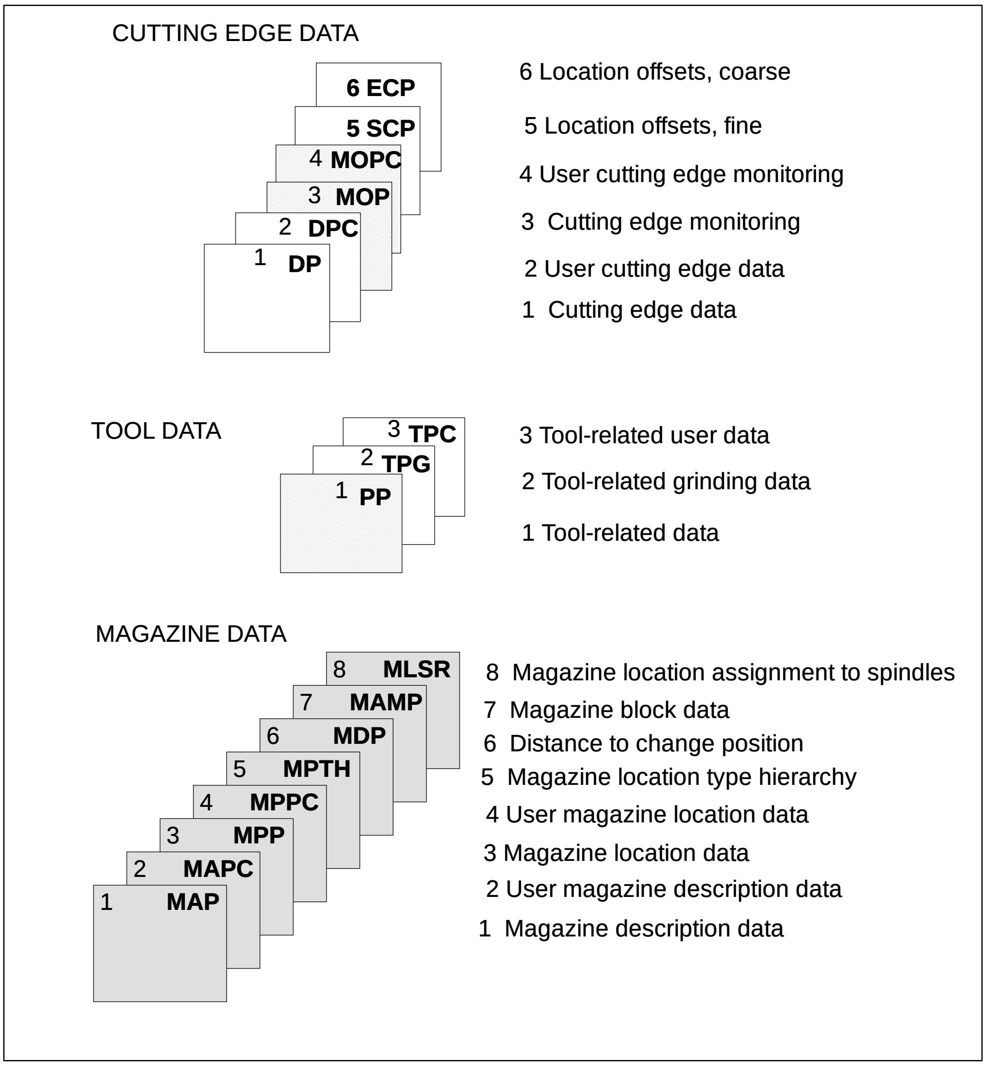


Bild 5-1 Overview of cutting edge, tool and magazine data

The identifiers (DP,...PP,...MAP,...) are taken from the NC language. They are part of the names of the system variables $TC_DP,...

Notice

The gray data fields are only available if tool management is active.

Shaded data fields are available without TOOLMAN function, but with monitoring function.

White data fields are available when the TOOLMAN function is not active.

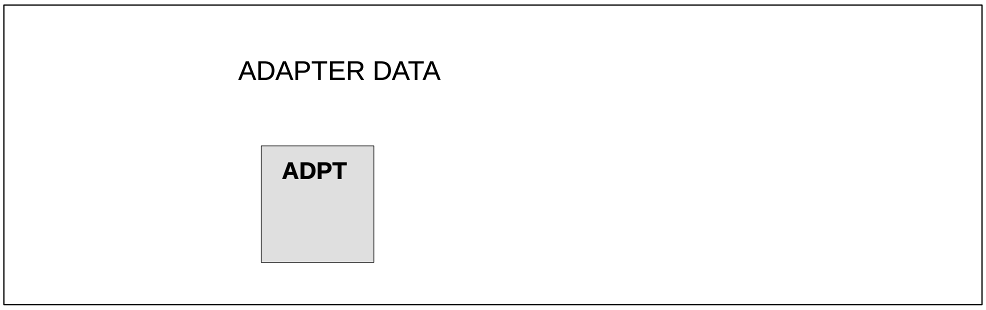


Bild 5-2 Adapter data

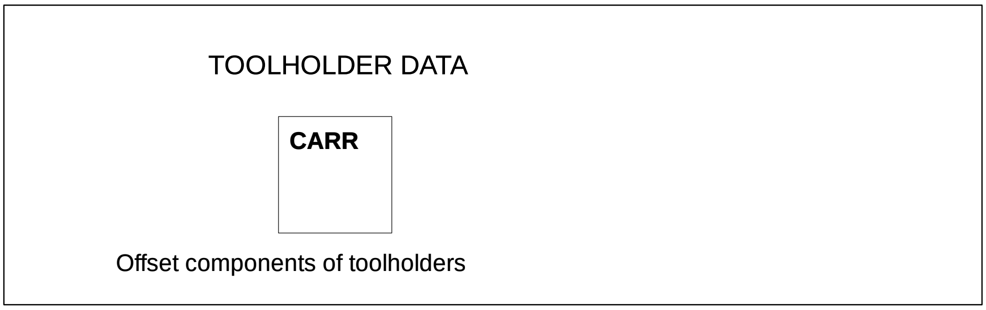


Bild 5-3 Toolholder data

5.2 Cutting edge data

Cutting edge data

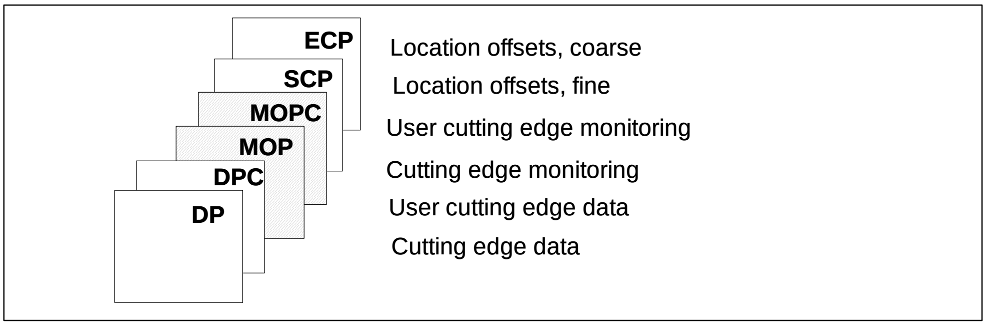


Bild 5-4 Overview of cutting edge data

This data exists for each cutting edge that is created (D1–D12). Tool management includes the geometry and user data as well as the optional monitoring data for the cutting edges.

If the cutting edges are created via HMI, the D number is counted up from 1. It is possible to program the D no. with gaps, e.g. D1, D3, D6 if cutting edges are set up using the NC program.

5.2.1 Cutting edge parameters

$TC_DPx[t,D]

Cutting edge parameters for geometry, technology and tool type.
Depending on the tool type, up to 25 cutting edge parameters can be programmed for each tool cutting edge.

References: /FB1/ W1, Tool Compensation

x = Parameter 1...25
t = T number 1...32000
d = Cutting edge number 1...12
D = D number

The maximum value of x is stored in OPI variable numCuttEdgeParams in block Y.

OPI block TO

Calculation of line: (d–1)*numCuttEdgeParams+parameter no.

Calculation of column: T number

Tool offset parameters (system variables)					
NCK identi-fier	Type	Description	OPI variable	Type	Preas-sign-ment
$TC_DP1	INT	Tool type	edgeData	REAL	9999
$TC_DP2	Double	Cutting edge position*	edgeData	REAL	0
$TC_DP3	Double	Geometry length 1	edgeData	REAL	0
$TC_DP4	Double	Geometry length 2	edgeData	REAL	0
$TC_DP5	Double	Geometry length 3	edgeData	REAL	0
$TC_DP6	Double	Geometry radius	edgeData	REAL	0
$TC_DP7	Double	Geometry – corner radius (tool type 700; slotting saw)	edgeData	REAL	0
$TC_DP8	Double	Geometry length 4 (tool type 700; slotting saw)*	edgeData	REAL	0
$TC_DP9	Double	Geometry length 5*	edgeData	REAL	0
$TC_DP10	Double	Geometry – angle 1*	edgeData	REAL	0
$TC_DP11	Double	Geometry – angle 2 for conical milling tools*	edgeData	REAL	0
$TC_DP12	Double	Wear – length 1	edgeData	REAL	0
$TC_DP13	Double	Wear – length 2	edgeData	REAL	0
$TC_DP14	Double	Wear – length 3	edgeData	REAL	0
$TC_DP15	Double	Wear radius	edgeData	REAL	0
$TC_DP16	Double	Wear – slot width b/rounding radius	edgeData	REAL	0
$TC_DP17	Double	Wear – projection k	edgeData	REAL	0
$TC_DP18	Double	Wear – length 5	edgeData	REAL	0
$TC_DP19	Double	Wear – angle 1	edgeData	REAL	0
$TC_DP20	Double	Wear – angle 2 for conical milling tools	edgeData	REAL	0
$TC_DP21	Double	Adapter length 1	edgeData	REAL	0
$TC_DP22	Double	Adapter length 2	edgeData	REAL	0
$TC_DP23	Double	Adapter length 3	edgeData	REAL	0
$TC_DP24	Double	Clearance angle	edgeData	REAL	0

NCK identi-fier	Type	Description	OPI variable	Type	Preas-sign-ment
$TC_DP25	Double	1. The cutting rate value is stored here for ManualTurn ": A bit-coded value for various states of tools of type 1xx and 2xx is stored here for ShopMill.	edgeData	REAL	0
$TC_DPCE [t,d]	INT	System variable of an offset data record with T=t and D=d containing cutting edge number CE. (unique D no. or user-assignment of D nos. for cutting edge numbers). Value range for permissible cutting edge numbers: 1 up to value of MD 18106.	–		0
$TC_DPH [t,d]	INT	H parameter (Y / extraCuttEdgeParams, Bit0=1	–		0
$TC_DPV	Double	Tool cutting edge orientation	–		–
$TC_DPV3		L1 component of the tool cutting edge orientation	–		
$TC_DPV4		L2 component of the tool cutting edge orientation	–		
$TC_DPV5		L3 component of the tool cutting edge orientation	–		

* The meaning of this data is different depending on the tool type.

$TC_DP11

$TC_DP11 contains the identification for the main direction of machining as is defined and required by the Siemens cycle 950. $TC_DP11 assumes an intermediate position between tool OEM parameter and NCK system variable.

$TC_DP11 is a tool OEM parameter in so far as NCK does not evaluate the contents of the value.

$TC_DP11 is a tool system variable in so far as when accessing via $P_ADT[n] – n=11, NCK is subject to the special values 1, 2, 3, 4 of the tool adapter transformation if TMMG and the subfunction “Tool adapter” are active. This system parameter property is also used with the analog OPI block TOT.

5.2.2 User cutting edge data

$TC_DPCx[t,D]

User-related cutting edge data
Up to 10 additional cutting edge parameters can be programmed for each cutting edge. Set with MD 18096: MM_NUM_CC_TOA_PARAM and enable with MD 18080 MM_TOOL_MANAGEMENT_MASK (set bit 2=1)

x = Parameter 1...10
t = T number 1...32000
d = Cutting edge number 1...12
D = D number

OPI block TUE/TUO

Calculation of line: (d–1)*numCuttEdgeParams_tu+parameter no.

Calculation of column: T number

User-related cutting edge data					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preassignment**
$TC_DPC1	Double	CC_Cutting edge parameter 1	edgeData	REAL	0
...	Double	...	edgeData	REAL	0
$TC_DPC10	Double	CC_Cutting edge parameter 10	edgeData	REAL	0

Notice

The data is displayed in the tool management. Here you could store “Max. cutting rate”, for example, which is then evaluated in the part program.

5.2.3 Edge-related tool monitoring

$TC_MOPx[t,D]

Tool cutting edges are monitored according to tool life, workpiece count and/or wear.

x = Parameter 1...15
t = T_number 1...32000
d = Cutting edge number 1...12
D = D number

The maximum value of x is stored in OPI variable numCuttEdgeParams in block Y.

OPI block TS

Calculation of line: (d–1)*numCuttEdgeParams_ts+parameter no.

Calculation of column: T number

Tool management monitoring data					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preassignment**
$TC_MOP1	Double	Prewarning limit for tool life in min.	data	REAL	0
$TC_MOP2	Double	Residual tool life in minutes	data	REAL	0
$TC_MOP3	INT	Pre-warning limit for count	data	REAL	0
$TC_MOP4	INT	Residual unit quantity	data	REAL	0
$TC_MOP11	Double	Setpoint for tool life	data	REAL	0
$TC_MOP13	INT	Setpoint for workpiece count	data	REAL	0
$TC_MOP5	Double	Wear prewarning limit – or local offset fine prewarning limit	data	REAL	0
$TC_MOP6	Double	Actual wear or actual value for local offset fine	data	REAL	0
$TC_MOP15	Double	Wear setpoint or setpoint for local offset fine	data	REAL	0

5.2.4 User cutting-edge monitoring

$TC_MOPCx[t,D]

Tool monitoring user data (edge-specific)

Up to 10 additional tool monitoring parameters can be programmed for each cutting edge. Set with MD 18098: MM_NUM_CC_MON_PARAM and enable with MD 18080 MM_TOOL_MANAGEMENT_MASK (set bit 2)

x = Parameter 1....10
t = T_number 1...32000
d = Cutting edge number 1...12
D = D number

OPI block TUS

Calculation of line: (d–1)*numCuttEdgeParams_tus+parameter no.

Calculation of column: T number

Tool monitoring user data (edge-specific)					
NCK identifier	**Type**	**Description**	**OPI variables**	**Type**	**Preassignment**
$TC_MOPC1	Int	CC monitoring parameters	userdata	REAL	0
. . .	Int	. . .	userdata	REAL	0
$TC_MOPC10	Int	CC monitoring parameters	userdata	REAL	0

5.2.5 Location offsets, fine (additive offsets)

$TC_SCPx[t,D]

Location offsets fine (the term “additive offsets” is also frequently used) comprise all the magnitudes of error which contribute to the total deviation between the actual workpiece and the specified dimensions. The parameters for the location offsets refer to the geometrical data of a cutting edge. DL stands for D Location, whereby Location refers to where the cutting edge is used.

x = Parameter for DL=1...DL=6
t = T number 1...32000
d = Cutting edge number 1...12
D = D number

OPI block TOS, TOST

Calculation of line: (d–1)*(maxnumEdgeSC*numParams_SC)+((EdgeSC–1)*numParams_SC)+parameter no.

Calculation of column: T number

Local offsets				
Name	**Type**	**Description**	**OPI variable**	**Type**
$TC_SCPx				
x = 13–21	Double	Can be activated with DL=1	edgeSCData	REAL
x = 23–31	Double	Can be activated with DL=2	edgeSCData	REAL
x = 33–41	Double	Can be activated with DL=3	edgeSCData	REAL
x = 43–51	Double	Can be activated with DL=4	edgeSCData	REAL
x = 53–61	Double	Can be activated with DL=5	edgeSCData	REAL
x = 63–71	Double	Can be activated with DL=6	edgeSCData	REAL
		Transformed location offsets fine, block TOST	edgeSCData	REAL

5.2.6 Location offsets, coarse (setup offsets)

$TC_ECPx[t,D]

The coarse location offsets (setup offsets) can be set by the machine setter before the machining operation (see also $TC_SCP).

x = Parameter for DL=1...DL=6
t = T number 1...32000
d = Cutting edge number 1...12
D = D number

OPI block TOE, TOET

Calculation of line: (d–1)*(maxnumEdge_SC*numParams_SC)+((EdgeSC–1)*numParams_SC)+parameter no.

Calculation of column: T number

Setup offsets				
Name	**Type**	**Description**	**OPI variable**	**Type**
$TC_ECPx			edgeECData	REAL
x = 13–21	Double	Can be activated with DL=1	edgeECData	REAL
x = 23–31	Double	Can be activated with DL=2	edgeECData	REAL
x = 33–41	Double	Can be activated with DL=3	edgeECData	REAL
x = 43–51	Double	Can be activated with DL=4	edgeECData	REAL
x = 53–61	Double	Can be activated with DL=5	edgeECData	REAL
x = 63–71	Double	Can be activated with DL=6	edgeECData	REAL
		Transformed setup offsets, block TOET	edgeECData	REAL

5.3 Tool data

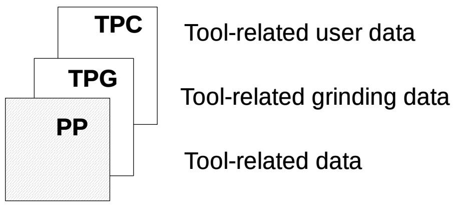


Bild 5-5 Overview of tool data

5.3.1 Tool-related data

$TC_TPx[t]

General tool data
This data describes the tool in the magazine.
Programming of general tool data with tool management.

x: = Parameter 1...11
t: = T number 1...32000

OPI block TD

Calculation of line: T number

Calculation of column: not applicable

Tool-related data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preas-sign-ment**
$TC_TP2	String	Tool identifier	toolIdent	String	“T No.”
$TC_TP1	INT	Duplo number	duploNo	WORD	T No.
$TC_TP3	INT	Size on left	toolsize_left	WORD	1
$TC_TP4	INT	Size on right	toolsize_right	WORD	1
$TC_TP5	INT	Size above	toolsize_upper	WORD	1
$TC_TP6	INT	Size below	toolsize_down	WORD	1
$TC_TP7	INT	Magazine location type	toolplace_spec	WORD	9999

Tool-related data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preas-sign-ment**
$TC_TP8	INT	Status Value 0 not enabled Bit 0 active tool (A) Bit 1 enabled (F) Bit 2 disabled (G) Bit 3 measure (M) Bit 4 prewarning limit reached (V) Bit 5 tool being changed (W) Bit 6 fixed-location coded (P) Bit 7 tool was in use (E) Bit 8 identifier for tools in buffer Bit 9=1 ingore disabled state Bit 9=0 do not ignore Bit 10 to be unloaded Bit 11 to be loaded Bit 12 master tool The letters in parentheses correspond to the ID on the HMI screen.	toolState	WORD	0=not enabled
$TC_TP9	INT	Tool monitoring method Value 0 no tool monitoring Bit 0 tool life Bit 1 workpiece count Bit 2 wear monitoring active Bit 3 wear monitoring, location offset fine active	toolMon	WORD	0
$TC_TP10	INT	Replacement-change strategy	toolSearch	WORD	0
$TC_TP11	INT	Tool information: Allows tool groups to be divided into subgroups. Tool selection only with tools of the subgroup (from SW 6)	toolInfo	Integer	0
$A_TOOLMN	INT	Magazine assignment tool	toolInMag	WORD	
$A_TOOLMLN	INT	Location assignment tool	toolInPlace	WORD	
$P_TOOLND	INT	Number of cutting edges	numCuttEdges	WORD	
–		Adapter no. Assignment	adaptNo	WORD	
$A_MYMN	INT	Owner magazine for tool	toolMyMag from SW6	WORD	
$A_MYMLN	INT	Owner magazine location for tool	toolMyPlace from SW6	WORD	

$TC_TP1 and $TC_TP2

Like the T no. is sufficient to identify a tool, a tool is equally unique in terms of its duplo number and its tool name (identifier).

TO units may therefore only contain names that have different duplo numbers. The write operations of $TC_TP1 and $TC_TP2 are checked for the above and rejected if collisions are found.

$TC_TP3 to TP6

Size in terms of half locations:
Size 1 means that the tool exactly completely occupies its own magazine location.
The maximum programmable size is 7.
There are rules governing how tool sizes are specified.

$TC_TP7

The magazine location type cannot be changed if the tool is in a magazine location.

$TC_TP8

The tool status is described with system variable $TC_TP8. This variable is bit-coded. In other words, a particular state of the tool is assigned to each bit of this data.

The status of a tool must be **bit 1** so that it can be loaded within the scope of a programmed tool change for processing on the toolholder.

During tool selection, the status of a tool that is loaded onto the toolholder (spindle, ...) is set by the NCK to **bit 0** (“active”).

A tool cannot be loaded if its status is **bit 2**. The status is set automatically by the tool monitoring function, when the monitoring value of at least one cutting edge reaches the limit value. The status bit 2=4 of the tool on the toolholder can or will be ignored when generating the INIT blocks (see MD 20110 and 20112). The PLC also has the option to make NCK ignore the status during tool selection.

The status **bit 4** is mainly for information purposes. With this status the tool can still be loaded.

The status **bit 7** (“was in use”) is set by the NCK if the tool is removed from a magazine location of the type spindle or toolholder.

The tool status **bit 5** (=“W”= is being change) is always reset by the software during buffered booting. A tool receives/loses this status within the scope of a programmed tool change.

The following applies: all tool (new and old) involved in the tool change are given the status bit 5=32 by the tool selection. The status is reset again by the end acknowledgement for each tool command.
The following applies in particular:

The end acknowledgement of the PLC command 2 (programming the T address with $MC_TOOL_CHANGE_MODE=1), resets the status “W” of the old tool.

The end acknowledgement of the PLC commands 3, 4, 5
(programming M06 in a block with $MC_TOOL_CHANGE_MODE=1, T, M06 in a block with $MC_TOOL_CHANGE_MODE=1
T address with $MC_TOOL_CHANGE_MODE=0)
resets the status bit 5=32 of the old tool and the new tool.

Tools that are in the buffer can be used for a new programmed tool command if the tool status bit 5=32.

Tools that are in the real magazine and have this status can be used as a dependency of the bit 21 of the MD 20310 or cannot be used for a different tool-change comment.

The status bit 5=32 is generally not considered for a tool selection within the scope of a block search or when init is generated.

For a reset, the status is reset for those tools that are involved in a tool change at that point in time.

The status bit 5=32 is not evaluated when a manual tool is selected.

The tool status **bit 8** ensures that during the next tool change, a tool that is at a buffer location and not intended for the next job in machining is returned to the real magazine. See also Subsection3.2.2.

Bit 9 ignores disabled state.
If this bit is set, the disabled state of this tool is ignored. This means the disabled tool can be used (depending on the search strategy).
This state acts independently of the PLC interface signal:
“Tool disable not effective” (DB21.DBx29.7).

Status bit 11 (to be loaded)

Bit 11 is set for tools that are not in a magazine and are to be loaded. The following definitions apply:

- The status is maintained beyond Power ON.
- It is included in the data back-up and rewritten when transferred back to the NCK.
- When assigning a tool to a real magazine the tool status is reset by the NCK (applies to locations of location type 1, i.e. not to internal magazines such as the loading magazine, buffer magazine, etc).

Bit	Value	Meaning
11	0	"Not to be loaded"
	1	**"To be loaded"**

Status bit 10 (to be unloaded)

This bit is set for tools that are located in a magazine and are to be unloaded. The following definitions apply:

- The status is maintained beyond Power ON.
- It is included in the data back-up and rewritten when transferred back to the NCK.
- Unloading a tool via an unloading location causes the NCK to reset the tool status.

Bit	Value	Meaning
10	0	"Not to be unloaded"
	1	**"To be unloaded"**

Status bit 12 (master tool)

Bit 12 is set for tools that are to be permanently assigned to a magazine. This status is only set to provide information and has no effect on the NCK (e.g. does not disable a location). The user defines via the unload program whether the tool can be unloaded.

Bit	Value	Meaning
12	0	"Not a master tool"
	1	**"Master tool"**

Notice

Take care when "manually" changing the tool status via the OPI during machining. This could undo any necessary internal changes in status by the NCK and result in incorrect machining.

$TC_TP9

If a monitoring type is activated for the tool with $TC_DP9, then the current monitoring parameters are evaluated and, if necessary, the tool status set to 'disabled' or 'prewarning limit reached'. An existing tool disable is not however lifted. Not even then when the monitoring function for this tool has been deactivated.

$TC_TP11

Tool subgroups

The system variable is bit-coded. Only bits 0...3 are evaluated. A tool group (the same identifier, different duplo no.) can be split into a maximum of 4 subgroups in this way. A tool can also be included in several subgroups.

If no bit is set, so $TC_TP11[x]=0, this means the same as "all bits set", i.e. the tool belongs to all the defined subgroups.

<u>Selection of the tool subgroup</u>

1. With the language command **$P_USEKT** (UseKindofTool)
   (only possible when <u>not</u> working with the setting T=location)
   During tool search, only tools that have one of these bits in system variable $TC_TP11, can be found. This means that it is possible to form so-called "Technology Groups", to differentiate between tools with the same identifier and specifically release them for machining.

   Example 1:
   $P_USEKT=4
   i.e. the only tools to be taken into account are those with bit 2 in $TC_TP11 or
   Example 2:
   $P_USEKT=9
   i.e. the only tools to be taken into account are those with bit 3 or 0 in $TC_TP11
2. By programming a tool
   with the function **T=location** $P_USEKT is set automatically at every tool change and in fact at the $TC_TP11 value of the loaded tool.

   Example: T3 M06
   The bit value of $TC_TP11 of T3 is now valid (is accepted in "USEKT").
   During the transition to a spare tool (and there only) the only tools to be taken into account are those with one of these bits set in system variable $TC_TP11.

5.3.2 Tool-related grinding data

$TC_TPGx[t]

Technology-specific grinding data
The default setting for grinding data is 0. Tools with **tool type 400 to 499** are always **grinding tools**, i.e. have these additional data which take up additional memory space. If a tool of type 400–499 is set to a value outside this range, then its loses its grinding-specific data – the associated memory is released again and can be used for other tools.

x: = Parameter 1....9
t: = T number 1...32000

OPI block TG

Calculation of line: T number

Calculation of column: not applicable

Tool-related grinding data				
Name	**Type**	**Description**	**OPI VAR**	**Type**
$TC_TPG 1	INT	Spindle number	spinNoDress	REAL
$TC_TPG 2	INT	Chain rule	conntectPar	REAL
$TC_TPG 3	Double	Minimum grinding wheel radius	minToolDia	REAL
$TC_TPG 4	Double	Minimum grinding wheel width	minToolWide	REAL
$TC_TPG 5	Double	Current grinding wheel width	actToolWide	REAL
$TC_TPG 6	Double	Max. speed	maxRotSpeed	REAL
$TC_TPG 7	Double	Maximum peripheral speed	maxTipSpeed	REAL
$TC_TPG 8	Double	Inclination angle of inclined wheel	inclAngle	REAL
$TC_TPG 9	INT	Parameter number for radius calculation	paramNrCCV	REAL

5.3.3 Tool-related user data

$TC_TPCx[t]

User-related tool data
An additional 10 tool-specific parameters can be set up per tool. Set with MD 18094: MM_CC_TDA_PARAM and enable with MD18080 MM_TOOL_MANAGEMENT_MASK (set bit 2)

x: = Parameter 1...10
t: = T number 1...32000

OPI block TU/TUD

Calculation of line: T number

Calculation of column: Parameter number

Tool-related OEM user data				
NCK identifier	**Type**	**Description**	**OPI VAR**	**Type**
$TC_TPC1	Double		data	REAL
. . .	Double		data	REAL
$TC_TPC10	Double		data	REAL

Notice

The data is displayed in the tool management. In addition, e.g. tool status information could also be stored here.

5.4 Magazine data

Magazine data

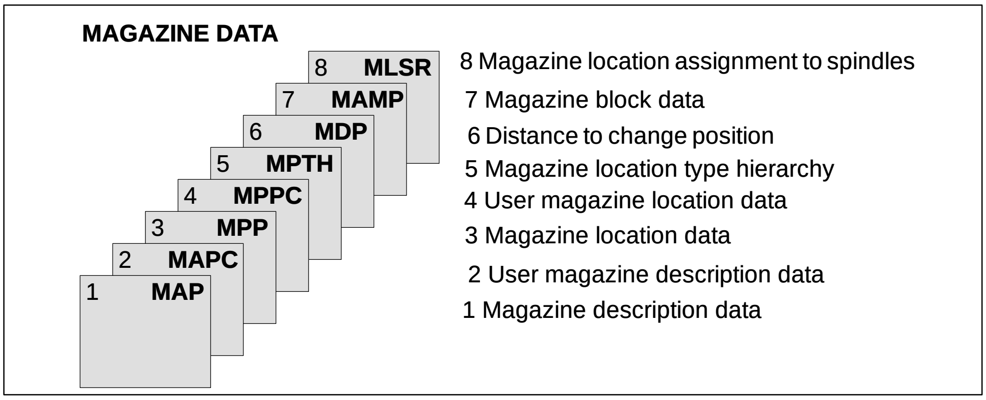


Bild 5-6 Overview of magazine data

5.4.1 Magazine description data

$TC_MAPx[n]

Magazine description data
This data identifies the real magazine

x: = Parameter 1...10
n: = Magazine number 1...30, 9998, 9999

OPI block TM

Calculation of line: Magazine number

Calculation of column: not applicable

Magazine description data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preas-sign-ment**
		Magazine number	magNo	WORD	0
$TC_MAP2	String	Identifier of the magazine	magIdent	String	“ ”

Magazine description data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preas-sign-ment**
$TC_MAP1	INT	Type of magazine **1 =** Chain **3** = Turret **5** = Box magazine **7** = Tool buffer magazine **9** = Loading station magazine	magKind	WORD	0
$TC_MAP3	INT	Status of magazine Bit 0 active magazine Bit 1 disabled Bit 2 Magazine in loading position Bit 3 Tool move is active Bit 4 Magazine/tool may be moved Released for loading Bit6 Magazine is fixed-location-coded, i.e. tools in this magazine are considered fixed-location-coded tools and treated accordingly Bit8 Edge location edge may be overlapped left Bit9 Edge location edge may be overlapped right Bit10 Edge location edge may be overlapped top Bit11 Edge location edge may be overlapped bottom	magState	WORD	2
$TC_MAP4 (currently not available)	INT	Chaining to next magazine Magazine type = 1, 3, 5. For background magazines only	magLink1	WORD	–1
$TC_MAP5 (currently not available)	INT	Chaining to preceding magazine Magazine type = 1, 3, 5. Reference (= number) to preceding magazine, backward chaining of background magazines	magLink2	WORD	–1
$TC_MAP6	INT	Number of tiers (box magazines only)	magDim	WORD	1
$TC_MAP7	INT	Number of columns	–	–	
–	–	Number of locations for the magazine, corresponds to TC_MAP6*TC_MAP7	magNrPlaces	WORD	0
$TC_MAP8	INT	Current magazine position relative to change position	magActPlace	WORD	0
–			magCmd	WORD	
–			magCmdState	WORD	
–			magCmdPar1	WORD	

Magazine description data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preassignment**
–			magCmdPar2	WORD	
$TC_MAP9	INT	Current wear grouping number	magWearCompoundNo	DINT	0
$TC_MAP10 (bit 0...7)	INT	Current tool search strategies of magazine (see $TC_MAMP2)	magToolSearchStrat	WORD	0
$TC_MAP10 (bit 8...15)	INT	Current empty location search strategy of magazine	magPlaceSearchStrat	WORD	0

$TC_MAP3

The magazine status **bit 3** (tool motion is active) is always reset when the software is booted with backup.

A magazine that has the status "Tool motion is active" cannot be deleted.

Empty locations are not sought in magazines with the "disabled" status. If a disabled magazine is explicitly defined for the empty location search the process is aborted with an error message.

A tool that is in a "disabled" magazine cannot be loaded into the spindle or the toolholder.

Overlapping edge locations (bit 8...11=1)
An oversized tool can overlap the edge locations that are marked with bits 8 to 11.

$TC_MAP8

The current magazine position $TC_MAP8 is refreshed by the NCK every time the magazine is moved.

When the magazine configuration has been loaded, variable $TC_MAP8 is assigned the value zero. The position value is the number of the magazine location that is located at the zero position of the magazine. As a maximum, the magazine position can have the number of magazine locations in the magazine. Larger or negative values are rejected.

$TC_MAP10

Magazine-specific tool search

The bit settings correspond precisely to the system variables $TC_MAMP2. For buffer magazines, the default setting “0” always applies.
Exception:
From SW 6.3.23 and 5.3.35, a 1:1 swap can be set for internal magazines.

5.4.2 Magazine user data

$TC_MAPCx[n]

Magazine user data

Up to 10 user data can be additionally created for each magazine. Setting in MD 18090 : MM_NUM_CC_MAGAZINE_PARAM and enable with MD18080: MM_TOOL_MANAGEMENT_MASK (set bit 2)

x: = Parameter 1...10
n: = Magazine number 1...30

OPI block TUM

Calculation of line: Parameter number

Calculation of column: Magazine number

Magazine description data OEM user				
Name	**Type**	**Description**	**OPI VAR**	**Type**
$TC_MAPC1			userData	DINT
...			userData	DINT
$TC_MAPC10			userData	DINT

5.4.3 Magazine location data

$TC_MPPx[n,m]

Magazine location data
The following data describes the magazine location.

x: = Parameter 1..7
n: = Physical magazine number 1..30, 9998, 9999
m: = Physical location number 1...32000

The maximum value of x is stored in OPI variable numMagPlaceParams in block Y.

OPI block TP

Calculation of line: (magazinLocNo–1)*numMagPlaceParams+parameter no.

Calculation of column: Magazine number

Magazine location data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preassignment**
$TC_MPP1	INT	Location type 1 = Magazine location 2 = Spindle, toolholder 3 = Gripper 4 = Loader 5 = Transfer location 6 = Loading station 7 = Loading point	placeData	WORD	0
$TC_MPP2	INT	Location type > 0: Location type for virtual location = 0: Every tool fits in this location 9999: Not defined	placeData	WORD	9999
$TC_MPP3	BOOL	Consider adjacent location ON/OFF	placeData	WORD	FALSE

Magazine location data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preassignment**
$TC_MPP4	INT	Location state Bit 0 disabled (A) Bit 1 free to hold a tool (occupied) (F) Bit 2 reserved for tool from buffer (G) Bit 3 reserved for new tool to be loaded (M) Bit 4 occupied in left half location (V) Bit 5 occupied in right half location (W) Bit 6 occupied in top half location (P) Bit 7 occupied in bottom half location (E) Bit 8 left half location reserved Bit 9 right half location reserved Bit 10 top half location reserved Bit 11 bottom half location reserved Bit 12 wear group disabled Bit 13 overlapping permitted. Tools can overlap disabled magazine locations. Only possible if consider adjacent location is active (see also MD 17520).	placeData	WORD	1
–		Reference phys. magazine (top right)	placeData	WORD	0
$TC_MPP5	INT	Location type index (location type numbering) or wear group number	placeData	WORD	0
$TC_MPP6	INT	T no. of tool on this location	placeData	WORD	0
$TC_MPP7	INT	No. of adapter in mag. location	placeData	WORD	0

Writing magazine location data

Points to be noted about writing magazine location data:
The first time one of the $TC_MPP... parameters is written all the magazine locations defined by magazine parameters are created with their default values (the memory for the locations is therefore “used up”), i.e. the magazine must have been defined by this time ($TC_MAP... parameter).

$TC_MPP1 (location type)

Only magazine locations of the type “Magazine location” ($TC_MPP1 = 1) may be defined at magazines that are not of the type “internal” ($TC_MAP1 = 7 or = 9).

Location types:

1 = Magazine location
Only locations of type “1” can be defined at real magazines.

2 = Spindle/toolholder

3 = Gripper

4 = Loader

5 = Transfer location
The distinction gripper/loader/transfer location is intended for future HMI applications. The NCK makes no distinction here.

6 = Loading station
After the tool moves to this location, the tool stays there.
It can only be removed by explicit operation (unloading).

7 = Loading point
If a tool is moved from the magazine or buffer to this location, after the PLC acknowledgement of this motion command, the tool is automatically removed from this location.

Please note when writing the location status and number of the tool in this location that the following dependencies on $TC_MPP2 to $TC_MPP4 apply; these are checked during the write operation:

- If the location already contains a tool, the location type to be written must be checked against the tool location type.
- The status “not occupied” may only be written when none of the “assigned” states is set and there is no tool at the location.
- The “Disabled” state can be set irrespective of the other states.
- If there is no tool here then the state “not occupied” is automatically active; i.e., the state “not free” cannot be set by the NC program or PLC, HMI.
- “Occupied” states can only be set by the NCK within the scope of the adjacent location consideration; i.e. these states are ignored during writing by the NC program or PLC, HMI.
- The state “Reserved for tool from buffer” is set when a tool is removed by the NCK from the real magazine during a tool change. This location is then not designated as “Free” for tools other than the tool removed.

- The states “Reserved for tool from buffer” and “reserved for new tool to be loaded” **of a location** are automatically reset when a tool is placed in this location.
- The states “reserved for tool from buffer” and “reserved for new tool to be loaded” of a **real magazine location** are automatically reset when a tool from this location is placed at a location in the loading/unloading magazine.
- The state “Reserved for tool from buffer” is reset during an empty location search if the tool for which the empty location is being sought is assigned a magazine location other than its previous real magazine location. The newly found empty location is assigned the state “Reserved for tool from buffer” and becomes the new owner of the tool being sought.

The magazine location state “Reserved for tool to be loaded” is always reset when the control system is restarted. If “Consider adjacent location” is active, reservations of adjacent locations are also considered.

It is only when wanting to make the magazine definition directly at the NC program level that the user has to deal with these rules. Data back-up is such that the rules are observed when data is imported to the NCK.

$TC_MPP5 (location type index)

This data contains the spindle number for magazine locations of type “spindle” ($TC_MPP1) and is thus made known to the tool management.

The value cannot be changed for location type = 1 ($TC_MPP1; i.e. for all locations of internal magazines) if there is a tool at the location.

$TC_MPP6 (T no.)

- Tools can only be placed in magazine locations when both the tool and the magazine, plus its magazine locations, have been defined.

The tool may occupy only one magazine location!

Procedure:
It first attempts to find the tool associated with the T no.

- If it is already defined, then an attempt is made – subject to appropriate check procedure – to add it to the magazine location.
- If it is not yet defined, then an error has occurred.

Tests:

- The type of the tool to be placed must match the type of the location. If the type has not been set explicitly at the time of writing (default = 9999 = “Not defined”), then the tool is not placed.
- The state of the location must be “Free” and must not be “Disabled”.

- If the value T no.=0 is programmed, then this means that the present tool will be removed from the magazine location.
  NOTICE ! $TC_MPP6 = 0 also changes the state of the location: a tool can only be placed in a magazine location if the location does not already contain a tool. The old tool might first have to be removed with $TC_MPP6 = 0.

Notice

Because of this dependency of the individual data, it is mandatory for the T number of the tool to be written as the last data in a magazine configuration. If you do not adhere to this sequence, default values might be set which may result in unwanted data.

5.4.4 Magazine location user data

$TC_MPPCx[n,m]

Magazine location user data

Up to 10 user data can be additionally created for each magazine. Setting for number of parameters in MD 18092 : MM_NUM_CC_MAGLOC_PARAM and enable with MD18080 MM_TOOL_MANAGEMENT_MASK (set bit 2)

x: = Parameter 1...10
n: = Magazine number 1...30
m: = Magazine location number 1...32000

OPI block TUP

Calculation of line: (m–1)*numMagLocParams_u+Parameternr.

Calculation of column: Magazine number

OEM user magazine location data					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preas-sign-ment**
$TC_MPPC1	INT		userplaceData	DINT	0
. . .	INT		userplaceData	DINT	0
$TC_MPPC10	INT		userplaceData	DINT	0

5.4.5 Magazine location type hierarchy

$TC_MPTH[n,m]

Magazine location type hierarchy
The location types can be organized in a hierarchy by programming these system variables.

n: = Index of hierarchy, from 0...7
m: = Index within hierarchy n, location type 0...7

Magazine location types, refer also to $TC_TP7 (Subsection 5.3.1) and $TC_MPP2 (Subsection 5.4.3).

OPI block TT

Calculation of line: Number of location type+1

Calculation of column: Number of location hierarchy+1

Magazine data: Magazine location type hierarchy					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preassignment**
$TC_MPTH[n,m]	INT	Location type hierarchy n: Hierarchy 0–7 m: Location type 0–7	placeType	WORD	9999

If a tool is to be loaded into the magazine, then the location type determines the availability of locations, i.e. $TC_TP7 and $TC_MPP2 must be defined.
If the location type of the tool is part of the location type hierarchy, then the location assignment is carried out in accordance with this hierarchy.
Several such hierarchies can be set up in one TO-area unit, but a location type can only be entered in one hierarchy.

Example

A chain magazine is to be split into six location types and the following hierarchy defined (the magazine no. is to be “1”, the numbers of the location types are selected at random).

Location type_124< Location type_3< Location type_15< Location type_1080< Location type_5< Location type_18

Definitions:

Magazine: $TC_MPP2[magazine no., location]
$TC_MPP2[1,1...6] = 124
$TC_MPP2[1,7...12] = 3
$TC_MPP2[1,13...18] = 15
$TC_MPP2[1,19...24] = 1080
$TC_MPP2[1,25...30] = 5

$TC_MPP2[1,31...36] = 18

Hierarchy: $TC_MPTH[0,0] = 124
$TC_MPTH[0,1] = 3
$TC_MPTH[0,2] = 15
$TC_MPTH[0,3] = 1080
$TC_MPTH[0,4] = 5
$TC_MPTH[0,5] = 18

If a tool of type_15 ($TC_TP/) is loaded, it is preferable for it to be stored at locations 13...18. If none of these locations are free, the search for an empty location continues, in accordance with the hierarchy, at locations of type_1080.

5.4.6 Distance to change position

$TC_MDPx[n,m]

Distance from magazine zero

$TC_MDPx[n,m]=value

x: = 1 : Loading magazine: loading point, loading station (1st int. mag.)
2 : Buffer magazine: spindle, gripper,..(2nd int. mag.)
n: = Magazine no. of real magazine
m: = Location no. of internal magazine (loading point,..).
Value: = Distance in no. of locations

OPI block TPM

Calculation of line: (location no.–1)*numPlaceMulti*numPlaceMultiPa rams+Parameter no.

Calculation of column: Magazine no.

Magazine data: Distance to change position				
NCK identifier	**Type**	**Description**	**OPI VAR**	**Type**
$TC_MDP1	INT	Distance between tool change position of magazine n and location m of 1st internal magazine (loading magazine, 9999)	multiPlace	WORD
$TC_MDP2	INT	Distance between tool change position of magazine n and location m of 2nd internal magazine (loading magazine, 9998)	multiPlace	WORD

Description

The current magazine position is required for tool change, loading and unloading. This position refers to the magazine zero point established by the machine manufacturer. This is usually at the change position.

The number of the location at the magazine zero point has to be given during initialization. Otherwise, non-existent location 0 is taken to be the change position.

If the magazine is moved by a task, the current position is changed accordingly. The NC does not know how many positions the magazine has moved but knows the targets of the relevant commands. On the basis of the defined distance between and object (e.g. spindle 2) and the change position, the NC is able to update the current position.

Note:
In SW 5 and later, the value of the distance and the current magazine position is also evaluated for box magazines.

For empty location searches and tool searches, search strategies based on reference to the current magazine position convert the position contained in system variable $TC_MAP8 to the change position, loading point in each case at which the search is started. With search tasks, the NCK always specifies internally which change position, loading point is to be used as reference for the search.

Notice

Command $TC_MDP2[n,m]=9999 can be programmed to unlink the relationship between spindle and magazine.

Example:

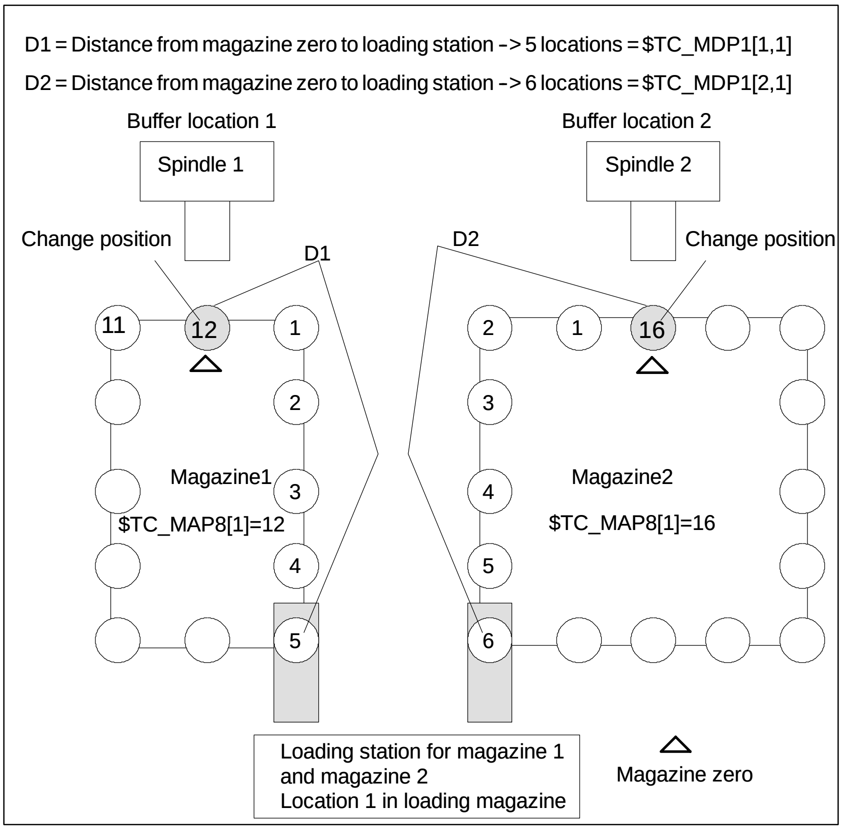


Bild 5-7 Distance to change position $TC_MDPx[y,z]=value

The magazine zero point is usually the change position of the spindle. Therefore, the following applies:

- If location 1 is located at zero position, the current magazine position = 1 ($TC_MAP8[1]).

Examples for programming the distance to the zero position:

$TC_MDP1[1,1] = 5	Distance between location 1 of the loading station and the zero position of magazine 1
$TC_MDP1[2,1] = 6	Distance between the same location and the zero position of magazine 2
$TC_MDP2[1,1]=0	Distance between location 1 of 2nd internal magazine and zero position of magazine 1
$TC_MDP2[2,2]=0	Distance between location 2 of 2nd internal magazine and zero position of magazine 2

5.4.7 Magazine blocks

$TC_MAMPx

Magazine block data
x: = Parameter 1, 2, 3

OPI block TMC

Calculation of line: N/A

Calculation of column: N/A

Magazine block data, magazine control block					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preas-sign-ment**
$TC_MAMP1	String	Identifier of the configuration magazine	magCBIdent	String	“ ”
		Number of the loading magazine	magBLMag	WORD	
		Number of the buffer magazine	magZWMag	WORD	

Magazine block data, magazine control block					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preas-sign-ment**
$TC_MAMP2	INT	Type of tool search (bits 0...7) and type of empty location search (bits 8...15) Bit0=0 Default strategy – take the first available tool found in the tool group. First search in the magazine from which the last change took place. Bit 0 Select the “active” tool in the magazine of the previously changed tool; otherwise search for replacement tool with lowest duplo number If no tool is found in this magazine, continue searching in the other associated magazines Bit 1 Search for next replacement tool at the shortest possible distance from the current magazine position Bit 2 Select the “active” tool otherwise replacement tool with the smallest number contained in $TC_TP10 Bit 3 Search for tool in the group with the **lowest actual value** for the monitored quantity Bit 4 Search for tool in group with the **highest actual value** for the monitored quantity	magSearch		0
$TC_MAMP2	INT	Bit 5 Only consider those tools whose actual value is at least the factor $AC_MONMIN of the set value away from the limiting value Bit 7 1: Search for tool acc. to assignment sequence between “Spindle and magazine” (always from 1st magazine of the distance table) 0: Start the tool search in that magazine where the tool last changed came from	magSearch		0

Magazine block data, magazine control block					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preas-sign-ment**
$TC_MAMP2		Bit 8 Search forwards starting at first location number Bit 9 Search forwards starting at current magazine position Bit 10 Search backwards starting at last location number			
$TC_MAMP2		Bit 11 Search backwards starting at current magazine position Bit 12 Symmetrical search starting at the current magazine position Bit 13 (1:1 exchange) behavior prior to SW version 5. If the 1:1 exchange is not possible, the search strategy for an emptly location is considered “symmetrical”. Behavior from SW version 6: The 1:1 exchange acts in addition to other set search strategies. If possible, the 1:1 exchange is treated as a priority.			
$TC_MAMP3	INT	Procedure for tools in a wear group (bit 0...7) Search strategies for wear groups (bit 8...15) Bit 0=0 When a wear group is activated, the tool status remains unchanged Bit 0=1 When a wear group is activated, the tool status changes. One tool from each tool group becomes active	modeWear-Group	WORD	0

Magazine block data, magazine control block					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Preassignment**
$TC_MAMP3		Bit 1=0 When a wear group is disabled, the tool status remains unchanged Bit 1=1 When a wear group is disabled, the tool status is changed Bit 2...7 Reserved Search strategy for next wear group Bit 8=0 Find the next possible wear group Bit 8=1 Find the wear group with the next highest activable group number Bit 9...11 Reserved			
$TC_MAMP3		Search strategy within the tool group for the tool to be set to active (language command SETTA or PI_SETTST) Bit 12=0 Smallest possible duplo number Bit 12=1 Smallest possible magazine location number Bit 13=1 Smallest number contained in $TC_TP10			

$TC_MAMP2

The tool search is magazine-specific.
Bit 7 is used to set whether the search is to start in the magazine from which the requesting spindle loaded its last tool for change (bit 7=0 is the default setting or in the 1st magazine of the distance table (bit 7=1)).
Bit 7 is only relevant for strategies which are set with bit 0...2.

The following applies:
The tool sequence is a tool group is not defined.
This means the tools are <u>not</u> sorted in any order (e.g. ascending duplo numbers).

5.4.8 Assignment of buffers to spindles

$TC_MLSR[x,y]

Assignment of buffer locations to spindles – $TC_MLSR[x,y]

x: = location no. in buffer 1... 32000
y : = location no. of the spindle in buffer magazine 1... 32000

OPI block N/A

Calculation of line: N/A

Calculation of column: N/A

NCK identifier	Type	Description	OPI variable	Type	Preassignment
$TC_MLSR[x,y]	INT	System variable for assigning magazine locations of the buffer magazine to the spindle	–	–	0

The programming sequence determines the sequence in which tools are automatically returned.

Notice

The content value of the system variable is not evaluated. The assignment is defined via indices x and y. In order to check via the part program whether a certain assignment exists, a read operation has to return the value zero.
If working with magazine configurations that were created before SW version 3.2, this system variable must be additionally defined, if, apart from the spindle and the tool magazine, additional buffer locations (e.g. gripper) are involved. The NCK can only find the tools in these buffers during tool search if the parameter is defined. The definitions that are established by this parameter allow, for example, the NCK to detect during boot with backup whether a tool change was interrupted during power OFF and to determine the buffer location where the tool is currently positioned.

Notice

No more than 16 magazines or buffer locations can be assigned to one spindle.

Magazine distance to buffer via toolholder/spindle

$TC_MDP2 and $TC_MLSR establish a relation between the buffer locations and magazines (see Fig.5-8).

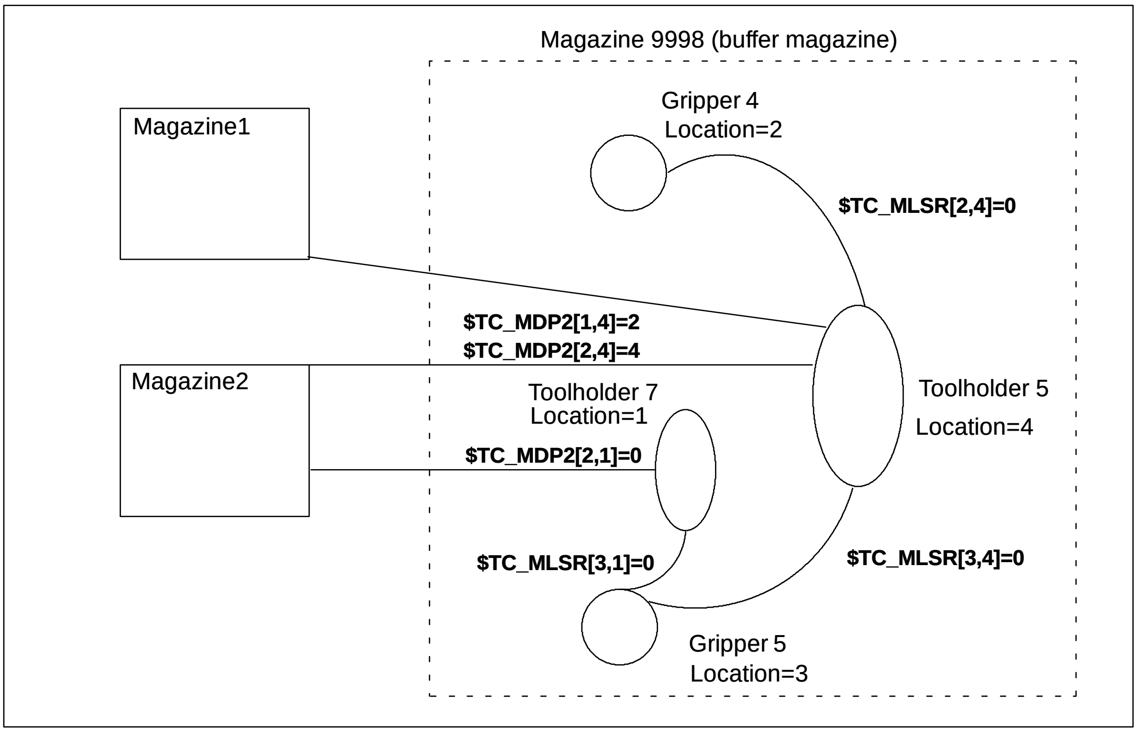


Bild 5-8 Magazine distance to the buffer

Configuration

Let us assume two magazines are defined with the numbers 1 and 2.
Four locations 1, 2, 3 and 4 are defined in buffer 9998; two toolholders 5, 7 and two grippers 4, 5.

```
$TC_MPP1[9998,1] = 2      ;location type = spindle or toolholder
$TC_MPP5[9998,1] = 7      ;tool holder no. = 7

$TC_MPP1[9998,2] = 3      ;location type = gripper
$TC_MPP5[9998,2] = 4      ;gripper no. = 4

$TC_MPP1[9998,3] = 3      ;location type = gripper
$TC_MPP5[9998,3] = 5      ;gripper no. = 5

$TC_MPP1[9998,4] = 2      ;location type = spindle or toolholder
$TC_MPP5[9998,4] = 5      ;tool holder no. = 5
```

Both grippers are linked to toolholder 5 via $TC_MLSR. They do not require their own distance definition for the magazines. They are linked to the magazines via toolholder 5 with the distance relations defined there. However, it is also possible to define separate distance relations for the grippers.

Toolholder 5 is linked with both magazines via $TC_MDP2.

Toolholder 7 is only linked with magazine 2; gripper 5 is assigned to it.

5.5 Adapter data

$TC_ADPTx[n]

If machine data $MN_MM_NUM_TOOL_ADAPTER is set to a value = –1 or > 0, the adapter data is defined, deleted, read and written via the following variables

x: = Parameter 1...3, T
n: = Number of the adapter

OPI block AD

Calculation of line: Length 1, 2, 3 = line 1, 2, 3, transformation = line 4

Calculation of column: Adapter number

Adapter data				
Name	**Type**	**Description**	**OPI VAR**	**Type**
$TC_ADPT1	Double	Adapter geometry: Length 1	adaptData	REAL
$TC_ADPT2	Double	Adapter geometry: Length 2	adaptData	REAL
$TC_ADPT3	Double	Adapter geometry: Length 3	adaptData	REAL
$TC_ADPTT[n]	Double	SC number	adaptData	REAL

The adapter geometry values act on the geometry values of the cutting edge in the same way as system variables $TC_DP 21, $TC_DP 22 and $TC_DP 23. These parameters are available only when the tool management is active.

Transformation numbers 1 to 8 can be programmed for the adapter transformation function. The parameter is available only when the tool management is active.

$TC_MPP7[m,p] Number of adapter assigned to magazine location
Value=0 No adapter assigned to location
Value>0 Number of assigned magazine

5.6 Toolholder data

The orientation of the tool can be changed for a class of tool machines. The orientation once set is however subsequently fixed during operation and in particular cannot be changed during traversing. Therefore, a kinematic orientation transformation for this type of machine is neither necessary nor meaningful.
It is however necessary to consider the changes in the tool length components attributable to the change in the orientation. These calculations are assumed by the PLC.

The following must be available to calculate the change of tool length components

- **Tool** data (geometry, wear ...)
- Toolholder data (data for the geometry of the toolholder with orientation capability).

A defined toolholder must be specified for the control for the function "orientable toolholder":

$TC_CARRx

x: = Parameter 1...33

The maximum number of toolholders can be defined in machine data 18088: MM_NUM_TOOL_CARRIER. The value is divided by the number of active TO units. The integer result indicates how many toolholders can be defined per TO unit. Values not set by the user are preset to 0.

OPI block TC

Calculation of line: Number of toolholder

Calculation of column: N/A

Toolholder data				
Name	**Type**	**Explanation/description**	**OPI VAR**	**Type**
$TC_CARR1	Double	No. of toolholder x component of offset vector L1	TcCarr1	REAL
$TC_CARR2	Double	No. of toolholder y component of offset vector L1	TcCarr2	REAL
$TC_CARR3	Double	No. of toolholder z component of offset vector L1	TcCarr3	REAL
$TC_CARR4	Double	No. of toolholder x component of offset vector L2	TcCarr4	REAL
$TC_CARR5	Double	No. of toolholder y component of offset vector L2	TcCarr5	REAL
$TC_CARR6	Double	No. of toolholder z component of offset vector L2	TcCarr6	REAL

Toolholder data				
Name	**Type**	**Explanation/description**	**OPI VAR**	**Type**
$TC_CARR7	Double	No. of toolholder x component of rotary axis V1	TcCarr7	REAL
$TC_CARR8	Double	No. of toolholder y component of rotary axis V1	TcCarr8	REAL
$TC_CARR9	Double	No. of toolholder z component of rotary axis V1	TcCarr9	REAL
$TC_CARR10	Double	No. of toolholder x component of rotary axis V2	TcCarr10	REAL
$TC_CARR11	Double	No. of toolholder y component of rotary axis V2	TcCarr11	REAL
$TC_CARR12	Double	No. of toolholder z component of rotary axis V2	TcCarr12	REAL
$TC_CARR13	Double	No. of toolholder Angle of rotation alpha1	TcCarr13	REAL
$TC_CARR14	Double	No. of toolholder Angle of rotation alpha2	TcCarr14	REAL
$TC_CARR15	Double	No. of toolholder x component of offset vector L3	TcCarr15	REAL
$TC_CARR16	Double	No. of toolholder y component of offset vector L3	TcCarr16	REAL
$TC_CARR17	Double	No. of toolholder z component of offset vector L3	TcCarr17	REAL
$TC_CARR18	Double	No. of toolholder x component of offset vector L4	TcCarr18	REAL
$TC_CARR19	Double	No. of toolholder y component of offset vector L4	TcCarr19	REAL
$TC_CARR20	Double	No. of toolholder z component of offset vector L4	TcCarr20	REAL
$TC_CARR21	Axis	No. of toolholder Axis name of 1st rotary axis	TcCarr21	String
$TC_CARR22	Axis	No. of toolholder Axis name of 2nd rotary axis	TcCarr22	String
$TC_CARR23	Char	No. of toolholder Kinematic type	TcCarr23	String
$TC_CARR24	Double	No. of toolholder Offset of 1st rotary axis in degrees	TcCarr24	REAL
$TC_CARR25	Double	No. of toolholder Offset of 2nd rotary axis in degrees	TcCarr25	REAL
$TC_CARR26	Double	No. of toolholder Offset of Hirth gears in degrees for 1st rotary axis	TcCarr26	REAL
$TC_CARR27	Double	No. of toolholder Offset of Hirth gears in degrees for 2nd rotary axis	TcCarr27	REAL

Toolholder data				
Name	**Type**	**Explanation/description**	**OPI VAR**	**Type**
$TC_CARR28	Double	No. of toolholder Increment of Hirth gears in degrees for 1st rotary axis	TcCarr28	REAL
$TC_CARR29	Double	No. of toolholder Increment of Hirth gears in degrees for 2nd rotary axis	TcCarr29	REAL
$TC_CARR30	Double	No. of toolholder Minimum position of 1st rotary axis	TcCarr30	REAL
$TC_CARR31	Double	No. of toolholder Minimum position of 2nd rotary axis	TcCarr31	REAL
$TC_CARR32	Double	No. of toolholder Maximum position of 1st rotary axis	TcCarr32	REAL
$TC_CARR33	Double	No. of toolholder Maximum position of 2nd rotary axis	TcCarr33	REAL

Further references:

/FB1/ Description of Functions, Basic Machine; Tool Compensation (W1) and

/PGA/ Programming Guide Advanced

5.7 Custom user variables

User-definable parameters

These programmable variables provide the user with three customizable parameters. These system variables are transferred to the PLC over the user interface with the T selection signal. They allow the user to send additional tool management information to the PLC. The parameters can be read and written by the NC program. They are not buffered and are set to “0” on Reset or end of program.

$P_VDITCP[x]

x: = Parameters 0, 1, 2

NCK identifier	Description	Format
$TC_VDITCP[0]	Tool management VDI user parameter 0	int
$TC_VDITCP[1]	Tool management VDI user parameter 1	int
$TC_VDITCP[2]	Tool management VDI user parameter 2	int

Interface DB72, DB73

The user parameters are output in DB 72 and DB 73 on the tool management interface. They are only valid when the status of the interface is active. The format is DINT.

Example

$P_VDITCP[0]=12;	DB72.DBD(n+4) =12 or
$P_VDITCP[1]=33;	DB72.DBD(n+8) =33 or
$P_VDITCP[2]=2000;	DB72.DBD(n+12) =2000

T=“tool”

The variables must be set in the part program before the T call if these shall transferred for a tool to the PLC as well.

Programming

The parameters can be programmed as required in the NC program. The output is however always in combination with the tool preparation command programmed in the following.

Example:

```
T= "TL1"
$P_VDITCP[0] = 1
M06
$P_VDITCP[0] = 2
T= "TL2"
```

Exactly the value = 2 is also given to the PLC with the command output of T = “T2” to the PLC and not the value 1 when the M06 command is outputted to the PLC.

From SW version 6, the output of the programmed value also takes place when M6 is programmed. I.e. the output can now also be realized with the command number 3 provided $MC_CHANGE_MODE=1 has been set.

5.8 NC commands

See also table in Subsection 5.12.5.

5.8.1 CHKDNO – Uniqueness check for D number

D number uniqueness is understood here (not for replacement tools) as being that the D numbers of all tools defined in the TO unit may occur exactly only once => the D numbers are unambiguous and absolute. This is known in the tool-management function as the possibility of assigning "unique" D numbers only. The distinction is made on the basis of replacement tools that are generally present.

Status = CHKDNO (T1, T2, D)

Parameters used:

TRUE	The D numbers have been assigned on the basis of unambious guity for the checked area
FALSE	A D-number collision is the result or the parametrization is invalid

The parameters are optional.

CHKDNO (T1,T2)	All D numbers of the referenced tools are checked.

D numbers of replacement tools

Replacement tools can be defined and used when tool management is active. The machining part program does not generally indicate whether any replacement tools are available. As a rule, the machining program addresses tools with T="Identifier". (Programming T="location number" is referred back to T="identifier" internally). The program otherwise only contains the actual programming of the offset (the D number). Therefore, the D number for the tool and replacement tool must be identical.

Example

Active tool and replacement tools for T="drill_5mm"

- T No. = 10 with D numbers 1, 2, 3 (active)
- T No. = 11 with D numbers 1, 2, 3 (replacement)
- T No. = 12 with D numbers 1, 2, 3 (replacement)

Active tool and replacement tools for T="drill_3mm":

- T No. = 20 with D numbers 1, 2, 3 (active)
- T No. = 21 with D numbers 1, 2, 3 (replacement)
- T No. = 22 with D numbers 1, 2, 3 (replacement)

CHKDNO without parameters specified, detects a collision of D numbers 1, 2 and 3 for “drill_5mm” with D numbers 1, 2 and 3 for “drill_3mm”, but not between the D numbers of the active and replacement tools.

The collisions are displayed as individual alarms, e.g.:

- “Channel 1 D number 1 defined for tool T no. 10 and 20”
- “Channel 1 D number 1 defined for tool T no. 10 and 21”

The state = FALSE is also returned in the event that the parameterization is invalid (the T or D number called is not defined in the channel).

If MAX_CUTTING_EDGE_NO <= MAX_CUTTING_EDGE_PER_TOOL, CHKDNO always returns the TRUE state, irrespective of the parameter settings.

5.8.2 CHKDM – Uniqueness check within a magazine

With active tool management, the command CHKDM checks the data in NCK for D number uniqueness within one or more magazines. The functionality corresponds to CHKDNO. The parameters are optional.

state = CHKDM (magazine no, D no., toolholder no.)

Result of check:

Value = TRUE Checked D numbers are unique.
Value = FALSE Check did not return uniqueness.

Explanation of parameters:

MagNo	Magazine number of magazine to be checked. Omission of the parameter or setting it to zero means that all tools in the magazines linked to the spindle no. or toolholder no. specified in the 3rd parameter are checked.
Dno	The D number against which the check will be made. Omission of the parameter or setting it to zero means that all D numbers of the called magazine will be checked for unambiguity.
Toolholder no.	Indicates which spindle numbers or toolholder numbers the magazines shall be checked for. Omission of the parameter means that the magazines are given for the check from the distance table of the spindle location for the master spindle or the master toolholder.

5.8.3 GETACTTD – Determine the T no. for a unique D no.

For an active tool management (e.g. measuring-cycle programs), this command serves to conclude the associated T number **of the tool active in the tool group** from a D number.

status = GETACTTD (Tnr, Dnr)

Dno	D number for which the T number is to be searched. No uniqueness check for D numbers is applied. If the same D numbers are defined in different tool groups of the same TO unit, then the T number of the first tool group that is found determines the tools that have the specified number.	
Tno	T number found	
status	Result of search:	
	0	T number found, T no. is assigned the value
	–1	No T number exists for the specified D number, T no. assigned the value 0.
	–2	D number is not unique; Tno is assigned the value of the D number that was first determined.
	–3	The tool group does not contain any tools of the specified status or D number. T no. assigned the value 0.
	–4	The tool group contains several tools that have the desired status and the D number that has been searched. T no. is assigned the value of the first tool be found with the desired D number.
	–5	Function could not be executed for other reasons.

5.8.4 GETDNO – Get D numbers

The language command

d = GETDNO(t, ce)

allows the offset number d to be read for the cutting edge ce of the tool with the T number t. If t or ce are parameters which have no data record, d=0 is returned. Any parameters violating the syntax rules will generate an alarm.

The command is only available if $MN_MAX_CUTTING_EDGE_NO > $MN_MAX_CUTTING_EDGE_PER_TOOL.
$MN_MAX_CUTTING_EDGE_NO <= $MN_MAX_CUTTING_EDGE_PER_TOOL returns GETDNO d=ce as D number.

5.8.5 SETDNO – Rename D numbers

The language command

state = SETDNO(t, ce, d) allows the offset number d of cutting edge ce of tool t to be set or changed. If t or ce are parameters which have no data record, state = FALSE is returned. Any parameters violating the syntax rules will generate an alarm.

t, ce, d must be > 0; d=0 cannot be set.

5.8.6 DZERO – Invalidate D numbers

Marks all D numbers of the TO unit as invalid. This command is used for support during conversion or re-equipping.
Offset data sets tagged with this command are no longer verified by the CHKDNO language command. The D numbers have to be set with SETDNO again in order to make these accessible again.

5.8.7 DELDL – Delete additive offsets

This command deletes the additive offsets for the cutting edge of a tool (to release memory space). Both the defined wear values and the setup values are deleted.

status = DELDL(t, d)

Explanation of the parameters:

DELDL(t, d)	All additive offsets of the cutting edge with D number d of tool t are canceled.	
DELDL(t)	All additive offsets of all cutting edges of tool t are canceled.	
DELDL	All additive offsets for all cutting edges of all tools of the TO units are canceled (of the TO unit of that channel where the command is programmed)	
status	Result of search:	
	0	Offsets have been successfully deleted.
	–1	Offsets have not been deleted (if the parameter settings specify exactly one tool edge), or not deleted completely (if the parameter settings specify several tool edges)

5.8.8 NEWT – Create a new tool

A new tool can bet set up in a number of ways by NC commands in NCK. Either by programming T no.=**NEWT**("TL", Duplo no.) or by programming a system variable **$TC_**...

Note that NEWT automatically generates a cutting edge with CE no. = 1, D no. = 1 (**SW 6**). If you want the tool to have a different CE no., you need to change this number after it has been generated.

The NEWT function allows a new tool to be created without specifying a T no. The function returns the automatically generated T no. with which the tool can subsequently be addressed. The 1st cutting edge is automatically created when a new tool is created. All offsets are set to "0" by default.

Return parameter = NEWT ("TL", duplo no.)

If it is not possible to create a new tool for any reason, the NEWT(...) function generates an alarm.

Specification of a duplo number is optional. It is generated in the NCK if it is not specified. (duplo no.= old duplo no. +1)

Examples

Example 1:

Create a new tool with NEWT and the CE/D numbers = 2, 47

def int tnr

```
tnr = NEWT("tool", 111)   ; Tool with Ident/duplo no.="tool"/111,
                          ; T no.=tnr=1 in the example and a cutting
                          ; edge
                          ; CE=1, D=1 is created
                          ; The cutting edge is to be named CE=2,
                          ; D=47
$TC_DPCE[tnr, 1]=2        ; Rename the CE number
SETDNO(tno,2,47)          ; Rename the D number
                          ; Assign the remaining data for the tool/cutting
                          ; edge
```

Example 2:

Create tool "tool"/111, T no.=tnr=1 with $TC... and CE numbers = 2 ,4 (let us assume that T no.=1 does not yet exist)

```
$TC_TP1[1]= 111        ; Create tool with T no.=1, duplo no.=111
$TC_TP2[1] = "tool"    ; Assign tool^ Ident="tool"
$TC_TPCE[1,47] = 2     ; Create new D=47, assign CE no.=2
                       ; Assign the remaining data for the tool/cutting
                       ; edge
```

The function is used for creating tools in a loading program (load cycle).

5.8.9 DELT – Delete tool

A tool can be deleted with the DELT(...) function by specifying the tool identifier and duplo number. Only tools that have been unloaded can be deleted.

DELT("MYTOOL",DUPLO_NR)

All tool-related data is set to 0 (user data, hierarchy data, ...).

Example:

DELT("DRILL", DUPLO_NO)

The function is for deleting tools in the part program.

5.8.10 GETT – Read T no.

The GETT function returns the associated T number on the basis of the tool identifier and its duplo number.

Return parameter = GETT("TL", DUPLO_NO);

If the tool identifier or duplo number cannot be assigned to a tool, value –1 is returned. Specification of the duplo number is optional.

If no duplo number is entered, the T number of the 1st tool from the group of tools with the specified identifier is returned.

Example:

Determine the T number for drill with duplo number

```
R10=GETT("DRILL", DUPLO_NO)                 ; The T number is in R10
$TC_TPx,[GETT("DRILL",DUPLO NO)]=value      ; Write tool-related data
```

This function is used to retroload tools via the part program.

5.8.11 SETPIECE – Decrement workpiece counter

With the SETPIECE function, the user can update the count monitoring data of the tools associated with the machining process. All the tools that have been loaded at change since the last time SETPIECE was activated are included in the update. The function serves as a rule for programming at the end of the NC main program to decrement the count from all the tools associated with count monitoring.

Notice

The command is not active in the block search (with/without calculation). If the value for the count = 0, the internal table for flagged tools/cutting edges is deleted.

Programming

SETPIECE(x,y)

x := 0 ... 32000	Value used when decrementing
y := 0...8	Spindle index, value 0 means index of main spindle (need not be programmed)

Example:

SETPIECE(1);	Workpiece counter of main spindle is decremented by 1
SETPIECE(1,1);	Workpiece counter of spindle no. or toolholder no. 1 is decremented by 1
SETPIECE(4,2);	Workpiece counter of spindle no. or toolholder no. 2 is decremented by 4

Example of SETPIECE with M06 tool change command:

The tools involved in a tool (program) are to be decremented by the value 1.

```
T1              ;T1 is preselected (relative to main spindle)
M06             ;T1 is changed
D1              ;D1 becomes active
T2              ;T2 is preselected
:               ;machining program
:
M06             ;T2 is changed
D1              ;D1 of T2 becomes active
T3              ;T3 is preselected
:               ;machining program
:
:
M06
T0              ;preparation for clearing the spindle
:
M06             ;clear spindle
SETPIECE(1)     ;SETPIECE to all tools
M30
```

The counter is to be decremented for each tool

In this example, tools T1, T2 and T3 are to machine a program. All three tools are monitored for workpiece count. The aim is to decrement tool T1 by the value 1, T2 by the value 2 and not to decrement T3.

As from NCK software versions **5.3.34, 6.3.15** the following needs to be programmed:

```
N500 T1
N600 M06
N700 D1              ; With the offset selection, the tool that was loaded at change
                     ; is stored in the SETPIECE memory
N900 T2              ; Preparation of next tool
                     ; Machining command
:
N1000 setpiece(1)    ; SETPIECE acts on T1, Setpiece memory is cleared
N1100 M06
N1200 D1
N1400 T3
:                    ; Machining commands
:
N1500 setpiece(2)    ; only acts on T2
N1600 M06
N1700 D1
:                    ; Machining commands
:
N1800 setpiece(0)    ; only acts on T3, no decrementing
N1900 T0
N2000 M06
N2100 D0
N2300 M30
```

Prior to NCK software versions **5.3.34, 6.3.15** the following needs to be programmed:

The command SETPIECE(0) must generally be programmed after the change, including the offset selection.

```
N500 T1
N600 M06
N700 D1
N800 setpiece(0)     ; previously flagged tools for workpiece count are deleted
N900 T2
N1000 setpiece(1)    ; SETPIECE acts on T1
:                    ; Machining commands
:N1100 M06
N1200 D1
```

```
N1300 setpiece(0)   ; delete command for flagged tools
N1400 T3            ; in this block, T2 is identified as the "active" tool and entered
                    ; in the table of flagged tools
N1500 setpiece(2)   ; only acts on T2
N1600 M06
N1700 D1
N1800 setpiece(0)   ; delete command for flagged tools
N1900 T0
N2000 M06
N2100 D0
N2200 setpiece(0)   ; delete command for flagged tools, now no tools are flagged
                    ; for SETPIECE
N2300 M30
```

5.8.12 GETSELT – Read the selected T no.

The function is available with TMMG and provides the T number of the tool preselected for the spindle. This allows, for example, the offset data to be accessed prior to M06.

GETSELT (return parameter, x);

x: = 1–32 spindle number
x: = 0 index for main spindle

Specification of "x" is optional. If "x" is not specified the function refers to the main spindle.

Return parameters

> 0	T no. of prepared tool
= 0	No preparation or T0 was programmed
= –1	Preparation failed (e.g. no tool ready to use)

Example:

```
T="DRILL"
...
. . .
GETSELT(R10)          ;read preselected T no. for the main spindle
```

This function compares in the tool change cycle whether the preselected tool is already placed in the spindle.

5.8.13 GETEXET – Read the T number to be loaded at change (SW 6)

The command GETEXET is specifically designed for block search. Its parameters are set in the same way as for GETSELT. It returns the T number of the tool that is active from the point of view of the NC program.
(For a detailed description, please refer to Section 3.2.15 Block search.)

GETEXET(return parameter, x)

Return parameter:	0	no tool active
	> 0	T no. of active tool
x:	1 – 32	spindle number
	0	master spindle

Specification of the spindle number is optional. If it is not specified, GETEXET refers to the current master spindle.

Example:

Let us assume the following conditions: Change is set with M06.
There is no tool in the spindle.
There are two tools “Drill_10mm” (T-Nr. 1), “Drill_4.2mm” (T no. 4)

```
. . .
N30 T="Drill_10mm"; T no. 1
. . .                              –> GETSELT=1 (T1 is prepared)
. . .                              –> GETEXET=0 (no tool active)
N40 M06
N42 G90 G00 D1 ...
. . .                              –> GETSELT=1 (last prepared tool)
. . .                              –> GETEXET=1 (active tool)
N50 T="Drill_4.2mm"; T no. 4
. . .                              –> GETSELT=4 (new preparation: T4)
. . .                              –> GETEXET=1 (T1 is active)
N60 M06
N62 G90 G00 D1 ...
. . .
```

5.8.14 GETACTT – Read the active internal T no.

This function provides the option to read the T number of the tool with the status “active” (a tool becomes “active” immediately before is is loaded into the toolholder) and “was in use” out of a tool group with the identifier “name” by means of the parameter “T no.”.

status=GETACTT(Tno,name)

The return parameter “status” indicates whether the call was successful/failed:

0	Function successful; T no. contains the desired value
–1	No tool matching the specified identifier exists; T no. contains value = 0
–2	The tool group does not contain a tool with the desired status; T no. contains value = 0
–3	There are several tools with the desired status in the in tool group; T no. contains the value of the first tool with the desired status

GETACTT can have several meanings! It is always conceivable there are several tools in the tool group that have the same status. The command will only then meaningfully function when the user ensures there is exactly one tool with the desired status in the tool group.
The command does not initiate a main synchronization. It may be necessary to enter STOPRE before the call.

Example:

Tool group “Drills” contains three tools with the duplo numbers 1, 2, 3 and the T numbers 1, 2, 3:

```
def int Tno, status              ; in the tool group “Drill” initially there is no
                                 ; active tool

status=GETACTT(Tno, “Drill”)     ; status=–2, Tno=0
T=“Drill”                        ; Preparation sets tool status to “active”

status=GETACTT(Tno, “Drill”)     ; status=0, Tno=0
                                 ; the tool is active, but the identifier “was in
                                 ; use” is not yet applied
M06                              ; Change
T=“Hugo”                         ; Preparation

status=GETACTT(Tno, “Drill”)     ; status=–2, Tno=0
                                 ; the tool is active, but the identifier “was in
                                 ; use” is still not yet applied
M06                              ; Change

status=GETACTT(Tno, “Drill”)     ; status=0, Tno=1
                                 ; Read request is performed
                                 ; The tool “Drill” is now assigned the status
                                 ; “was in use” due to removal at change
                                 ; The status “active” remains applied
```

Notice

1. GETACCT cannot detect a tool which is positioned in the spindle for its first use.
2. The tool sequence is not defined for a group. This means that GETACCT will read any random tool in the group where the status bits “active” and “was in use” are set.

5.8.15 SETMS – Spindle can be declared master spindle

Available with TMBF, TMFD, TMMO, TMMG.

SETMS(n) declares the spindle specified in n to be the master spindle. A spindle can also be defined as the master via a machine data.

The programmed values from SETMS can remain active beyond program end/ reset/Start.

When SETMS is programmed without a spindle name, the spindle programmed in the machine data used instead.

5.8.16 SETMTH – Set master toolholder number

Available with TMMG.

Using the machine data **MD 20124: TOOL_MANAGEMENT_TOOLHOLDER**, you can determine whether to assign a toolholder number instead of a spindle number to determine the location for a tool that is to be loaded at change. Use of this language command is only meaningful if the MD > 0.

Programming example:

T=“Miller” M06	No address extension programmed –> this refers to the master toolholder; i.e. toolholder 1 (value of machine data TOOL_MANAGEMENT_TOOLHOLDER)**.** The tool change is performed in the buffer location with $TC_MPP5=1. The path is corrected with the tool offsets.
. . .	
T2=“Drill” ..M2=6	Address expansion for the secondary toolholder was programmed. The tool is changed in buffer location 2. The path is not corrected.
. . .	
SETMTH (2)	Declare toolholder 2 the**master toolholder**

T="Miller_2" M06	No address extension programmed -> this refers to the master toolholder; i.e. toolholder 2. Tool change is performed and tool is placed into buffer location 2. The path is corrected with the tool offsets.
. . .	
T1="Drill_1" M1=6	Address extension for the secondary toolholder has been programmed. The tool change is performed in the buffer location with $TC_MPP5=1. The path is not corrected
. . .	
SETMTH	Declare the toolholder specified in TOOL_MANAGEMENT_TOOLHOLDER as the master toolholder
T="Miller_3" M06	No address extension programmed -> this refers to the master toolholder; i.e. toolholder 1 (value of MD TOOL_MANAGEMENT_TOOLHOLDER). Tool change is performed and tool is placed into buffer location 1. The path is corrected with the tool offsets.

Notice

SETMTH does not change the active tool. The new master toolholder definition cannot be taken into account until the tool change is then programmed.

The programmed values from SETMS can remain active beyond program end/ reset/Start.

Example 1:

The following applies:

$MC_RESET_MODE_MASK = "H18041"
$MC_SPIND_DEF_MASTER_SPIND = 1
$MC_TOOL_MANAGEMENT_TOOLHOLDER = 2

After the end of program/RESET both the active tool offset and the programmed values for SETMTH and SETMS remain active. The tool change still does not take place at the spindle but at the toolholder instead.

```
T="Drill" M6 D2   ;tool change on master toolholder=2
SETMS(3)          ;new master spindle=3
SETMTH(1)         ;new master toolholder=1
T="Miller" M6 D1  ;tool change on master toolholder=1
M17
```

After end of program or RESET,
spindle no. = 3 is the master spindle,
tool holder no. =1 is the master toolholder and a
tool = "Miller" with offset D1 determines the path correction.

After Power ON the settings for the MDs become effective:
spindle no. = 1 is the master spindle,
tool holder no. =2 is the master toolholder.

The tool offset is derived from the smallest D number of the tool that is located in the master toolholder; i.e.

```
T="Drill" with D1
```

(assuming that the tool has two D offsets D1, D2).

Example 2:

The following applies:

$MC_RESET_MODE_MASK = "H41"
$MC_SPIND_DEF_MASTER_SPIND = 1
$MC_TOOL_MANAGEMENT_TOOLHOLDER = 0

After the end of program/RESET both the active tool offset and the programmed value for SETMS remain active. The tool change takes place at the spindle which now becomes the toolholder.

```
T="Drill" M6 D2     ;tool change on master toolholder=1
SETMS(3)            ;new master spindle = master toolholder=3
T="Miller" M6 D1    ;tool change on master toolholder=3
M17
```

After end of program or RESET,
Spindle no. = 1 the master spindle and a
tool = "Miller" with offset D1 (on spindle with no. = 3) determines the path correction.

After Power ON the settings for the machine data become effective:
Spindle no. = 1 is the master spindle/master toolholder.

The tool offset is derived from the smallest D number of the tool that is located in the master toolholder; i.e.

```
T="Drill" with D1
```

(assuming that the tool has two D offsets D1, D2).

5.8.17 POSM – Position magazine

This NC language command enables you to initiate a magazine positioning operation to a particular location in an internal magazine (e.g. spindle, toolholder, loading magazine), irrespective of how the location is assigned or the status of the tool it contains. The language command includes some of the functions of OPI PI service (see Subsection 5.12.5) _N_TMPOSM.

The full command is: **POSM (p, m, ip, im)**

Description of function

p Location number at which the internal magazine is to be positioned.

m Magazine number of the magazine to be moved.
This parameter is optional.
If it is not set, the location number refers to the magazine contained in the distance table as the first magazine for the specified internal location.

ip Location number for the specified internal magazine (spindle location, loading magazine etc.)
This parameter is optional.
If it is not specified, the positioning operation refers to the main spindle location or the main toolholder location.

im Magazine number of internal magazine in relation to location number ip to which the magazine must be moved. An internal magazine is either a loading or a buffer magazine.
This parameter is optional.
If it is not specified, then the command refers to the buffer magazine.

The magazine (number m) must be linked by a distance relationship with the selected loading or buffer-magazine location. Alarms are generated when incorrect parameters are specified (e.g. undefined location numbers).

Sample parameter settings

Specified configuration:

- Magazine (magazine number = 1),
- Spindle (buffer magazine = 9998, location 1),
- Loading magazine (loading magazine = 9999, location 2).

Move from magazine 1, location number 4 to the spindle.

Command:

```
N100          POSM(4, 1, 1, 9998)
```

Command for traversal to loading magazine:

```
N100          POSM(4, 1, 1, 9999)
```

Example with result check

This example assumes a magazine with the configuration shown in the diagram below.

Location 12 is to be positioned at the change position and the program must not be continued until positioning has been successfully completed (simplest case with only one magazine and one defined change position).

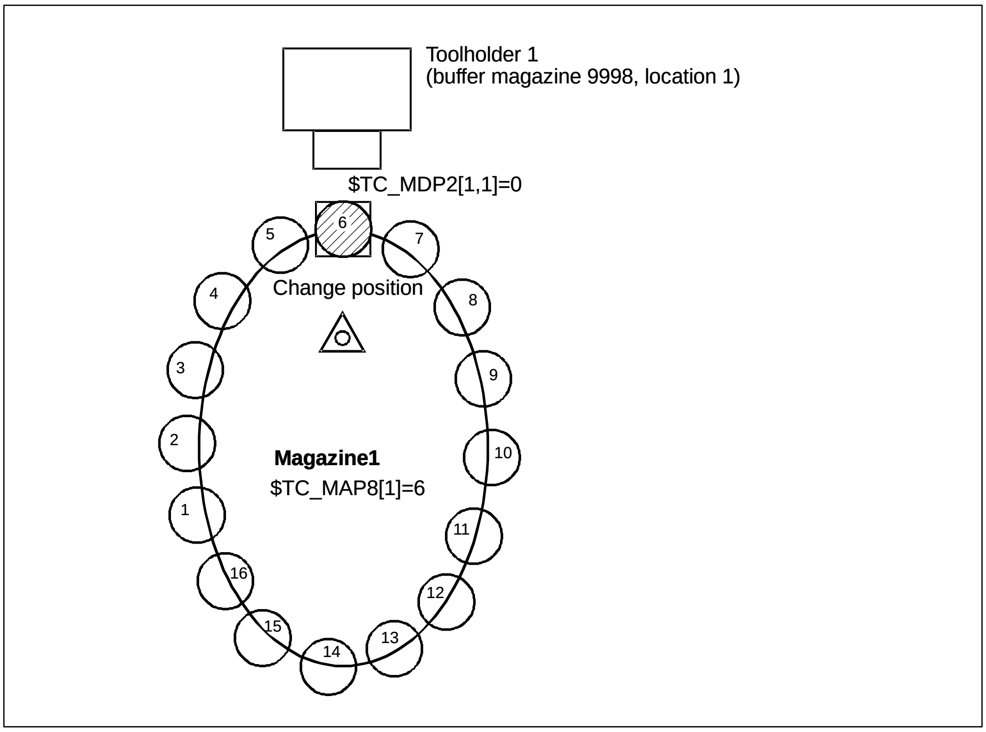


Bild 5-9 Magazine positioning with check of positioning operation result

In this example, the magazine zero point is the location in front of toolholder 1. It is defined by system variable $TC_MDP2. Toolholder 1 is assigned to the master spindle of the channel.

```
N100 POSM(12)          ; Moves location 12 to the change position, any unprogrammed
                       ; parameters are set internally to POSM (12, 1, 1, 9998)
N200 wait:
N300 G4 F1             ; waiting time according to the conditions prevailing at the machine
                       ; (exit possibly necessary if reaction is required to positioning er-
                       ; rors)
N400 if ( $TC_MAP8[1] <> 12 ) goto wait;
                       ; after POSM(12) is executed, the current magazine position must
                       ; be equal to 12.
```

References: /PGA/, Programming Guide Advanced

Notice

The language command POSM(...) is terminated without waiting for an acknowledgment from the PLC.

5.8.18 MVTOOL – Language command to move tool

The function MVTOOL allows tools to be loaded and unloaded via NC programming only. It can be used to transport a tool from one magazine location to another – regardless of where.
It is mandatory for a tool to be positioned at the source magazine location.

This language command does not generate an alarm.
Whether MVTOOL was carried out with or without error(s) must be checked via the return value of parameter “state”.

MVTOOL (state, magFrom, locFrom, magTo, locTo)

state	Success status of command	
	0	Execution was successful (PLC acknowledgement may still be pending)
	–1	Command cannot be used because TMMG is not active
	–2	Function is not carried out because of block search, program testing
	–3	Tool cannot be moved (for example because tool status “being changed” is set)
	–4	No tool located in source location
	–5	magFrom has invalid value
	–6	locFrom has invalid value
	–7	magTo has invalid value

	–8	locTo has invalid value
	–9	No distance relation defined (if exactly one magazine is an internal magazine)
	–10	No empty location found (if parameter locTo is not programmed)
	–11	Target location for tool not free
	–12	The parameter locTo must be programmed, as magTo is an internal magazine

magFrom	Magazine number of magazine in which the tool to be moved is located

locFrom	Location number of location in which the tool to be moved is located

magTo	Target magazine number of magazine to which the tool is to be moved. This can be a loading magazine, buffer magazine or another real magazine. This parameter is optional. If it is not programmed, the value from magFrom is used.

locTo	Target location number of location to which the tool is to be moved. This parameter is optional. If it is not programmed, a location search is conducted in magazine magTo – if this is a real magazine. A location search cannot be performed in an internal magazine.

Notice

Deselecting a possibly active offset is not linked with moving the tool.

If source and target location of the tool are identical, the command is not executed, “state” then has value 0.

If a tool is moved from a real magazine to an internal magazine (or vice versa), the respective magazine distance relation must be defined.

PLC

The language command generates the command (CMD=1) in the NCK for the PLC and is terminated when the command is generated. It does not wait for the acknowledgement from the PLC.

Sample parameter settings

Let us assume the following configuration:

Magazine (magazine no. = 5, locations 1,...10),
One spindle (buffer magazine no. = 9998, location 1),
One loading point (loading magazine. = 9999, location 1)
The magazine is linked to the spindle and the loading point via a distance relation (see $TC_MDP1/$TC_MDP2).

<u>1st scenario</u>

The tool from loading location 9999/1 is to be loaded in magazine 5. The following is programmed to this effect:

```
def int state
$TC_MPP6[9999, 1] =123       ;place tool with internal T no.=123 onto
                             ;loading location
MVTOOL(state, 9999, 1, 5)    ;search for a suitable empty location in magazine 5
```

If the tool is to be loaded onto precisely location 7:

```
MVTOOL(state, 9999, 1, 5, 7)  ;before loading a check is performe to ensure
                              ;location 7 is empty
```

The tool with T no.=123 is to be loaded from the loading location to spindle 1 (magazine no.=9998, location no.=1):

```
$TC_MPP6[9999, 1] =123
MVTOOL(state, 9999, 1, 9998, 1)
```

<u>2nd scenario</u>

With the same configuration, a tool that is loaded on location 7 is to be moved to another suitable location in the same magazine.

MVTOOL(state, 5, 7)

or to location 3 in the same magazine

MVTOOL(state, 5, 7, 5, 3)

or to any location in magazine 11

MVTOOL(state, 5, 7, 11)

Example with result check for tool move process

Let us assume the following configuration:

Magazine (magazine no. = 5, locations 1,...10),
One spindle (buffer magazine no. = 9998, location 1),
One loading point (loading magazine. = 9999, location 1)
The magazine is linked to the spindle and the loading point via a distance relation (see $TC_MDP1/$TC_MDP2).
The tool with the internal T no.=123 is to be loaded into location 7 in magazine 5.
The program is only to be continued if loading was successfully completed. This is the case when the PLC sends an acknowledgement. Successful execution can be

seen In the program when the tool to be loaded is positioned at the desired location.

1. Concrete target location specification:

```
def int state
$TC_MPP6[ 9999, 1] = 123
MVTOOL(state, 9999, 1, 5, 7)          ;load tool at location 5/7
if ( state < > 0 ) gotof error
wait:
 G4 F1                ; wait time corresponding to the conditions on the machine
                      ; (it would be meaningful to program time monitoring here
                      ; as well)
if ( $TC_MPP6[5,7] < > 123 ) gotob wait
                      ; after MVTOOL has been executed, the tool must be lo-
                      ; cated at the programmed location
```

2. Location search for target location

```
int state
$TC_MPP6[9999, 1] = 123
MVTOOL(state, 9999, 1, 5)     ;load tool to magazine 5 – search for empty
                              ;location
if ( state < > 0 ) gotof error
wait:
 G4 F1                ; wait time corresponding to the conditions on the machine
if ($A_MYMN[ 123] < > 5) gotob wait
                      ; after MVTOOL has been executed, the tool must be posi-
                      ; tioned in the programmed magazine
                      ; $TC_MPP6 cannot be used here as the location no. is not
                      ; known
                      ; It can be determined via $A_MYMLN[123]
```

5.8.19 SETTIA – Dectivate tool from wear group

The SETTIA function cancels the “active” status for all active tools in the selected wear group. By parameterizing the language command, this can be either magazine-specific or wear group-specific.

SETTIA (STATUS, MNR, VNR,USEKT)

STATUS	Return parameters	
	0	Function was executed correctly.
	–1	Function was not executed as there is no active wear group in the selected magazines.
	–2	Function was not executed as the programmed wear group number does not exist.

	–3	Function was not executed as the programmed magazine number does not exist.
	–4	Function was not executed as the function "wear group" is not enabled (MN_TOOL_MANAGEMENT_MASK).
	–5	Function was not executed.

All parameters are optional.
If SETTIA is not parameterized, the inactive setting refers to all loaded tools in the TO area which are in "active" state.

MNR	Magazine number	
	0	The inactive setting refers to all magazines regardless of any assignment to a spindle. In this case, the tools in the buffer are also considered as well as the toolholder.
	> 0	Magazine number in which the inactive setting is to be applied. Tools in this magazine which are in the buffer are not considered. This means that if these tools are placed back into the magazine, they still have "active" status.
	–1	All magazines with a distance relation to a spindle or toolholder.

VNR	Wear group number	
	0	The inactive setting refers to all tools which are not assigned to a wear group. If no wear group is defined, the inactive setting is applied to all tools in the magazine.
	> 0	Wear group number in which the inactive setting is to be applied.
	–1	Active wear group ($TC_MAP9).

USEKT	Tool subgroup	
	0	All tools in the group are assessed.
	> 0	The tools are assessed which have a bit set in the value programmed in USEKT in parameter $TC_TP11.
	–1	The currently programmed value of USEKT is used.

A search strategy can be set in parameter $TC_MAMP3 for the tool to be activated by SETTIA.

Bit 12 = 0	Smallest possible duplo number (default)
Bit 12 = 1	Smallest possible magazine location
Bit 13 = 1	Smallest possible number contained in parameter $TC_TP10 (sequence of use)

Notice

It is mandatory to set the wear group for the function SETTIA.

5.8.20 SETTA – Activate tool from wear group

The SETTA function sets a tool in a group to active. One tool becomes active for each tool group contained in a wear group. SETTA does not affect disabled tools If a tool is already active in the group, SETTA does not set any more to active.

Notice

The tool sequence within a tool group is not defined, i.e. SETTA will act on any tool.

Description of function

SETTA (STATUS, MNR, VNR, USEKT)

STATUS	Return parameter which can consist of the following values:	
	0	Function was executed correctly.
	1	Function was carried out, but another tool is active in the group (e.g. an unloaded tool).
	–1	Function has not been executed because there is no active wear group in the selected magazines.
	–2	Function has not been executed because the programmed wear group number does not exist.
	–3	Function was not executed as the programmed magazine number does not exist.
	–4	Function was not executed as the function “wear group” is not enabled (MN_TOOL_MANAGEMENT_MASK).
	–5	Function was not executed.

All parameters are optional.
If SETTA is not parameterized, the active setting refers to all loaded tools that are ready for use in the TO area.

MNR	Magazine number	
	0	The active setting refers to all magazines regardless of any assignment to a spindle.
	> 0	Magazine number in which the active setting is to be applied.
	–1	All magazines with a distance relation to a spindle or toolholder.
	–2	One tool in one group becomes active for each spindle/toolholder in the assigned magazine(s). This means there can be e.g. two toolholders with one magazine assigned to each. The tools of a group can be distributed in any configuration among the two magazines. If tools from one group are distributed among two magazines, then two tools are set to active for each group.

VNR	Wear group number	
	0	The active setting refers to the entire magazine.
	> 0	Wear group number in which the inactive setting is to be applied.
	–1	Active wear group ($TC_MAP9)

USEKT	Tool subgroup	
	0	All tools in the group are assessed.
	> 0	The tools are assessed which have a bit set in the value programmed in USEKT in parameter $TC_TP11.
	–1	The currently programmed value of USEKT is used.

A search strategy can be set in parameter $TC_MAMP3 for the tool to be activated by SETTA.

Notice

It is mandatory to set the wear group for the function SETTA.

5.8.21 RESETMON – Language command for setpoint activation

RESETMON (state, t, d, mon, resetStates)

Sets the actual value of tool to the setpoint.

state	Return parameter which can consist of the following values:	
	0	Command was successfully executed
	–1	The cutting edge with the specified D number d does not exist.
	–2	The tool with the specified T number t does not exist.
	–3	There is no monitoring function defined for the specified tool. This status is only possible if t has been specified explicitly.
	–4	Monitoring function is not active in the NCK, i.e. the command has not been executed.
t	Internal T number	
	t = 0	Command applies to all tools.
	t > 0	Command applies to this particular tool only.
	t < 0	The absolute value of t is formed and all sister tools of this tool are affected.
d	D number of the tool (optional parameter). If the parameter is not specified at all or is assigned the value 0, all D numbers or all cutting edges of the tool are processed.	
	d > 0	The command applies specifically to the specified D number.
mon	Optional bit-coded parameter. If the parameter is either not specified at all or assigned the value 0, all actual values of the active, tool-specific monitoring functions for the designation edge(s) are set to the setpoints.	
	mon > 0	The command applies specifically to the actual value of the specified monitoring type. Possible values are the positive values of the system parameter **$TC_TP9** (1, 2, 4, 8) or the corresponding bit combinations when several monitoring types are activated.
	mon < 0	The command applies specifically to the actual value in the “value for mon” in the specified monitoring type. There is no restriction by the system variable values $TC_TP9. The values of non-activated monitoring types can be reset in this way too. This applies in particular to the simultaneous resetting of the actual values for wear and additive offset data.

resetStates	Optional bit-coded parameter from SW version 7.3	
	Bit 0	Tool status “active” is deleted
	Bit 1	Tool status “enabled” is set
	Bit 2	Tool status “disabled” is reset if a) permitted by the monitoring data b) parameter “mon” is set accordingly
	Bit 3	Tool status “measure” is set
	Bit 4	Tool status “prewarning limit” is reset if c) permitted by the monitoring data d) parameter “mon” is set accordingly
	Bit 5	Reserved
	Bit 6	Reserved
	Bit 7	Tool status “was in use” is deleted
	Bit 8	Reserved
	Bit 9	Tool status “ingore disabled state” is deleted
	Bit 10	Tool status “to unload” is deleted

From SW version 7.3 the parameter “resetStates” allows selective modification of the tool status in addition to the monitoring parameters. The bit coding for “resetStates” corresponds to that for the tool status parameter $TC_TP8[x].

If this parameter is not specified, machine data $MN_TOOL_RESETMON_MASK is accessed. The bit coding for this data is identical to that for parameter “resetStates”. With the analog PI service PI_TRESMO, this machine data is also effective.

Notice

There is no explicit generation of alarms. The user can carry out the error handling himself/herself via the **state** parameter.

5.8.22 DELTC – Delete toolholder data block (from SW version 6)

The function “Toolholder orientation” (Section 5.6) must be active. The function can additively superimposed over the functionsTMBF, TMFD, TMMO and TMMG.

DELTC(n,m)

n	First number of the toolholder data area the values of which shall be set to zero. This parameter is optional. If it is not specified, all toolholder data blocks are set to zero starting at the smallest through to the largest block.
m	Last number of the toolholder data area the values of which shall be set to zero. This parameter is optional. If it is not specified, then the toolholder data block specified by n is set to zero. If m if greater than the largest number of a toolholder data block in this channel, then those data blocks up to the largest number are set to zero.

The toolholder data blocks are defined by the system variables $TC_CARRx. Only the command $TC_CARR1[0] was available up to now for setting all data blocks to zero. With DELTC a range of numbers for the toolholder data from n to m for the toolholder data can now be set to zero.

In particular, the contents of DELTC() and $TC_CARR1[0]=0= set all data blocks to zero, are the same.

The parameters n, m have to be programmed with values larger than zero. Other values lead to an alarm.

Parameter n must be smaller than m. Programming otherwise leads to an alarm. Also, n must lie in the range of numbers permitted for toolholder data.

The selected range of numbers must include the range of numbers for the toolholder data blocks on the channel. Programming is otherwise rejected and an alarm is issued.

If the function “Toolholder data” is not activated ($MN_MM_NUM_TOOL_CARRIER 0 0), then DELTC will also generate an alarm.

Example

In the TO unit there are 14 toolholder blocks defined with the numbers 1 to 14.

DELTC(5,8)	;sets the values of the data blocks 5, 6, 7, 8 to zero
DELTC(5,20)	;sets the values of the data blocks 5, 6, 7, ..., 14 to zero
DELTC(9)	;sets the values of the data block 9 to zero
DELTC()	;sets the values of the data blocks 1, ..., 14 to zero
DELTC(0,1)	;error –> alarm – n, m must be greater than zero
DELTC(0,–2)	;error –> alarm – n, m must be greater than zero
DELTC(0)	error –> alarm – n must be greater than zero
DELTC(15,20)	error –> alarm – n may be max. 14
DELTC(20)	;error –> alarm – n may be max. 14

5.8.23 TCA – Tool selection/tool change irrespective of tool status

This function is only available for TMMO and TMMG.

It is necessary for certain routines (e.g. measuring cycle) to load a specific tool onto the spindle/the toolholder for tool change regardless of its status (e.g. a disabled tool).

TCA(“TL name”, duplo no., toolholder no.)

“TL name”	Identifier of the tool to be loaded at change
Duplo no.	Duplo no. of the tool to be loaded at change This parameter is optional. If it is not specified, then the tool with the smallest duplo number is loaded at change
Toolholder no.	Toolholder or spindle on which the change is to take place. This parameter is optional. If it is not specified, the change refers to the currently set or programmed master spindle/master toolholder. The following applies for TMMO without TMMG: The parameter corresponds to the address extension of the T command. (The setting for MD $MC_T_M_ADDRERSS_EXT_IS_SPINO is taken into account.)

TCA behaves like the T command in respect of alarm and command output.

If neither TMMG nor TMMO are active, an alarm is generated.
Any alarms occurring during programming are handled in the same way as the alarms during T programming.

Notice

Offset selection, in accordance with $MC_CUTTING_EDGE_DEFAULT, acts in the same way as for the T command.
TCA and D must not be programmed in the same block.

Examples

1. **Preparation and change with T command (i.e. $MC_TOOL_CHANGE_MODE=0)**
   Configuration 1x turret, 1x toolholder
   There are two tools with identifier “Finish cutting” and duplo numbers 1 and 2.

   TCA(“Finish cutter”, 1,1)

   The tool “Finish cutter” with duplo number 1 is loaded onto toolholder 1 at change.

   With the machine configuration assumed above, the following programming would have the same result:

   TCA(“Finish cutter”)

   The duplo number is not specified, this means that the tool with the smallest duplo number is changed, i.e. duplo “1”.
   The toolholder no. is not specified. Therefore, the change is effective for the current master toolholder, i.e. “1”.

2. **Change with M6 ($MC_TOOL_CHANGE_MODE=1)**
   Configuration: 1x chain magazine, 2x spindles, spindle_1 is the master spindle.
   4 tools are loaded, “MILLER_20MM”, with duplo numbers 4, 5, 8 and 15.
   “MILLER_20MM”, duplo “8” was disabled and must be measured. Measuring takes place on spindle 2.

   TCA(“MILLER_20MM”,8,2)
   M2=6

   The tool “MILLER_20MM”, duplo “8” is prepared for spindle “2” and changed.

   Here the following programming would lead to a different result:

   TCA(“MILLER_20MM”)
   M6

   Tool “MILLER_20MM” with duplo “4” (smallest duplo number) was prepared for spindle “1” (this is the master spindle) and changed with M6.

Notice

The following particularities apply for TCA in comparison with T commands:

1. TCA and D cannot be programmed in the same block.
2. TCA renders the set search strategies ($TC_MAMP2 and/or $TC_MAP10) ineffective and ignores the programmed valued of $P_USEKT.
3. The tool must have status “available”.
4. The PLC interface signals “Transition to new replacement tool” and “Last replacement tool of group” are not set.
5. TCA cannot be substituted (T replacement cycle).

––> I.e. TCA cannot be used as an alternative to the T command.

PLC

The PLC is not allowed to refuse a tool prepared with “TCA”.

Caution: Currently the interface does not have criteria whether a tool is to be refused or not.
If you are working with this function, an additional identifier must be used to indicate this to the PLC.

Example:	$TC_VDITCP[2]=101	(101” identifier which the PLC is not allowed to refuse)
	TCA(“Miller”,1)	

5.8.24 TCI – Change tool from buffer into magazine

This function is available for TMMG.

The command TCI returns the tools from buffer locations back to the magazine. Toolholder locations are however excepted from this. Applications are for chain and box-type magazines.
The necessary empty-location search is carried out in the same way as for a programmed tool change with T (see Subsection 3.2.1).

Notice

TCI cannot be programmed together with M06 in one NC block. Tool change preparation and execution are carried out as one operation.

The TCI command cannot be substituted (T function replacement).

TCI(locNo, toolholder no.)

locNo	Number of the buffer with the tool that shall be returned to the magazine. Since the locNo cannot be the location number of a toolholder, returning the tool has no effect on active tool compensation.
Toolholder no.	This specifies the number of the toolholder from where disposal of the tool shall is to take place. This parameter is optional. If this parameters is not specified, then the current master toolholder is automatically selected.

Alarm 6403 is generated is an invalid location number is programmed.

The location number locNo is invalid

- if locNo indicates a toolholder / spindle (alarm 6450)
- if locNo indicates a non-defined buffer location (alarm 6403)
- if locNo is not linked with the programmed toolholder or master toolholder by $TC_MLSR (alarm 6454)
- if no distance table is defined either for the buffer locNo or the toolholder/ spindle (alarm 6454)

Alarm 6451 is generated if no buffer magazine has been defined.
Alarm 6452 is generated if the specified toolholder is not defined.
Alarm 6431 is generated if TMMG is not active.

It is necessary for programming TCI successfully that the specified location number locNo is assigned by $TC_MLSR to the toolholder. Empty locations are searched for in the magazines defined in the distance table (defined by $TC_MDP2) of the buffer locNo or of the toolholder. If both the buffer locNo and the toolholderhave a distance table, the buffer distance table is the one that is used. Alarm 6454 is generated if neither has a distance table.

Notice

The command TCI receives the location number of a location (gripper, loader, transfer point) of the buffer magazine as the parameter. The location number can be obtained by the the system variables $P_MAGNREL, $P_MAGREL in order to use this NC command in other cycle programs.

Example

Let us assume the following magazine configuration:

- Magazine1

is defined in the buffer magazine with 5 locations:

- Spindle 2 (location 1) with grippers 1 and 2 (locations 3 and 4 coupled with the spindle by $TC_MLSR[3,1] = 0 and $TC_MLSR[4,1] = 0)
- Spindle 1 (location 2) with gripper 3 (location 5 coupled with the spindle by $TC_MLSR[5,2] = 0)

The following is programmed:

TCI(2) ; generates alarm 6450
TCI(5) : changes the tool from location 5 (gripper 3) back to the magazine
TCI(9) : alarm 6403 (buffer only has the numbers 1 to 5

The user determines the sequence of disposal of the buffer locations by programming.

PLC

TCI is excuted in the PLC like the programming of T0 M06.
The buffer number transferred in the DB72 has to be evaluated.

5.8.25 GETFREELOC – Search for empty location

This function is available for TMMG.

For a given tool, search for an empty location in those magazines assigned to the specified loading location or the specified spindle / toolholder by an entry in the distance table. The strategy set by $TC_MAMP2 or $TC_MAP10 is used as the search strategy.

Defined location type hierarchies are taken into consideration when searching for an empty location by the PI service or for a programmed tool change.

Notice

GETFREELOC does not reserve the empty location that is found!

GETFREELOC(magNo&, locNo&, T no., refMag, refLoc)

magNo	> 0	Magazine number of the magazine where the empty location was found. Can also be used to specify the magazine number of the magazine in which the search is to take place.
	0	if no empty location was found
	–1	TMMG not active
	–2	invalid magazine number specified
	–3	invalid magazine location number specified The location number is also regarded as invalid if the magazine number is invalid.
	–4	invalid T number specified
	–5	invalid letter for “refMag”
	–6	if “refMag” = = “S”, invalid toolholder number “refLoc” specified if “regMag” = = “L”, invalid loading location number “refLoc” specified
locNo	> 0	Magazine location number of the empty location that was found Can also be used to specify the location number of the magazine with the nominated magNo that is to be checked to see if it can accept the nominated tool.
	0	if no empty location was found
	–1	TMMG not active
	–2	invalid magazine number specified
	–3	invalid magazine location number specified The location number is also regarded as invalid if the magazine number is invalid.

	–4	invalid T number specified
	–5	invalid letter for “refMag”
	–6	if “refMag” = = “S”, invalid toolholder number “refLoc” specified if “regMag” = = “L”, invalid loading location number “refLoc” specified
T No.	T number of the tool for which an empty location is to be searched. Thesearched location must be suitable for the tool size and the type of magazine location defined in the tool. If an invalid T number is programmed, parameters magNo, locNo each return the value –4.	
refMag	Reference magazine referred to for the empty location search (optional parameter). “S” = buffer magazine “L” = loading magazine “–” = no reference magazine. Is used if a magazine definitely has to be specified. If a value not equal to “S”, “L” is programmed, parameters magNo, locNo each return the value –5. If the nominated reference magazine is not yet defined, parameters magNo, locNo also return the value –5.	
refLoc	If refMag equals “S”, then the spindle number/toolholder number is specified here for empty location search. If an invalid toolholder number is programmed, parameters magNo, locNo each return the value –6. If refMag equals “L”, then the number of the location in the loading magazine is given for empty-location searching. This parameter is optional. If no parameters is programmed then the search for the master toolholder is carried out for refMag = “S”. When refMag = “L”, the search for location number = 1 is carried out in the loading magazine. When refMag = “–”, the parameter is not taken into account.	

Notice

If several parameters are incorrect, the value of magNo, locNo will depend on which parameter NCK checks first.

An alarm is generated if the TMMG is not active.

5.8.26 $P_USEKT – Tool change only with tools of subgroup

This function is only available for TMMO and TMMG.

This command selects a subset of a tool group which is then taken into account for the subsequent tool change.

The subgroups are set via system variables $TC_TP11[t] (see Section 5.3.1)

Name	**$P_USEKT**			
Meaning	$P_USEKT is a bit-coded value. Only the contents of the bits 0 – 3 are of significance. All tools having the parameter $TC_TP11 has set one of the bits of $P_USEKT are available in the following tool changes. The value 0 means that “all bits are set”. An alarm is generated if there is no such tool in a tool group for which a tool change was programmed.			
Data type	INT		Effective from SW version 4.2	
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	x	x	–	–
Implicit preprocessor stop	–	–		

It is only when $P_USEKT has been programmed that the selection takes effect. The selection is disabled by end of program or by a reset.

Bit coding makes it possible for a tool to belong to several tool subgroups. A maximum of 4 different tool subgroups is foreseen by the configuration of NCK, i.e. only the bits 0, 1, 2 and 3 are considered.

Notice

The system variable $TC_TP11 was not evaluated in NCK up to now. The value is automatically assigned 0. A check should be made in existing data blocks whether the values included here are suitable.

The programming $P_USEKT = 0 means that all tools of the tool group are considered in the tool selection.

The value $TC_TP11[t] = 0 means "The tool belongs to all defined tool subgroups". This assures compatibility with existing data blocks.

If working with the function T=location, $P_USEKT **cannot** be programmed.
$P_USEKT is set automatically at each tool change.
At Power ON, reset and end of program, $P_USEKT = 0 is set.

Example

The tool group "Miller_25" comprises 4 tools.
(The following applies: Tool_Change_Mode=1)

Miller_25	duplo 1	T_no. 1	TP11=1
Miller_25	duplo 2	T_no. 2	TP11=2
Miller_25	duplo 3	T_no. 3	TP11=4
Miller_25	duplo 4	T_no. 4	TP11=8

%MPF
. . .

T="Miller_25"	
M06	Every tool in this group can be loaded, as no selection has been made The search strategy that has been set prevails.
. . .	
$P_USEKT=2	
. . .	
T="Miller_25"	
M06	"Miller_25", duplo 2 loaded at change
. . .	
$P_USEKT=9	
. . .	
T="Miller_25"	
M06	"Miller_25", duplo 1 or duplo 4 loaded at change (depending on the search strategy set)
. . .	
$P_USEKT=0	
. . .	

T="Miller_25"
M06 Every tool in this group can be loaded, as USEKT=0 has canceled the selection. The search strategy that has been set prevails.

. . .
$P_USEKT=15
. . .
T="Miller_25"
M06 Every tool in this group can be loaded, as all bits are set. The search strategy that has been set prevails.

T=location, automatic tool selection

First the attempt is made to load the tool from the programmed magazine location, independently of the value in $TC_TP11.
If this tool is disabled, then the $TC_TP11 value of the tool at the programmed magazine location is considered in order to access the replacement tool. Only tools that have one of the bits of the disabled tool in $TC_TP11 can be replacement tools.

5.8.27 $A_TOOLMN – read magazine no. of tool

Note: TOOLMN stands for = "**tool m**agazine **n**umber". The name component $A_TOOL was selected to show the association with the existing system variables.

Name	**$A_TOOLMN[t]**			
Meaning	Returns the magazine number of the tool with T no.=t. If the tool is not assigned to a magazine, 0 is returned. If the tool management function is not active, –1 is returned. If there is no tool with T no.=t, –2 is returned. An alarm is issued if the value range for the T number was violated.			
Data type	INT		Effective from SW version 4.2	
Value range	1–32000			
Indices	Meaning			Value range
	The index specifies the T number			1–32000
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	x	–	x	–
Implicit preprocessor stop	x	–		

5.8.28 $A_TOOLMLN – read magazine location no. of tool

Note: TOOLMLN stands for = "**tool m**agazine **l**ocation **n**umber".

Name	**$A_TOOLMLN[t]**			
Meaning	Returns the magazine number of the tool with T no.=t. If the tool is not assigned to a magazine, 0 is returned. If the tool management function is not active, –1 is returned. If there is no tool with T no.=t, –2 is returned. An alarm is issued if the value range for the T number was violated.			
Data type	INT		Effective from SW version 4.2	
Value range	1–32000			
Indices	Meaning			Value range
	The index specifies the T number			1–32000
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	x	–	x	–
Implicit preprocessor stop	x	–		

Note: The following combinations are not possible: $A_TOOLMLN[t]==0 and $A_TOOLMN[t]>0 / $A_TOOLMLN[t]>0 and $A_TOOLMN[t]==0.

5.8.29 $P_TOOLND – Read number of cutting edges for tool

Note: TOOLND stands for = "**tool n**umber of **D**s".

Name	$P_TOOLND[t]			
Meaning	Returns the number of cutting edges for tool with T no.=t. A tool always has at least one cutting edge. Default: If there is no tool with T no.=t, –1 is returned. The value 0 is rejected as index error.			
Data type	INT		Effective from SW version 4.2	
Value range	Default: –1, 1 – 9 Function "flat D numbers": –1, 1 – "Machine data value for the maximum number of D numbers"			
Indices	Meaning			Value range
	The index specifies the T number			1–32000
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	x	–	–	–
Implicit preprocessor stop	–	–		

Function "flat D numbers" (only without active tool management)

If the function "flat D numbers" is active, the behavior differs somewhat. With parameter t=1 the total number of offset block records of the TOA unit is returned. Other values for t return –1. If no offset block record is defined in the TOA unit, –1 is returned.

5.8.30 $A_MONIFACT – Factor for reading tool life monitoring

If different tool materials are to be machined with the same tool, it may be necessary to increase or reduce the time intervals for monitoring in order to detect the varying degrees of tool wear. The factor is set accordingly before the tool is used. The write operation is performed synchronously with the main run.

A channel-specific parameter, used to multiply the current time measurement, has been defined.

Setting a value = 0 deactivates the time monitoring function for all tools used on the channel via the part program (see Subsection 3.9.2).

Name	$A_MONIFACT[t]			
Meaning	Only relevant when time monitoring is active in the tool management. Factor for influencing the time measurement for tracking time for time-monitored tools. Values < 1 and > 0 slow down time measurement (the clock "runs slower"). Values > 1 speed up the time measurement (clock "runs faster"). Value 1 is active after the control has been powered up, after Reset and M30 (default) and corresponds to real time. Value 0 is also permitted and disables time measurement of all time-monitored tools that are operated on a time-monitored spindle on this channel. Note: You can cause the monitoring time to "run backwards" by using negative values.			
Data type	REAL		Effective from SW version 4.2	
Value range	Value range of type REAL			
Indices	Meaning			Value range
				–
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	x	x	x	x
Implicit preprocessor stop	–	x		

Tool life counter on monitor

If system variable $A_MONIFACT is set accordingly, the tool life counter on the monitor can run at a speed other than real time. The time values of the OPI block TS are thereby converted at the interface (see Subsection 5.8.48). NCK-internally the values are retained as before. These values are real-time values.

Read OPI:	The time values are divided by the current value of $A_MONIFACT and transferred.
Write OPI:	The time values output by the OPI are multiplied by the current value of $A_MONIFACT and stored in the NCK.

Example

The current values are specified (units in real time, i.e. normalized to $A_MONI-FACT = 1).

Programmed tool life:	10 minutes
Actual tool life:	2 minutes – the prewarning limit will be reached in **one minute**
Prewarning limit:	1 minute

The values 10, 2, 1 are displayed on the screen.

$A_MONIFACT = 2 is programmed in the part program (clock runs faster). The actual tool life displayed on the monitor jumps and continues to run in real time. The programmed tool life and prewarning limit displayed also jump as soon as **$A_MONIFACT = 2** takes effect.

Programmed tool life	5 minutes
Actual tool life:	1 minute – the prewarning limit will be reached in **half a minute**
Prewarning limit:	0.5 minutes

5.8.31 $AC_MONMIN – Factor for tool search

The following is defined by the variable $AC_MOMIN:
Only consider those tools whose actual value is at least a factor $AC_MONMIN (0, ...1) of the set value away from the limiting value.

Definition of smallest/largest actual value

Absolute smallest/largest **actual values** are, in accordance with the tool-search strategy "Search for the tool with the smallest/largest actual value" used for the tool search exactly then when **all tools of a tool group have the same monitoring type defined** (via $TC_TP9).
This means all tools of the tool group are either time-monitored or count-monitored, or are wear or alternatively sum-offset-monitored.

Relative smallest/largest **actual values** are, in accordance with the tool-search strategies "Search for the tool with the smallest/largest actual value" used for the tool search exactly then when **the tools of a tool group have different monitoring types defined** in $TC_TP9.

This means one tool can be time-monitored, the other tool can be count-monitored. A third tool could be both wear as well as time-monitored.

Smallest/largest actual value for exactly one monitoring type

This is the standard application.
Each smallest/largest actual value here of the monitored variable ($TC_MOP2, $TC_MOP4, $TC_MOP6 for time, count, wear or additive offset) corresponds to smallest/largest actual value of the tools in the tool group.

Example:
A tool group “TL1” is defined. E.g. $TC_MAMP2=“H108” applies – smallest actual value:

Duplo no. $TC_TP1	Actual value $TC_MOP2	Set value $TC_MOP11	Absolute Smallest actual value = $TC_MOP2
1	9	10	9
2	8	10	8
3	6	6	6 Smallest actual value in the tool group

Therefore the order of tools for use is: Duplo no.= 3 –> 2 –> 1.

Smallest/largest actual value with several parallel monitoring types

Tools in a tool group can be monitored in different ways.
Or different types of tool monitoring can be defined for a tool. These situations are detected by the NCK and handled accordingly:

The definition of the smallest/largest actual value is determined for these cases by the product of dividing actual value and set value; i.e.

Quotient (Q)= actual value / setpoint

The tool with the smallest quotient has the **smallest actual value** of the tools in the tool group.

The tool with the largest quotient has the **largest actual value** of the tools in the tool group.

Example 1:

Tool group “milling machines” has two tools with T nos. =1 and 2 each with a cutting edge D1.
Time monitoring is active for T1; $TC_TP9[1]=1.
Workpiece count monitoring is active for T2; $TC_TP9[2]=2.

Q(T1) = $TC_MOP2[1,1] / $TC_MOP11[1,1] = 0.5
Q(T2) = $TC_MOP4[2,1] / $TC_MOP13[2,1] = 0.9

Therefore, T1 has the smaller actual value.

Example 2:

A tool group “TL1” is defined. E.g. $TC_MAMP2=“H108” applies – smallest actual value:

Duplo no. $TC_TP1	Actual value $TC_MOP2	Set value $TC_MOP11	Absolute Smallest actual value = $TC_MOP2
1	9	10	0.9
2	8	10	0.8 Smallest actual value
3	6	6	1

Therefore the order of tools for use is: Duplo no.=2–>1–>3.

$AC_MONMIN

The above definition of the actual value applies for the actual value that is checked against the set value given the factor $AC_MONMIN here.

The following check is made for the absolute actual-value comparison (time monitoring taken here as the example):
$TC_MOP2 >= $AC_MONMIN * $TC_MOP11.
This is the criterion for the usability of the tool.

The following check is made for the relative actual value comparison (time monitoring taken here as the example):
$TC_MOP2 / $TC_MOP11 >= $AC_MONMIN
This is the criterion for the usability of the tool.

The result is the same in each case.

Notice

The smallest of the actual values (both absolute as well as relative) of the cutting edges of a tool is used for the comparison with the actual values of other tools.

Name	$AC_MONMIN	
Meaning	Only when TOOLMAN function is active Gives the factor for the tool-search strategy “Only consider those tools whose actual value is at least a factor $AC_MONMIN* of the set value away from the limiting value. The programmed value is ignored if the tool status “disabled” shall be ignored during the tool search. This can be initiated either by the command TCA, PLC signal or machine data for start/reset. See also the system variables $TC_MOPx, $TC_MAMP2.	
Data type	REAL	in software version 6 and higher

Name	$AC_MONMIN			
Value range	0–1			
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	X	X	X
Implicit preprocessor stop	–	X	–	

Restrictions

If different types of tool monitoring are selected for the tools of a tool group, then the decision has to be made whether it is meaningful for the specific application to use the tool-search strategy "Search tool with smallest or largest actual value" in this tool group.
Similar conditions apply when working with multiple-edge tools. Also, it is necessary to consider whether it is meaningful to apply the tool search strategy "Search tool with smallest or largest actual value" in this tool group.

Notice

As for the other tool search strategies, that tool is preferred for use that is on the spindle or in one of the assigned buffer at the time of the tool search; i.e. the tool search strategy is not applied.

The PLC signal "Do not disable tool" renders the tool search strategy ineffective in accordance with $AC_MONMIN.

Activation

The following must apply so that the tool-monitoring-specific tool search strategies can be effective:

- The sub-function "Tool-monitoring function" must be active within the toolmanagement function,
- The appropriate monitoring parameter values ($TC_MOP1, ...) must have been set for the cutting edges of the tools.
- The monitoring must be activated for the appropriate tool (system variable $TC_TP9).
- $AC_MONMIN be programmed in the part program as well. The programmed value is only meaninful if points 1, 2, and 3 are met.

5.8.32 $P_TOOLNG – Number of tool groups

This function is only available for TMMO and TMMG.

Name	$P_TOOLNG			
Meaning	Number of defined tool groups that are assigned to the channel. > 0 Read access successful 0 no tool group defined (tool group is defined by writing the tool name) –1 neither function TMMG nor TMMO active			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

5.8.33 $A_MYMN / $A_MYMLN – Owner magazine/location of the tool

This function is available for TMMG.

System variables $A_TOOLMN (Section 5.8.27) and $A_TOOLMLN (Section 5.8.28) contain definitions which magazine/magazine location the specified tool is currently located at. This can be a real or an internal magazine.

The system variables $A_MYMN and $A_MYMLN indicate the magazine/magazine location (real magazine only), at which the specified tool was loaded or from which a tool contained in an internal magazine was loaded.

Name	$A_MYMN[t] / $A_MYMLN[t]			
Meaning	Application: $A_MYMN[t] – number of the owner magazine of the tool with the T no. = t > 0 Tool is loaded 0 Tool is not loaded –1 Function TMMG is not active –2 Tool with the T no. = t does not exist – not for t = 0 either $A_MYMLN[t] – number of the owner-magazine location of the tool with the T no. = t > 0 Tool is loaded 0 Tool is not loaded –1 Function TMMG is not active –2 Tool with the T no. = t does not exist – not for t = 0 either			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	X	–
Implicit preprocessor stop	–	–	–	

The following applies for tools that are not loaded:
$A_MYMN = $A_MYMLN = $A_TOOLMN = $A_TOOLMLN = 0

The following applies for manual tools that were loaded at change and for tools newly loaded on toolholders:
$A_MYMN = $A_MYMLN = 0
$A_TOOLMN != 0, $A_TOOLMLN != 0

The following applies for tools that have been loaded but are not contained in an internal magazine:
$A_MYMN = $A_TOOLMN > 0
$A_MYMLN = $A_TOOLMLN >0

For fixed-location-coded tools in buffers, the two parameters indicate where the respective tool shall be brought when returning to the magazine.

5.8.34 $P_TOOLNT / $P_TOOLT – T numbers

This function is available for TMMO, TMMG, TMFD and TMBF.

Name	$P_TOOLNT / $P_TOOLT[i]			
Meaning	These system variables enables an overview of the tools defined in NCK. $P_TOOLNT Number of defined tools that are assigned to the channel. > 0 Read access successful 0 No tool defined $P_TOOLT[i] i-th tool number T > 0 T number 0 i is a value outside of the permitted range			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	N = number of tools that are assigned to the channel i = i-th T no.; i=1, ..., $P_TOOLNT			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

The following applies specifically for TMFD:
$P_TOOLNT returns value 1 provided D offsets have been defined and returns value 0 when there are no D offsets.
The system variable $P_TOOLT returns value 1 for index i = 1 if at least one D offset has been defined and returns value 0 for other values of i.

5.8.35 $P_TOOLD – D numbers

This function is available for TMMO, TMMG, TMFD and TMBF.

Name	$P_TOOLND / $P_TOOLD[t,i]			
Meaning	Determine the defined D numbers of a tool. The command can generally be programmed. i-th number of tool compensations D of the tool with the T no. = t > 0 D number 0 i is a value outside of the permitted range –2 t is the value of a non-defined tool			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	t = T number i = i-th T no.; i=1, ..., $P_TOOLND			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

5.8.36 $P_TOOLNDL – Number of defined DL offsets

This function is available for TMMO, TMMG, TMFD and TMBF. The function “additive offset” must have been activated via MD.

Name	$P_TOOLNDL[t,d]		
Meaning	Determine the number of defined DL numbers of a D offset. The command can generally be programmed. Number of DL offsets for D offset provided by T no. = t, D no. = d > 0 Numer of DL offsets 0 No DL offsets for this D offset –1 Additive offset function not active –2 t is the value of a non-defined tool –3 d is the value of a non-defined D offset		
Data type	INT	in software version 6 and higher	
Value range	1–32000		
Indices	Meaning		Value range
	t = T number d = D number		

Name	$P_TOOLNDL[t,d]			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

5.8.37 $A_USEDND – Workpiece count

This function is available for TMMO.

Name	$A_USEDND[s]			
Meaning	Number of different cutting edges used on toolholder s since the last workpiece count including the currently used active cutting edge. Each tool used is included at least once. Index s means: <u>TMMG + TMMO</u> Spindle number/toolholder number s = 0 means that the currently active master toolholder is selected. <u>TMMO active without TMMG</u> a) $MC_T_M_ADDRESS_EXT_IS_SPINO = FALSE: s is not evaluated. It is not possible to count the workpieces separately according to toolholders. b) $MC_T_M_ADDRESS_EXT_IS_SPINO = TRUE: s = 0 means that the currently active master toolholder is selected. > 0 Number of adapter cutting edges used. 0 No more tools used since the last workpiece count –1 TMMO is not active –2 s is the value of a non-defined toolholder			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	s = 1, ..., MAXNUM_AXES_PER_CHAN s = 0 indicates the master toolholder			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Examples see Section 5.8.38.

5.8.38 $A_USEDT – Workpiece count

This function is available for TMMO.

<table>
<tr><th>Name</th><th colspan="4">$A_USEDT[i,s]</th></tr>
<tr><td>Meaning</td><td colspan="4">T number of the tool of the i-th cutting edge of cutting edges that have been used on the toolholder s since the last workpiece count or are still being used. The index s means:
<u>TMMG + TMMO</u>
Spindle number/toolholder number
s = 0 means that the currently active master toolholder is selected.
<u>TMMO active without TMMG</u>
a) $MC_T_M_ADDRESS_EXT_IS_SPINO = FALSE: s is not evaluated. It is not possible to count the workpieces separately according to toolholders.
b) $MC_T_M_ADDRESS_EXT_IS_SPINO = TRUE: s = 0 means that the currently active master toolholder is selected.
>0 T number (can also exist multiple times if different D offsets of the tool were in use)
0 No more tools used since the last workpiece count
–1 TMMO is not active
–2 s is the value of a non-defined toolholder</td></tr>
<tr><td>Data type</td><td colspan="2">INT</td><td colspan="2">in software version 6 and higher</td></tr>
<tr><td>Value range</td><td colspan="4">1–32000</td></tr>
<tr><td>Indices</td><td colspan="3">Meaning</td><td>Value range</td></tr>
<tr><td></td><td colspan="3">s = 1, ..., MAXNUM_AXES_PER_CHAN
i = 1, ..., $A_USEDND</td><td></td></tr>
<tr><td>Access</td><td>Read in part program</td><td>Write in part program</td><td>Read in synchronous action</td><td>Write in synchronous action</td></tr>
<tr><td></td><td>X</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>Implicit preprocessor stop</td><td>–</td><td>–</td><td>–</td><td></td></tr>
</table>

Example

Two toolholders are defined with numbers 1 and 2. Toolholder no. 1 is the master toolholder. On toolholder 1 up to now 3 tools with T numbers 10, 20, 30 were used; on toolholder 2 one tool was used with T number 666. Each tool only has offset D1 defined.

The following program section is run in the status:

```
def int n1, n2, i, tNo
n1 = $A_USEDND[1]            ; n1 = 3 same content would have been
                             ; $A_USEDND[0]
n2 = $A_USEDND[2]            ; n2 = 1
for i = 1 to n1
    tNo = $A_USEDT[1,i]
    MSG ("to T no. participating in workpiece machining =" << tNo
    endfor
                             ; The loop displays T numbers 10, 20, 30
T2=0                         ; Bits 7, 8, 19 are set for synchronization in
                             ; $MC_TOOL_MANAGEMENT_MASK. (auto-
                             ; matic read-in disable until tool change is ac-
                             ; knowledged with "End".)
setpiece(5,2)
if (n2 == 1) tNo = $A_USEDT[1,1]
                             ; sets tNo to value 0. setpiece was programmed
                             ; since determination of n2. This deletes the list of
                             ; tools used and there is currently no entry for the
                             ; specified Index1 in the list of tools used.
```

5.8.39 $A_USEDD – Workpiece count

This function is available for TMMO.

Name	**$A_USEDD[i,s]**			
Meaning	D number of the i-th cutting edge of cutting edges that have been used on the toolholder s since the last workpiece count or are still being used. The index s means: <u>TMMG + TMMO</u> Spindle number/toolholder number s = 0 means that the currently active master toolholder is selected. <u>TMMO active without TMMG</u> a) $MC_T_M_ADDRESS_EXT_IS_SPINO = FALSE: s is not evaluated. It is not possible to count the workpieces separately according to toolholders. b) $MC_T_M_ADDRESS_EXT_IS_SPINO = TRUE: s = 0 means that the currently active master toolholder is selected. >0 D number 0 No more tools used since the last workpiece count –1 TMMO is not active –2 s is the value of a non-defined toolholder			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	s = 1, ..., MAXNUM_AXES_PER_CHAN i = 1, ..., $A_USEDND			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

5.8.40 $P_MAGN / $P_MAG – Magazine

This function is available for TMMG.

Name	**$P_MAGN / $P_MAG[i]**			
Meaning	$P_MAGN Number of defined magazines that are assigned to the channel. > 0 Read access successful 0 no magazine defined –1 TMMG is not active $P_MAG i-th magazine number > 0 Read access successful 0 i is outside of the permitted range –1 TMMG is not active			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	i = 1, ..., $P_MAGN			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Example

See Chapter 5.8.44.

5.8.41 $P_MAGNDIS / $P_MAGDISS / $P_MAGDISL – Magazine distance tables

This function is available for TMMG.

Name	**$P_MAGNDIS[n,m] / $P_MAGDISS[l,i] / $P_MAGDISL[l,i]**			
Meaning	$P_MAGNDIS**[n,m]** Number of magazines that are linked to the internal magazine n by the location m > 0 Read access successful 0 no magazine linked with the buffer –1 TMMG is not active –2 n is not the number of an internal magazine –3 m is not the number of an internal magazine location $P_MAGDISS[l,i] Number of the i-th magazine that is linked with location l of the buffer magazine. > 0 Read access successful 0 i is outside of the permitted range –1 TMMG is not active –2 m is not the number of a buffer location –3 No buffer location defined $P_MAGDISL[l,i] Number of the i-th magazine that is linked with location l of the loading magazine. > 0 Read access successful 0 i is outside of the permitted range –1 TMMG is not active –2 m is not the number of a loading magazine location –3 No buffer location defined			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	i = 1, ..., $P_MAGNDIS			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Example

See Chapter 5.8.44.

5.8.42 $P_MAGNS / $P_MAGS – Toolholder

This function is available for TMMG.

Name	$P_MAGNS / $P_MAGS[n]			
Meaning	$P_MAGNS Number of spindle locations/toolholder locations in the buffer assigned to the channel. > 0 Read access successful 0 no spindle location defined –1 TMMG is not active –2 No buffer magazine defined. $P_MAGS[n] n-th number of the spindle/toolholder in the buffer > 0 Read access successful 0 n is outside of the permitted range –1 TMMG is not active –3 No buffer magazine defined.			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	n = 1, ..., max. toolholder number			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Example

See Chapter 5.8.44.

5.8.43 $P_MAGNREL / $P_MAGREL – Assigned buffer

This function is available for TMMG.

Name	$P_MAGNREL[n] / $P_MAGREL[n,m]			
Meaning	$P_MAGNREL[n] Number of buffers assigned to the spindle no./toolholder no. > 0 Read access successful 0 spindle location has not buffer location assigned –1 TMMG is not active –2 n is not the number of a spindle location –3 No buffer magazine defined. $P_MAGREL[n,m] m-th buffer number of the n-th spindle no./toolholder no. > 0 Read access successful 0 m is outside of the permitted range –1 TMMG is not active –2 n is not the number of a spindle location –3 No buffer magazine defined.			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	m = 1, ..., $P_MAGNREL n = 1, ..., max. toolholder number			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Example

See Chapter 5.8.44.

5.8.44 Example of magazine configuration system variables

Let us assume the magazine configuration displayed in the diagram below. Information about the current magazine configuration can be obtained by reading the system variables described here.

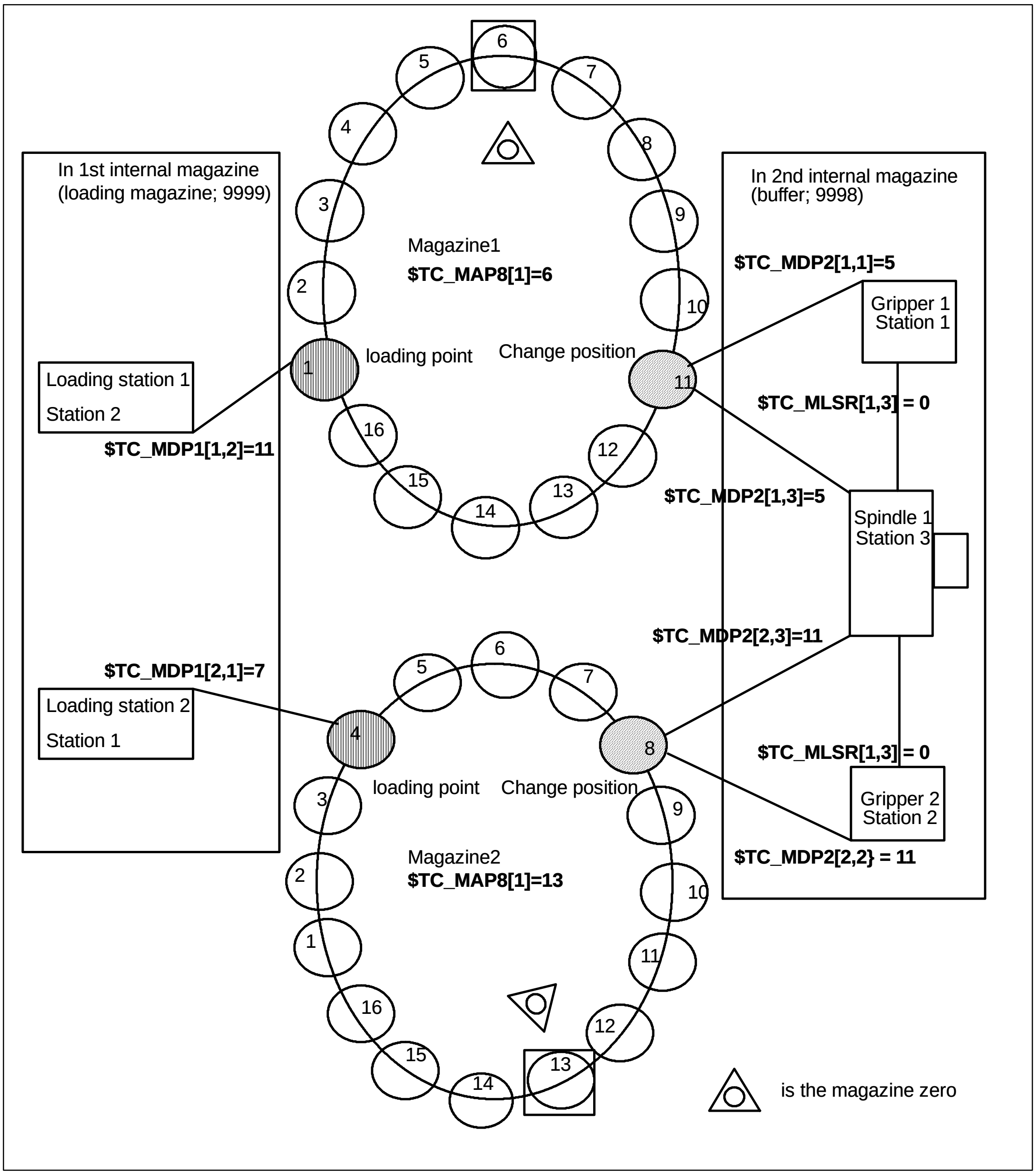


Bild 5-10 Magazine configuration

```
N10 def int noOfMag=0, noOfLoc=0, noOfDist=0, noOfRel=0,
    noOfSpindles=0, spindleNo=0
N20 def int i=0
```

;total number of defined magazines

```
N100 noOfMag = $P_MAGN      ; noOfMag is assigned value =4 – 2 real maga-
                            ; zines
                            ; 1, 2+2 internal magazine 9998, 9999
```

;display all magazine number

```
N200 for i=1 to noOfMag
N220 MDG (“Magazine no.=”<<$P_MAG[i])
                            ; Display numbers 1, 2, 9998, 9999
N240 endfor
```

;Total number of defined magazine locations

```
N300 for i=1 to noOfMag
N320 noOfLoc=noOfLoc + $TC_MAP7[$P_MAG[i]]
N340 endfor                 ; noOfLoc is now assigned value 16+16+3+2=37
```

;Number of magazines linked with Spindle 1

```
N400 noOfDist=$P_MAGNDIS[9998,3]
                            ; noOfDist is assigned value=2 – Mag.1, 2 are
                            ; linked with the spindle location
```

;Display the magazine numbers linked with Spindle 1 (=location 3)

```
N500 for i=1 to noOfDist
N520 MSG (“Magazine no.=” << $P_MAGDISS[ i ] )
                            ; Display numbers 1, 2
N540 endfor
```

; Number of magazines linked with Loading station 2

```
N410 noOfDist = $P_MAGNDIS[9999,1]
                            ; noOfDist is assigned value=1 – Mag. 2 is linked
                            ; with Loading station 2
```

;Display the magazine numbers linked with Loading station 2 (=location 1)

```
N510 for i=1 to noOfDist
N530 MSG (“Magazine no.=”<< $P_MAGDISL[i] )
                            ; Display number 2
N550 endfor
```

;Total number of defined spindles

```
N600 noOfSpindles=$P_MAGNS  ; noOfSpindles contains value = 1
                            ; – one spindle location is defined
```

;display the numbers of the spindles defined in the magazine configuration

```
N620 for i=1 to noOfSpindles
N640 MSG (“Magazine no.=”<< $P_MAGS[i])
                            ; Display number 1
N660 endfor
```

;Total number of buffer locations assigned to Spindle 1
(=gripper in example)

```
N700 noOfRel=$P_MAGNREL[1]
                                         ; noOfRel contains value=2 grippers 1 and 2 are
                                         ; assigned to the spindle
```

; display the numbers of the grippers defined in the magazine configuration for spindle no. 1

```
N720 for i=1 to noOfRel
N740 MSG (“Magazine no.=”<< $P_MAGREL[1,i] )
                                         ; Display numbers 1, 2
N760 endfor
```

5.8.45 $P_MAGNH / $P_MAGNHLT / $P_MAGHLT – Location type hierarchies

This function is available for TMMG.

Name	**$P_MAGNH / $P_MAGNHLT[n] / $P_MAGHLTn,m]**			
Meaning	$P_MAGNH Number of defined magazine location type hierarchies that are assigned to the channel. > 0 Read access successful 0 No location type hierarchies are defined –1 TMMG is not active $P_MAGNHLT[n] Number of the defined location types in the n-the defined hierarchy > 0 Read access successful 0 n is outside of the defined range –1 TMMG is not active $P_MAGHLTn,m] m-th location type of hierarchy n > 0 Read access successful 0 m is outside of the defined range –1 TMMG is not active –2 Hierarchy n does not have any defined location types			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	n = 1, ..., $P_MAGNH m = 1, ..., $P_MAGNHLT			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocess or stop	–	–	–	

Example

The following three hierarchies are defined

Hierarchy 1: 5 < 4 < 3:

$TC_MPTH[0,0] = 5
$TC_MPTH[0,1] = 4
$TC_MPTH[0,2] = 3

Hierarchy 2: 7 < 8:

$TC_MPTH[1,0] = 7
$TC_MPTH[1,1] = 8

Hierarchy 3: 1 < 2 < 9 < 6:

$TC_MPTH[2,0] = 1
$TC_MPTH[2,1] = 2
$TC_MPTH[2,2] = 9
$TC_MPTH[2,3] = 6

We want to know how many hierarchies in total are defined and how many magazine location types are contained in each hierarchy.

```
N10 def int noOfH, noOfTypes[8], locTypeNo
N100 noOfH=$P_MAGNH                     ; noOfH is assigned value = 3

N220 for i=1 to noOfH
N240 noOfTypes[i - 1]=$P_MAGNHLT[ i ]
                                        ; set the values 3, 2, 4 in the array
N260 endfor
```

Furthermore, we want to know which magazine location types are defined in the 2nd hierarchy

```
N220 for i=1 to noOfTypes[1]
N240 MSG ("Magazine no.="<<$P_MAGHLT[2, i])
                                        ; display values 7, 8
N260 endfor
```

5.8.46 $P_MAGNA / $P_MAGA – Tool adapter

This function is available for TMMG.

Name	$P_MAGNA / $P_MAGA[i]			
Meaning	$P_MAGNA Number of defined adapters that are assigned to the channel. > 0 Read access successful 0 no adapters defined –1 Function “Adapter” or TMMG is not active $P_MAGA[i] i-th adapter number > 0 Read access successful 0 i is outside of the defined range –1 Function “Adapter” or TMMG is not active			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	i = 1, ..., $P_MAGNA			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

5.8.47 Additional language commands

Name	$P_TOOLNO			
Meaning	Active tool numbers T0 to T32000, T can take eight digits with TMFD			
Data type	Integer		in software version 2 and higher	
Value range	1–32000			
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Name	$P_TOOLP			
Meaning	Tool number last programmed Command is available for TMBF, TMFD and TMMO. It is analogous to the TMMG-specific command GETSELT.			
Data type	Integer		in software version 5.3 and higher	
Value range	1–32000			
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Name	$P_TOOL			
Meaning	Active tool cutting edge (Dx)			
Data type	Integer		in software version 2 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Name	$P_DLNO			
Meaning	Active additive offset DL=0–DL=max; max=value of $MN_MM_MAX_SUMCORR_PER_CUTTEDGE $P_DLNO is analogous to the already existing parameters $P_TOOL, $P_TOOLNP and active D and T numbers.			
Data type	Integer		in software version 5.3 and higher	
Value range	0–6			
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–		

Name	$P_TOOLL[n]			
Meaning	Active tool total length; n = 1...3			
Data type	REAL		in software version 2 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Name	$P_TOOLR			
Meaning	Active radius			
Data type	REAL		in software version 2 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Name	$P_TC			
Meaning	Active toolholder			
Data type	Integer		in software version 5.3 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Name	$P_TCANG[n]			
Meaning	Active angle of a toolholder axis; n = 1–2			
Data type	REAL		in software version 5 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Name	$P_TCDIFF[n]			
Meaning	Difference between calculated and used angle of a toolholder axis for the matrix (Hirth tooth system) of the angle			
Data type	REAL		in software version 5.3 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Name	$P_AD[n]			
Meaning	Active tool offset; n = 1...31 n=1–25 $TC_DP1 to $TC_DP25 n=26 $TC_DPCE (optional) n=27 $TC_DPH (optional) n=28 $TC_DPV (optional) n=29 $TC_DPV3 (optional) n=30 $TC_DPV4 (optional) n=31 $TC_DPV5 (optional)			
Data type	DOUBLE		in software version 2 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	X	–	–
Implicit preprocessor stop	–	–	–	

$P_ADT[n] – Transformed data of the active tool (SW 6)

For the functional description, refer to Subsection 3.11.3.

Name	$P_ADT[n]			
Meaning	When compensation parameters are read, this parameter returns transformed values of the parameters controlled by the tool adapter transformation – if the active tool is attached to an adapter. n=1–25 $TC_DP1 to $TC_DP25 n=26 $TC_DPCE (optional) n=27 $TC_DPH (optional) n=28 $TC_DPV (optional) n=29 $TC_DPV3 (optional) n=30 $TC_DPV4 (optional) n=31 $TC_DPV5 (optional)			
Data type	DOUBLE		in software version 6 and higher	
Value range				
Indices	Meaning			Value range
	n: Parameter numbers 1 to 31			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Name	$AC_MSNUM			
Meaning	Master spindle, return value 0: No spindle configured 1...n: Number of the master spindle			
Data type	Integer		in software version 3 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	X	–
Implicit preprocessor stop	X	–	–	

Name	$P_MSNUM			
Meaning	Master spindle 0: No spindle configured 1...n: Number of the master spindle			
Data type	Integer		in software version 5.2 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–	–	

Name	$AC_MTHNUM			
Meaning	Master toolholder Value=0 if no master toolholder defined Value>0 number of the master toolholder			
Data type	Integer		in software version 5 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	X	–
Implicit preprocessor stop	X	–	X	–

Name	$P_MTHNUM			
Meaning	Master toolholder Value=0 if no master toolholder defined Value>0 number of the master toolholder			
Data type	Integer		in software version 5.3 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	–	–
Implicit preprocessor stop	–	–		

5.8.48 Variables for subroutine replacement technique

Tool language command	Functions	SW version
$C_T	Number of T word (without TOOLMAN) for substitute subroutine for T (MD 10717)	5
$C_T_PROG	Bool variable: contents in $C_T?	5
$C_TS	Programmed TL identifier (with TOOLMAN) for substitute subroutine for T (MD 10717)	5
$C_TS_PROG	Bool variable: contents in $C_TS?	5
$C_TE	Address extension of the T word	5.3
$C_D	Number of the D buffer	5.3
$C_D_PROG	Bool variable: contents in $C_D?	5.3
$C_DL	Number of the DL buffer	5.3
$C_DL_PROG	Bool variable: contents in $C_DL?	5.3

5.8.49 Variables for tool change in synchronized action

Tool language command	Functions	SW version
$AC_TC_FCT	Command number 1: Move (load, unload,...) 2: Prepare change 3: Change ON 4: Change ON (turret, without M06) 5: Prepare change and change ON (with M06)	5
$AC_TC_STATUS	Acknowledgment status of PLC FC8	5
$AC_TC_THNO	Number of the toolholder or the spindle where the new tool shall be loaded	5
$AC_TC_TNO	The internal T number of the tool to be loaded at change 0: there is no new tool	5
$AC_TC_MFN	Source magazine number of the new tool 0: there is no new tool	5
$AC_TC_LFN	Source location number of the new tool 0: there is no new tool	5
$AC_TC_MTN	Target magazine number of new tool 0: there is no new tool	5
$AC_TC_LTN	Target location number of the new tool 0: there is no new tool	5
$AC_TC_MFO	Source magazine number of the old tool 0: there is no old tool	5
$AC_TC_LFO	Source location number of the old tool 0: there is no old tool	5
$AC_TC_MTO	Target magazine number of old tool 0: there is no old tool	5
$AC_TC_LTO	Target location number of the old tool 0: there is no old tool	5
$AC_TC_CMDT	Trigger variable to NCK command output Set for one interpolation cycle when NCK outputs a new command.	6.3
$AC_TC_CMDC	Counter for NCK command output This variable is incremented by 1 at each NCK command output. Can also be written (zero setting).	6.3

Tool language command	Functions	SW version
$AC_TC_ACKT	Trigger variable to PLC command Set for one interpolation cycle when PLC outputs a command to NCK. Command acknowledgement or independent message (asynchronous transfer).	6.3
$AC_TC_ACKC	Counter for PLC commands This variable is incremented by 1 at each PLC command. Can also be written (zero setting).	

Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	–	X	–
Implicit preprocessor stop	X	–		

5.9 Conventions for programming data

5.9.1 Tool and cutting edge data

If a parameter for a cutting edge, tool or magazine that does not exist is written, a new cutting edge, tool or magazine is created.

Notice

When a tool is created, all the cutting-edge-specific data of cutting edge D1 are created with it.
(DP, DPC, MOP, MOPC are preset to “0”). The grinding-specific tool data ($TC_TG1...) is not created until one of the tool types ($TC_DP1) 400–499 has been programmed for one of the cutting edges of the tool.

Deleting data

When data is deleted the memory area is deleted with it and automatically released again.

A tool can only be deleted if it is not involved in the current machining process. This applies both to tools selected or inserted with a “T” call and tools for which constant grinding wheel surface speed or tool monitoring is active.

Notice

If tool management is active you must ensure that the tool being deleted is not assigned to a magazine location ($TC_MPP6). This assignment must be removed before the tool is deleted.

The grinding-specific tool data ($TC_TG1...) is created as soon as one of the tool types ($TC_DP1) 400–499 has been programmed for any of the cutting edges of the tool.

If the tool type is set from the current value taken from the range 400–499 to a value outside this range, the grinding data memory is enabled again, i.e. the grinding-specific data is lost.

Action	Program command	Description
Create a tool	Without tool management: $TC_DPx[y,z] = value;	Create tool T if T does not yet exist! y= T number z = D number
	With active tool management: T_NR = NEWT(“tool identifier”, duplo number) Or $TC_TP1[y] = duplo number; $TC_TP2[y] = “tool identifier”;	y= T number
Create a cutting edge	$TC_DPx[y,z] = value	Create cutting edge D = z if D = z does not yet exist! y= T number z = D number
Set tool data	With active tool management: $TC_TPx[y] = value; Or $TC_TPx,[GETT(“DRILL”,DUPLO NO)] = value Or $TC_TPCx[y] = value; $TC_TGx[y] = value;	y = T number Write tool-related user data Write tool-related grinding data
Set data of a cutting edge	$TC_DPx[y,z] = value $TC_DPCx[y,z] = value $TC_MOPx[y,z] = value $TC_MOPCx[y,z] = value	Write compensation data Write cutting edge-related user data Write cutting edge-related monitoring data Write CC (OEM) cutting edge monitoring data y = T number z = D number
Delete cutting edge data	Without tool management: $TC_DP1[0,0] = 0;	All tools of the channel are deleted, the memory is released.
	With tool management: $TC_TP1[0,0];	When deleting tools, the entries for the location data must also be corrected.

Action	Program command	Description
Delete tool data	Without tool management: $TC_DP1[y,0] = 0;	y = T is deleted, memory is released.
	With tool management: $TC_TP1[y] = 0; Or $TC_TP1[GETT("tool identifier", duplo number)] = 0; Or DELT["tool identifier", duplo number]	All tool-related data is set to "0" (user data, hierarchy data, ...). When deleting a tool, the entries for the location data must also be corrected.
Delete data for all tools	Without tool management: $TC_DP1[0,0] = 0;	All tools of the channel are deleted and the memory is released.
	With tool management: $TC_TP1[0,0] = 0;	When deleting tools, the entries for the location data must also be corrected.

5.9.2 Magazine data

Sequence for defining data

The process: "Assign tool to a magazine location" creates a codependency between the tool data and the magazine/magazine location data.

Example:

The tool contains the magazine location type for which it is intended. The magazine type contains its own magazine location type. If the tool is assigned to the magazine location, as a rule the location type cannot be changed again as this can cause inconsistencies.

The resulting requirement is for tools and magazines to be loaded by a special routine into the PLC and that the structure-determining definitions may no longer be changed during the preparation (these are e.g. magazine dimension, magazine-location type, duplo no., tool name, ...). They do not include: cutting edge data, magazine location status, tool status, ...

Load data

Because tools are linked to magazines via magazine location parameter $TC_MPP6, the following rules for correct definition of tools and magazines must be adhered to:

1. Load tool data
2. Load magazine data
3. Load $TC_MPP6 parameters (=> place tool in magazine location)

The same sequence is used for data backup.

The grinding data of a tool cannot be written until tool type = "grinding tool" has been defined for at least one cutting edge.

The distance parameter ($TC_MDPx) and the buffer assignment parameter ($TC_MLSR) cannot be written until the magazines and their locations have been defined.

Delete data

A tool cannot be deleted while it is still contained in a magazine. The following sequence of operations must be followed when deleting:

1. Delete the magazine data (this removes tools from the magazine); or remove the tool explicitly from the magazine.
2. Delete tool data

In addition, a magazine cannot be deleted if it has status $TC_MAP3[i]= 8 (motion is active). The delete command is rejected for all magazines even if only **one** magazine is preventing the command from being executed.

Notice

If a single tool is to be deleted it must first be removed from the magazine location with an unload operation and then it can be deleted.

Tools that are currently selected cannot be deleted! You can ensure that no tool is selected beyond a part program by programming T0 before the end of a part program independently of the settings in the machine data (see MD for selecting tools beyond the end of a program).

Action	Program command	Description
Create new magazine	$TC_MAPx[y]= value;	Value <>0, y = magazine no. of a magazine not yet set up
Delete a magazine	$TC_MAP1[y] = 0;	The data of the magazine and its magazine locations as well as any defined distances to change positions are deleted. The associated memory is released.
Delete a magazines and the tools contained in it	$TC_MAP6[y] = 0;	The data of the magazine and its magazine locations as well as any defined distances to change positions are deleted. Any tools contained in the magazine are also deleted. The associated memory is released.
Delete all magazines	$TC_MAP1[0] = 0;	All data of all magazines of the selected TO area unit is deleted and the associated memory is released again. The magazine data block is then empty.
Create new magazine location	$TC_MPPx[y,z]=value;	Value <>0 , y = location number not yet available. Before the data of the first location can be created, the associated magazine must be defined. When the first parameter of the first magazine location to be set up is written, then all magazine locations belonging to the magazine are set up using their default values in accordance with the requirements for number of lines and columns for the magazine.
Set magazine location type hierarchy	$TC_MPTHx[y]=value;	
Set magazine distances (distance to change position)	$TC_MPTHx[y]=value;	
Delete magazine distances (distance to change position)	$TC_MDPx[y,0]=0 $TC_MDPx[0,0]= 0;	Delete all defined distances of the magazine with the number “y”, i.e. the magazine is no longer “seen” during a tool search and an empty location search. Delete all defined distances of all magazines of the TO unit.
Delete the assignments of the buffer to the spindles	$TC_MLSR[x,0]= 0; $TC_MLSR[0,0]= 0;	Delete all defined assignments of a buffer location with the number “x”, i.e. the magazine is no longer “seen” during the tool search. Delete all defined assignments of buffer locations of the TO unit to spindles
Set magazine block data	$TC_MAMPx = value;	

5.9.3 Tool Change

Programming the tool selection

Tool selection can be divided into two different steps:

1. Tool change preparation
2. Tool change execution

Steps 1–2 can be programmed separately or together in the NC program (see MD 22550 TOOL_CHANGE_MODE).

Examples

Tool change in one step: (turret)

Tx; Make new tool x available and execute tool change

Tool change in two steps:

1. Tx; Prepare tool change (select the tool)
2. M06; Execute tool change

Notice

If tool management is active, a tool can only be selected with the tool identifier (name). If a T number is now programmed, then the number is used as the identifier (name). The tool must then have received a T number a name during loading.

Tool change with identifier:

T="DRILL"; A search is performed for tool with identifier "DRILL".

Tool change with number as identifier:

T="123"; A search is performed for tool with identifier "123". Alternatively, T123 can also be programmed.

5.9.4 Cutting edge selection

Cutting edge selection after tool change

When a tool change has been completed, the tool cutting edge can be selected in one of the following ways:

1. The offset number D is programmed.

2. The offset number D is not programmed and is preset by MD20270 CUTTING_EDGE_DEFAULT.
   = 0 No automatic cutting edge selection after M06.
   > 0 Number of cutting edge selected after M06.
   = –1 The cutting edge no. of the old tool remains valid and also selected for the new tool after M06.
   = –2 The offset of the old tool remains valid and also selected for the new tool after M06.

Examples:

Tool selection with the following cutting edge selection
Cutting edge selection always refers to the tool that is changed with command M06.

T1 M06	Tool change – no D programmed; therefore offset selection according to MD 20270
T5	Tool preselection
X .. Y.. Z...	Working with T1 and the offset from MD 20270
D2	Offset D2 from T1 !!!
M06	Tool change; T5 is loaded at change – offset selection according to MD 20270
T1	Tool preselection
X.., Y...	Working with T5 and the offset from MD 20270

When programming tool commands, main spindles and secondary spindles are programmed differently. Only tool offset values of the main spindle tool are taken into account by the geometry because only one active offset can be processed per channel. Processing of tool commands for a secondary spindle is only relevant for signal output to the PLC and the function GETSELT(...).

Spindle no. 2 = main spindle:

T2 = “DRILL”	
M2 = 06	
T1 = “MILLER”	Select tool for secondary spindle
M1 = 06	Tool change: load tool in the secondary spindle
D1	Select cutting edge of “DRILL” (main spindle)

Spindle no. 2 = main spindle:

T2 = “DRILL”	Selection a tool for the main spindle. As an alternative, T= “DRILL” could also be specified.
T1 = x;	Selection a tool for a secondary spindle.
M2 = 06	Tool change As an alternative, M06 could also be specified.
D1	Select cutting edge of a tool with identifier “DRILL”

5.9.5 Tool transfer from program test mode

In MD 20110 RESET_MODE_MASK, **bit 3** you can set whether the active tool and tool offset are to be taken

- (= 1) from the test program which was last terminated in test mode

  or

- (= 0) from the program which was last terminated before the test program was activated.

Requirement: Bits 0 and 6 must be set in MD 20110.

$P_ISTEST

The system variable $P_ISTEST is for checking from the part program whether a program test is active. The system variable returns the value TRUE when program testing is active.

5.10 Programming T=location number

This function is only available when tool management is active. This type of programming is not only suitable for turrets, but for all other types of magazine.

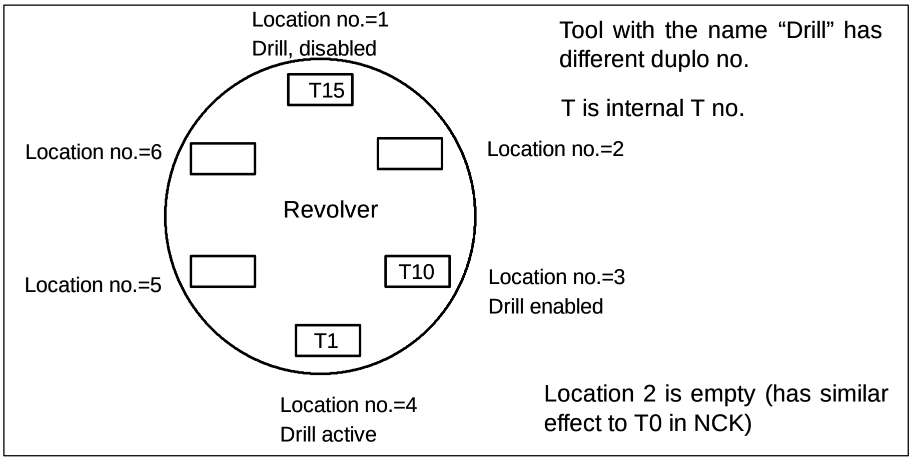


Bild 5-11 Programming of T=location number

The machine data MD 20310: TOOL_MANAGEMENT_MASK, bit16=1 is used for setting the programming method:

- T = “**x**” with **x** as tool **identifier**
- Tx, with **x as location number** of the magazine containing the tool used for machining

When the function is active, **T1** selects the tool in location number 1 instead of the tool with identifier “1”. The first magazine linked with the toolholder is taken here. The identifier of the tool in this location is then determined (“Drill”).

The subsequent procedure is as if T=“Drill” had been programmed. Which of the three tools from the “Drills” group is determined as the first step of the tool change process.

The set tool search strategy is taken into consideration.

- When the strategy “Take the first available tool from the group” is applied, T10 from location 3 is loaded.
- When the strategy “Take the first tool with “active” status from the group” is applied, T1 is “loaded”.

T15 at location no. 1 cannot be used, because it is disabled. No alarm is generated if the programmed location does not contain a tool when the T = the location programming method is used.

If more than one magazine is assigned to the toolholder, then the programmed location number refers to the magazine that is the first magazine defined in the distance table.

If the tools of the tool group are different magazines of the toolholder, the search procedure is the same as with the standard TOOLMAN system.

Notice

With the T=location function, T= “Drill” can be programmed alternatively
T = 1 ;Tool
T = “Drill” ;Tool with identifier Drill

5.10.1 Call multiple turrets with “T=location number”

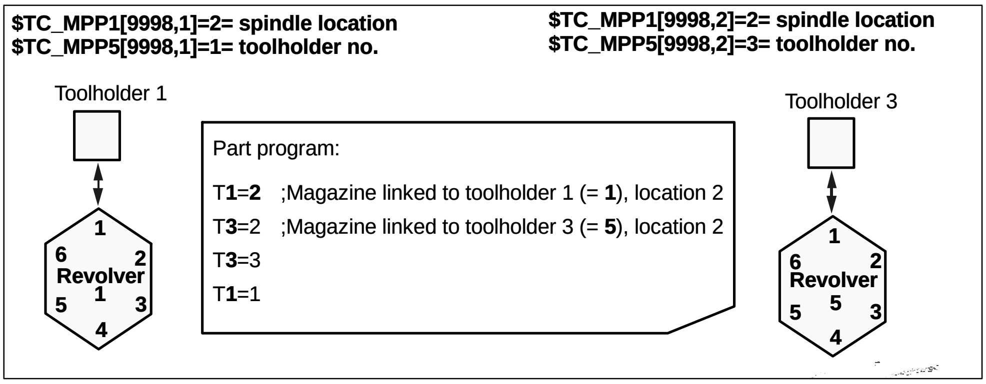


Bild 5-12 T=location number as tool management function on turning machines

The programming option “T = location number” and several magazines can be used to work in one channel or in one TO unit.

- The NC address T can be programmed with an address expansion T1 =...
- Tool management then interprets this as the spindle number or as the toolholder number.
- T without address extension then refers to the main spindle.

5.11 Programming examples

Action	Program command	Description
Create tool	DEF INT DUPLO_NO DEF INT T_NO DUPLO_NO = 7 T_NO=NEWT(“DRILL”, DUPLO_NO)	Create new tool called Drill with duplo no.= 7. The automatically generated T no. is stored in “T_NR”.
	T_NR = GETT(“DRILL”, DUPLO_NO) or $TC_TP2[1] = “DRILL” ; $TC_TP1[1] = DUPLO_NO	Determine the T number of tool “Drill” with duplo no. 7 that has already been created. The T no is however also assigned here by the programming.
Tool data read/write	$TC_DP1[GETT(“DRILL”, DUPLO_NO), 2] = 210	Write tool type for the 2nd cutting edge of tool “Drill”/DUPLO_NO
	$TC_DP1[T_NR, 2] = 210	Write tool type for the 2nd cutting edge of the tool “T - number”
Tool Select	T=“DRILL” or: T=GETT(“DRILL”, DUPLO_NO) or Tx	If there are several tools with this identifier, then the T no. is returned to the first tool possible Determines T number for “DRILL” with duplo = DUPLO_NO and selects it. Call with T no. e.g. T1,T2,T3,....
ToolDelete	$TC_TP1[T_NR,0]=0 or DELT (“DRILL”, DUPLO_NO) $TC_TP1[GETT(“DRILL”),0]=0 or alternatively: DELT(“DRILL”)	Tool with T_NO is deleted, Tool “DRILL”, DUPLO_NO is deleted

5.12 Overview of the remaining OPI blocks of tool management

The line need only be calculated if the OPI variable is followed by a field [ ]. The value of the line is otherwise 1.

5.12.1 Magazine directory data, HMI internal

OPI block TMV

Calculation of line: Magazine number if [ ] present

Calculation of column: N/A

NCK identifier	Description	OPI variable	Type
None	Number of magazines	numActMags	WORD
	Number of the magazine	magVNo[ ]	WORD
	Identifier of the magazine	magVIdent[ ]	String

5.12.2 Tool directory data, HMI internal

OPI block TV

Calculation of line: Sequential number of the tools, if [ ] present

Calculation of column: N/A

NCK identifier	Description	OPI variable	Type
None	Number of tools in TO area	numTools	WORD
	Last T number assigned for tool management	TnumWZV	WORD
	T number	toolNo[ ]	WORD
	Tool designation	toolIdent[ ]	String
	Duplo number	nrDuplo[ ]	WORD
	Number of cutting edges	numCuttEdges[ ]	WORD
	Current magazine	toolInMag[ ]	WORD
	Current location	toolInPlace[ ]	WORD
	Number of tool groups	numToolsGroups	WORD

5.12.3 Parameterization, return parameters TMGETT, TSEARC

OPI block TF

Calculation of line: See table

Calculation of column: N/A

Description	OPI variable	Calculation of line	Type
Return: found tools	resultNrOfTools	1	WORD
Return: T numbers of found tools	resultToolNr[ ]	1... resultNrOfTools	WORD
Number of cutting edges used	resultNrOfCut EdgesUsed	1	WORD
T number of cutting edge used	resultToolNr Used	Number of cutting edges of tool	WORD
D number of cutting edge used	resultCutting EdgeNrUsed	Number of cutting edges of tool	WORD
Mask for search criterion of PI TSEARCH (OPI block TD)	parMasksTD	Parameter index of block TD	WORD
Comparison value for PI TSEARCH of variables of the OPI block TD	parDataTD	Parameter index of block TD	WORD
Comparison value for PI TSEARCH of variables of the OPI block TD	parDataToolIdentTD	Parameter index of block TD	String
Comparison value for PI TSEARCH of variables of the OPI block TU	parMasksTU	Parameter index of block TU	WORD
Comparison value for PI TSEARCH of variables of the OPI block TU	parDataTU	Parameter index of block TU	REAL
Comparison value for PI TSEARCH of variables of the OPI block TO	parMasksTO	Parameter index of block TO	WORD
Comparison value for PI TSEARCH of variables of the OPI block TO	parDataTO	Parameter index of block TO	REAL
Comparison value for PI TSEARCH of variables of the OPI block TUE	parMasksTUE	Parameter index of block TUE	WORD
Comparison value for PI TSEARCH of variables of the OPI block TUE	parDataTUE	Parameter index of block TUE	REAL
Comparison value for PI TSEARCH of variables of the OPI block TS	parMasksTS	Parameter index of block TS	WORD

Description	OPI variable	Calculation of line	Type
Comparison value for PI TSEARCH of variables of the OPI block TS	parDataTS	Parameter index of block TS	REAL
Comparison value for PI TSEARCH of variables of the OPI block TUS	parMasksTUS	Parameter index of block TUS	WORD
Comparison value for PI TSEARCH of variables of the OPI block TUS	parDataTUS	Parameter index of block TUS	REAL

5.12.4 Working offsets

OPI block AEV

Calculation of line: Cutting edge number if [ ] present

Calculation of column: N/A

NCK identifier	Description	OPI VAR	Type
None	Number of D numbers in block	numActDEdges	WORD
	D numbers	Dno[...]	WORD
	Internal T number	toolNo[...]	WORD
	Cutting edge number	cuttEdgeNo[...]	WORD
	Tool identifier	toolIdent[...]	STRING
	Duplo number	duploNo[...]	WORD
	Magazine	toolInMag[...]	WORD
	Location	toolInPlace[...]	WORD

5.12.5 PI services and NC language commands for tool management

FB 4 (PI_SERV) or FB 7 can be used to start program instance services (PI services) in the NCK area. A program section which carries out a particular function (e.g., with tool management, search for empty location in a magazine), is executed in the NCK by making a request via the PI service.

References: /FB1/ P3, Basic PLC Program

PI service	Functions	NC language command	SW version
MMCSEM	Semaphores for various PI services		
DELETO	ToolDelete	DELT("TL", Duplo)	
DELETE	Delete a cutting edge	$TC_DP1[t,d]=0	
CREATO	Generate tool	NEWT("TL", Duplo)	
CRTOCE	Create tool specifying cutting edge no.	$TC_DPx[t,D] $TC_DPCx[t,D] $TC_DPCSx[t,D] $TC_MOPx[t,D] $TC_MOPCx[t,D]	SW 5
TMCRTO	Create tool	$TC_TPx[t]	
TMCRTC (not available in PLC)	Create tool specifying cutting edge no.	$TC_DPx[t,d]	SW 5
CREACE	Create cutting edge	$TC_DP[t,d]=value	
CRCEDN	Create new cutting edge	$TC_DPx[t,d]	
TMFDPL	Empty location search for loading	GETFREELOC	SW 6
TMMVTL	Prepare magazine location for loading, unload tool		
TMPCIT	Set incremental value for workpiece counter, decrement count by y	SETPIECE(SpinNo,y)	
TMPOSM	Position magazine location or tool	POSM (p, m, ip, im)	SW 5
TMFPBP	Find empty location acc. to properties		
TSEARC	Complex search using search screen forms	User cycle program	
TMRASS	Reset active status		SW 5
TMGETT (not available in PLC)	Confirm T number for specified tool identifier with duplo number	GETT("TL", Duplo)	
	Read the preset T number	GETSELT(SpinNo)	
CHKDNO (not available in PLC) TMCHKD (not available in PLC)	Check the uniqueness of D numbers of the tool data of the TO unit assigned to the executing channel. Parameters t1, t2,d are optional.	Status=CHKDNO (t1,t2,d)	SW 5

PI service	Functions	NC language command	SW version
DZERO (not available in PLC)	Set D number for all tools of the TO unit assigned to the channel to “invalid” D numbers of this type are displayed with value 0 on the OPI. The invalid D number is generated NCK-internally by assigning the value “old D number” + 32000 to the D number.	DZERO	SW 5
	For the offset number D=d, get the associated internal T no. = t of the tool The tool that has the status “active” and “was in use” is taken from the tool group.	Status=GETACTTD (t,d)	SW 5
	Get the D no. for tool t and its cutting edge ce	d = GETDNO(t, ce)	SW 5
	Set the D no. of TL t and its cutting edge ce to value d	Status=SETDNO (t,ce,d)	SW 5
	Read the active T no. and status	Status=GETACTT (Tno,“WZ”)	SW 4
	Delete command for all location-dependent/setup offsets of a cutting edge or tool if d is not specified	Status = DELDL(t, d)	SW 5
SETTST (not available in PLC)	Set tool status to “active”	SETTA(Stat,m,vnr)	SW 5
SETTST (not available in PLC)	Set tool status to “not active”	SETTIAStat,m,vnr)	SW 5
CHKDM (not available in PLC)	Check uniqueness of D nos. in magazine; m=Magazine	CHKDM(m)	SW 5
Value of MD can be manipulated	Set toolholder no. (h=holder no.)	SETMTH(h)	
	Set master spindle (s=spindle no.)	SETMS(s)	
TRESMO (not available in PLC)	Activate setpoint for tool life/workpiece count/wear	RESETMON	
TMAWCO (not available in PLC)	Set a wear group to active	$TC_MAP9	SW 5

NC commands

NCK states are read with the following NC commands.

Functions	NC language command	SW version
Active TL no. T	$P_TOOLNO	
Last programmed TL no. (without tool management)	$P_TOOLP	
Active tool offset D	$P_TOOL	
Active tool length; n=1–3	$P_TOOLL[n]	
Active toolholder	$P_TC	
Active angle of a toolholder axis	$P_TCANG[n]	
Diff angle	$P_TCDIFF[n]	
Active radius	$P_TOOLR	
Number of cutting edges of tool t	$P_TOOLND[t]	
Tool exists with number	$P_TOOLEXIST[t]	
Active tool offsets, n=1–25,....31	$P_AD[n] $P_ADT[n]	
Active DL number	$P_DLNO	
Number of T word substitute subroutine for T	$C_T	SW 5
Programmed TL identifier (with TOOLMAN) for substitute subroutine for T	$C_TS	SW 5
Bool variable: contents in $C_T?	$C_T_PROG	SW 5
Bool variable: contents in $C_TS?	$C_TS_PROG	SW 5
1: Move (load/unload, relocate...); 2: Prepare change; 3: Change ON; 4: Change ON (turret, without M06); 5: Prepare change and change ON (with M06)	$AC_TC_FCT	SW 5
Acknowledgment status of PLC FC 8	$AC_TC_STATUS	SW 5
Toolholder or spindle number	$AC_TC_THNO	SW 5
New tool from magazine	$AC_TC_MFN	SW 5
New tool from location	$AC_TC_LFN	SW 5
New tool to magazine	$AC_TC_MTN	SW 5
New tool to location	$AC_TC_LTN	SW 5
Old tool from magazine	$AC_TC_MFO	SW 5
Old tool from location	$AC_TC_LFO	SW 5
Old tool to magazine	$AC_TC_MTO	SW 5
Old tool to location	$AC_TC_LTO	SW 5
Master spindle	$AC_MSNUM	SW 5
Master spindle	$P_MSNUM	SW 5

Master toolholder	$AC_MTHNUM	SW 5
Master toolholder	$P_MTHNUM	SW 5
Magazine number of tool t	$A_TOOLMN[t]	SW 6
Magazine location of tool t	$A_TOOLMLN[t]	SW 6
Number of the owner magazine	$A_MYMN	SW 6
Number of the owner magazine	$A_MYMLN	SW 6
Number of defined magazines	$P_MAGN	SW 6
Number of defined magazines, i-th magazine number	$P_MAG[i]	SW 6
Number of defined adapters	$P_MAGNA	SW 6
Number of defined adapters, i-th adapter	$P_MAGA[i]	SW 6
Number of linked magazines	$P_MAGNDIS	SW 6
Number of the i-th magazine that is linked with location l of the buffer magazine	$P_MAGNDISS[l,i]	SW 6
Number of the i-th magazine that is linked with location l of the loading magazine	$P_MAGNDISL[l,i]	SW 6
Number of defined magazine location type hierarchies	$P_MAGNH	SW 6
Number of the defined location types in the n-the defined hierarchy	$P_MAGNHLT[n]	SW 6
m-th location type of hierarchy n	$P_MAGHLT[n,m]	SW 6
Number of spindle numbers, toolholder numbers n for assigned buffer	$P_MAGNREL[n]	
m-th buffer number of the n-th spindle no., toolholder no.	$P_MAGREL[n,m]	SW 6
Number of spindle locations, toolholder locations in the buffer magazine	$P_MAGNS	SW 6
n-th number of the spindle/toolholder in the buffer	$P_MAGS[n]	SW 6
Determine the defined D numbers of a tool	$P_TOOLD	SW 6
Determine existence of a tool	$P_TOOLEXIST	SW 4
Number of DL offsets for the D offset	$P_TOOLNDL[t,d]	SW 4
Number of defined tool groups that are assigned to the channel	$P_TOOLNG	SW 6
Number of tools that are assigned to the channel	$P_TOOLNT	SW 6

i-th tool number T	$P_TOOLT[i]	SW 6
A subset of the tool of the tool group is named that is then available for a subsequent tool change.	$P_USEKT, $TC_TP11	SW 6

Please refer to Chapter 3 for explanations.

6 Data backup

6.1 Back up the NCK data

How to read the data in over the RS-232 (V.24) interface is described in:

References: /BAD/ Operator's Guide HMI Advanced

Complete backup

All the data of the active file system are output via file INITIAL.INI.

Tool data

All tool-specific data is backed up in file _N_TOx_TOA.

Magazine data

All magazine data is backed up in file _N_TOx_TMA.

Tool and magazine data

All tool and magazine data is backed up in file _N_TOx_INI.

The presence/absence of the data reference in the following is determined primarily by the appropriateness of the MD settings.

Notice

Please make sure that the spindle is empty before backing up data. Should it no longer possible e.g in the service case to change the tool, then the back-up can still be executed. Alarm "22070 TO unit 000x, please load tool T=000x to magazine and repeat data backup" is issued. The data is backed up correctly; however, you must make sure that this backup is used only for this machine, since the current states are also saved.

The format in the backup file is as follows:

1. Tool definitions
2. Magazine definitions
3. Parameters which set up a relationship between defined tools and defined magazine locations.

1. Tool definitions

$TC_TP1[i]	**Tool data**
. . .	
$TC_TP11[i]	
;	
$TC_TPC1[i]	CC user tool data
. . .	
$TC_TPC10[i]	
;	
$TC_DP1[i, j]	**Cutting edge data** (available without/with tool management)
. . .	
$TC_DP25[i , j]	
;	
$TC_DPC1[i, j]	CC cutting edge data
. . .	
$TC_DPC10[i , j]	
;	
$TC_MOP1[i, j]	Monitoring data
$TC_MOP4[i , j]	
;	
$TC_MOPC1[i, j]	CC monitoring data
. . .	
$TC_MOPC10[i , j]	
$TC_TPG1[i]	Grinding (exists only for tools of type ‘Grinding tool’ with/without TM)
. . .	
$TC_TPG9[i]	
;	
$TC_TP1[i+1]	**Tool data**
. . .	
$TC_TP11[i+1]	
;	
$TC_TPG1[i+1]	Grinding
. . .	
$TC_TPG9[i+1]	
;	
. . .	
. . .	

2. Magazine definitions

$TC_MAMP1	**Magazine module parameters**
$TC_MAMP2	
;	
$TC_MPTH[n, m]	**Magazine location type hierarchy structures**
;	
$TC_MAP1[i]	**Magazine parameters**
. . .	
$TC_MAP8[i]	
;	
$TC_MAPC1[i]	CC magazine parameters
. . .	
$TC_MAPC10[i]	
;	
$TC_MPP1[i, j]	**Magazine location parameters**
. . .	
$TC_MPP5[i, j]	
;	
$TC_MPPC1[i, j]	CC magazine location parameters
. . .	
$TC_MPPC10[i, j]	
;	
$TC_MAP1[i+1]	**Magazine parameters**
. . .	
$TC_MAP8[i+1]	
. . .	
;	
$TC_MDP1[k, l]	**Magazine distance to spindles,**...
. . .	
$TC_MDP2[k, l]	
. . .	
;	
$TC_MLSR[k, l]	Relationship between buffer locations and spindles;
. . .	

3. Relationship between tools and magazine locations

$TC_MPP6[i, j]	Tool in magazine location
$TC_MPP6[i, j +1]	
. . .	
$TC_MPP6[i, j +J]	
$TC_MPP6[i+1, j]	
$TC_MPP6[i+1, j +1]	
. . .	
$TC_MPP6[i+1 j +J]	
;	

```
$TC_MPP6[ i+I, j ]
$TC_MPP6[ i+I, j +1 ]
. . .
$TC_MPP6[ i+I, j +J ]
;
M17
```

The data in the magazine module is only backed up if at least one magazine location has been defined.

Notice

Tool management data of tool management functions not available are ignored when writing data into the active file system. No alarm is produced.

An alarm (17020 = ‘Index error’) is however always generated by reading tool management data that is not present.

This means that tool management data records (backup files) that have been generated in the NCK with a special tool management function configuration can be transferred to other SINUMERIK 840D controls that have different tool management functions. The permitted data are then “filtered out”.

6.2 Back up the PLC data

Use the programming device (S7) to save DB 4. The type and number of magazines, loading points, stations and spindles are stored here. The basic program uses this information to set up the interface blocks.

6.3 Data backup on hard disk

The access database from the directory **Services \ Tool management \ Tool management data \ WZACCESS.MDB**.

This file contains all tool data of the HMI
– IB data (configuration, buffer, loading magazine)
– Tool catalog, tool cabinet
– Magazine configurations
. . .

Notice

Attention shall be paid under all circumstances during the back-up routine that a Power ON for the the HMI and NCK has been executed beforehand, e.g. by OFF/ON, to ensure that the database is not opened.

6.4 $TC_MPP66 – Expansion for the data backup with tools in the buffer

Loaded tools that are in a buffer when the data backup is executed have caused the magazine location from where they were loaded to go to state “reserved for tools from buffer”.

In the backup file, the new system variable $TC_MPP66 contains the information which is not yet backed up. This information is necessary to make the tool in the buffer known the the (proprietor) location in the magazine when importing the data again. This makes it possible to load fixed-location-coded tools back to their specific location in the magazine.

Name	**$TC_MPP66[n,m]**			
Meaning	T number (of the tool in the buffer) that has been reserved for the location defined by n, m. A write operation is only meaningful when uploading a backup file to the NCK. The name is based on $TC_MPP6–T-No. of the tool in the buffer.			
Data type	INT		in software version 6 and higher	
Value range	1–32000			
Indices	Meaning			Value range
	n = Magazine number m = Location number			
Access	Read in part program	Write in part program	Read in synchronous action	Write in synchronous action
	X	X	–	–
Implicit preprocessing stop	–	–	–	

Restrictions

7

Hardware

- 8MB user memory
- OP 030 only in conjunction with HMI Embedded
- PCU50 with OP012

software

The PLC blocks for tool management must be integrated into the PLC from the “basic program” toolbox (FC 6, FC 7, FC 8, FC 22).

Options

The tool management option must be active.

M06 and T command

The T number and the M06 command are not transferred to the PLC as auxiliary functions when tool management is active but to the tool management interface DB71 to DB73 in the PLC instead.

Notes

Machine data 8

8.1 Machine Data

8.1.1 Display machine data for HMI

No.	Name TM_DEFAULT	Description	Default setting	Max. value
9412	TOOLSIZE	Default setting for tool size	1111	7777
9416	TOOLTYPE	Default setting for loading, tool type	120	900
9417	TOOLSTATE	Default setting for loading, tool status	0	256
9418	SHOW_TOOL_SIZE	The tool size is shown in the displays as two or four digits	0	256
9419	DELETE_TOOL	Automatic deletion of tool data during unloading 0: No automatic deletion 1: Automatic deletion	0	1

No.	Name TM_DEFAULT	Description	Default setting
9250	SKMGLIST	Display of magazine list (horizontal)	7
9251	SKTLLIST	Display of tool list (horizontal)	7
9252	SKTOOLLOAD	Access rights for loading	7
9253	SKTOOLUNLOAD	Access rights for unloading	7
9254	TOOL_MOVE	Access rights for relocating	7
9255	SKMGLREPR1	Display of 1st magazine list (horizontal)	7
9256	SKMGLREPR2	Display of 2nd magazine list (vertical)	7
9257	SKMGLREPR3	Display of 3rd magazine list (vertical)	7

No.	Name TM_DEFAULT	Description	Default setting
9258	SKCNNEWTOOLE	Access rights: Create new cutting edge	7
9259	SKNCDELTOOL	Access rights: Delete tool	7
9260	SKMGBUFFER	Access rights: Power ON/Power OFF Buffer	7
9261	SKMGFIND	Access rights: Find	7
9262	SKMGLISTPOS	Access rights: Position	7
9263	SKMGNEXT	Access rights: Scroll to next magazine	7
9264	SLTLNEWTOOL	Access rights: Create a new tool	7
9265	SKMTLREPR1	Display 1st tool list (vertical)	7
9266	SKMTLREPR2	Display 2nd tool list (vertical)	7
9267	SKMTLREPR3	Display 3rd tool list (vertical)	7
9268	SKFINDPL	Access rights: Empty location softkey	7
9269	SKFINDPLACE	Access rights: Empty location softkey and display tool list	7
9270	SKACTPLACE	Access rights: Load current location	7
9271	SKLDTOOLDAT	Access rights: View and edit tool data (the tool data can be protected individually with machine data 9201, 9202 and 9209).	7

8.1.2 Memory settings for tool management

18080 MD number	**MM_TOOL_MANAGEMENT_MASK**				
Default setting: 0x0		Minimum input limit: 0		Maximum input limit: 0xFFFF	
Changes effective after: Power ON			Protection level: 1/7		Unit: –
Data type: DWORD				Applies as of SW 2	
Meaning:	Activation of the tool management memory with “0” means: The set tool management data do not occupy any memory, tool management is not available. Bit 0=1: Memory for tool-management-specific data will be made available, the memory-reserving machine data must however be accordingly set (MM_NUM_MAGAZINE_LOCATION, MM_NUM_MAGAZINE) Bit 1=1: Memory for monitoring data is made available Bit 2=1: Memory for user data (CC data) is made available Bit 3=1: Memory for “consider adjacent location” is made available Bit 4=1: Memory and function enable for the PI service _N_TSEARC = “complex search for tools in magazines” is made available. Bit5=1: Wear monitoring active Bit 6=1: Wear group available Bit 7=1: Reserve memory for adapter of magazine locations Bit 8=1: Memory for insert and/or setup compensations Bit 9=1: Tools of a turret no longer exit their turret location for a tool change (in terms of display). The coded type of memory reservation enables economic use of memory management for the functionality provided. Example: Standard memory reservation for tool management: MM_TOOL_MANAGEMENT_MASK = 3 (Bit 0 + 1=1) means tool management and tool monitoring data are made available MM_TOOL_MANAGEMENT_MASK = 1 means tool management without tool monitoring function data				
Special cases, errors,...					

8.1.3 NC-specific machine data

17500 MD number	**MM_MAXNUM_REPLACEMENT_TOOLS** Maximum number of replacement tools	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 32
Changes effective after: Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 5.1	
Meaning:	Only meaningful if tool management function or tool monitoring function are active Value Meaning 0 The number of replacement tools is not monitored. 1 There can be exactly one replacement tool for each identifier. This data does not affect memory requirements, but merely has a monitoring function.	
Corresponding to...	MD 18080: MM_TOOL_MANAGEMENT_MASK	
Further references:	Description of Functions: Memory Configuration (S7)	

17510 MD number	**$MN_TOOL_UNLOAD_MASK** Behavior of tool data at unloading	
Default setting: –	Minimum input limit: –	Maximum input limit: –
Changes effective after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 6.3	
Meaning:	When the tool is unloaded, some tool data can be assigned fixed values:	
Corresponding to...		
Further references:		

Bit no.	Bit value	Hexadecimal Value	Meaning
0	0		Tool status "active" remains unchanged
	1	"H1"	Tool status "active" is deleted ($TC_TP8, bit 0)
1	0		Tool status "was in use" remains unchanged
	1	"H2"	Tool status "was in use" is deleted ($TC_TP8, bit 7)
2	0		Tool parameter $TC_TP10 remains unchanged
	!	"H4"	Tool parameter $TC_TP10 is set to value 0. This means the tool replacement strategy is reset.
3	0		Tool parameter $TC_TP11 remains unchanged
	1	"H8"	Tool parameter $TC_TP11 is set to value 0. This means the assignment to the tool subgroup is removed.

17515 MD number	$MN_TOOL_RESETMON_MASK Behavior of tool data at RESETMON
Default setting: 0x14	Minimum input limit: 0 / Maximum input limit: 0xfffff960
Changes effective after POWER ON	Protection level: 2/7 / Unit: –
Data type: DWORD	Applies as of SW 7.3
Meaning:	The RESETMON command specifies in the 5th parameter which tool status is to be reset. If the 5th parameter is omitted, it is replaced with the value from this MD. This value is always used with PI service "_N_TRESMON". The bits are assigned like the bits in tool status $TC_TP8[x].

Bit no.	Bit value	Hexadecimal Value	Meaning
0	0		Tool status "active" remains unchanged
	1	"H1"	Tool status "active" is deleted
1	0		Tool status "enabled" remains unchanged
	1	"H2"	Tool status "enabled" is set
2	0		Tool status "disabled" remains unchanged
	!	"H4"	Tool status "disabled" is deleted if permitted by the monitoring data and the 4th parameter is set accordingly.
3	0		Tool status "measured" remains unchanged
	1	"H8"	Tool status "measured" is set
4	0		Tool status "prewarning limit" remains unchanged
	1	"H10"	Tool status "prewarning limit" is deleted
5			Not permitted (tool status "tool being changed")
6	0		Not permitted (tool status "tool is fixed-location-coded")
7	0		Tool status "was in use" remains unchanged
	1	"H80"	Tool status "was in use" is deleted
8	0		Not permitted (tool status "being transported back")
9			Tool status "disabled is ignored" remains unchanged
	1	"H200"	Tool status "disabled is ignored" is deleted
10	0		Tool status "to unload" remains unchanged
	1	"H400"	Tool status "to unload" is deleted
11			Not permitted (tool status "to load")
12	0		Not permitted (tool status "master tool")
13, ff			Not permitted (is reserved)

The default setting corresponds to behavior up to now.
The bits that are not permitted as filtered and hidden by the limit screen.

Corresponding to...	
Further references:	

17520 MD number	**$MN_TOOL_DEFAULT_DATA_MASK** Create new tool: Data default setting		
Default setting: –	Minimum input limit: –	Maximum input limit: –	
Changes effective after POWER ON	Protection level: 2/7	Unit: –	
Data type: DWORD	Applies as of SW 6.3		

Meaning: When a tool is redefined, some tool data can be assigned fixed default values. This way simple applications do not need to process data which does not necessarily need to be assigned individual values.

Bit no.	Bit value	Hexadecimal Value	Meaning
0	0		Default value of tool status ($TC_TP8), Bit1=0="not enabled"
	1	"H1"	Default value of tool status ($TC_TP8), Bit1=1="enabled"
1	0		Default value of tool status ($TC_TP8), Bit6=0="not fixed-location-coded"
	1	"H2"	Default value of tool status ($TC_TP8), Bit6=1="fixed-location-coded"
2	0		The tool is only included in the tool group with the explicit write command for the tool name. Only then can it be loaded at change.
	1	"H4"	The tool is automatically included in the tool group when redefinded. (Now tool change can be performed with the default name ("t"=t-no.). The "tool name" ($TC_TP2) can be hidden to the user. This is only meaningful if replacement tool are not used.)
3	0		TMMG only: Default value of location type ($TC_TP7)=9999=not defined
	1	"H8"	TMMG only: Default value of location type ($TC_TP7)=1 and associated default values of magazine location type (TC_MPP2)=1. All magazine locations can now accept all tools.

Corresponding to...	
Further references:	

17530 MD number	$MN_TOOL_DATA_CHANGE_COUNTER Identify tool data change for HMI	
Default setting: –	Minimum input limit: –	Maximum input limit: –
Changes effective after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 6.3	
Meaning:	HMI display support. This data setting allows explicit inclusion/exclusion of individual data in the OPI variables (block C/S) toolCounter, toolCounterC, toolCounterM.	
Corresponding to...		
Further references:		

Bit no.	Bit value	Hexadecimal Value	Meaning
0	0		Value changes to tool status ($TC_TP8) are not taken into accont in toolCounterC
	1	“H1”	Value changes to tool status ($TC_TP8) are taken into accont in toolCounterC
1	0		Value changes to remaining tool count ($TC_MOP4) are not taken into accont in toolCounterC
	1	“H2”	Value changes to remaining tool count ($TC_MOP4) are taken into accont in toolCounterC

“Value changes to tool status” and “Value changes to remaining tool count” are relative to the value changes which are caused by internal processes in the NC, as well as to value changes caused by writing the respective system variables.

17540 MD number	**MM_TOOLTYPES_ALLOWED** Definition of tool types permitted in the NCK for tool offset selection				
Default setting:		Minimum input limit:		Maximum input limit:	
Changes effective after: Power ON			Protection level: 2/7		Unit: -
Data type: DWORD				Applies as of SW 6.4	
Meaning:	Definition of tool types (see $TC_DP1) permitted in the NCK for tool offset selection. Tools of any tool type can be loaded to the NCK, however, only the tool types specified here can be defined in the tool determining the offset. A bit value = 1 means that the specified tool type range is permitted for the offset selection. A bit value = 0 means that the specified tool type range is not permitted for the offset selection. When an attempt is made to select an offset for a cutting edge of this type, the selection is refused and an offset-capable alarm is issued. Value = 0, 9999 for the tool type means “not defined”. In general, it is not possible to select tool offsets with this value for the tool type. Bit no. Description 0 Tool types 1 to 99 permitted 1 Tool types 100 to 199 permitted (milling tools) 2 Tool types 200 to 299 permitted (drilling tools) 3 Tool types 300 to 399 permitted 4 Tool types 400 to 499 permitted (grinding tools) 5 Tool types 500 to 599 permitted (turning tools) 6 Tool types 600 to 699 permitted 7 Tool types 700 to 799 permitted 8 Tool types 800 to 899 permitted 9 Tool types 900 to 999 permitted				
Corresponding to...					
Further references:					

18082 MD number	**MM_NUM_TOOL** Number of tools the NCK can manage				
Default setting: 30		Minimum input limit: 0		Maximum input limit: 600	
Changes effective after POWER ON			Protection level: 2/7		Unit: -
Data type: DWORD				Applies as of SW 2.	
Meaning:	The number of tools which the NCK can manage is entered here. The maximum possible number of tools corresponds to the number of edges in the NCK. Battery-backed memory is reserved for the number of tools.				
Special cases, errors,...					
Corresponding to...	MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA				
Further references:	Description of Functions: Memory Configuration (S7), Tool Offset (W1)				

18084 MD number	MM_NUM_MAGAZINE Number of magazines the NCK can manage	
Default setting: 3	Minimum input limit: 0	Maximum input limit: 32
Changes effective after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 2	
Meaning:	Tool management (TMG) only if MD tool management and option tool management are set: Number of magazines that the NCK can manage (active and background magazines). The non-volatile memory for the magazines is reserved with this machine data. Important: One load magazine and a buffer magazine is set up in the tool management for each TOA unit. These magazines must be taken into account. Value = 0: Tool management cannot be activated because no data can be created.	
Special cases, errors,...		
Corresponding to...		
Further references:	Description of Functions: Memory Configuration (S7)	

18086 MD number	MM_NUM_MAGAZINE_LOCATION Number of magazine locations the NCK can manage	
Default setting: 30	Minimum input limit: 0	Maximum input limit: 600
Changes effective after: Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 2	
Meaning:	TMG – only if MD for tool management and tool management option are set: Number of magazine locations that the NCK can manage. The battery-backed memory for the magazines is reserved with this MD. Important: The number of all buffer locations and loading points has to be taken into account here. Value = 0: Tool management cannot be activated because no data can be created.	
Special cases, errors,...		
Corresponding to...		
Further references:	Description of Functions: Memory Configuration (S7)	

18088 MD number	MM_NUM_TOOL_CARRIER Maximum number of toolholders	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 99999999
Changes effective after: Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 4.1	
Meaning:	The maximum number of toolholders that can be defined for orientational tools in the TO range. The value is divided by the number of active TO units. The integer result indicates how many toolholders can be defined per TO unit. The data for defining a toolholder is set with the system variables $TC_CARR1, ...$TC_CARR14, or via HMI operator screens or generally via the variable service of the OPI. Example: 2 channels are active, with one channel per TO unit (=default). 3 holders must be defined in channel 1 and one holder in channel 2. The value to be set is 6 because 6/2 = 3.	
Special cases, errors,...		
Corresponding to...		
Further references:	Description of Functions: Tool Offsets (S7)	

18090 MD number	MM_NUM_CC_MAGAZINE_PARAM Number of magazine data for users/compile cycles	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 10
Changes effective after POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies as of SW 2	
Meaning:	Number of magazine parameters (of the integer type) that are made available to the user or the Compile Cycle. If this machine data is set, the amount of non-volatile memory required is increased by sizeof(int)* max. number of magazines.	
Special cases, errors,...		
Corresponding to...	MD 18084: MM_NUM_MAGAZINE	
Further references:		

18091 MD number	MM_TYPE_CC_MAGAZINE_PARAM[n] Type definition for magazine-oriented user data	
Default setting: 3	Minimum input limit: 1	Maximum input limit: 6
Changes effective after: Power ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies as of SW 5.2	
Meaning:	**The default settings for this machine data must not be altered.** Used to assign individual types to the parameters. The array index n can assume values between 0 and the setting in machine data MD 18090: MM_NUM_CC_MAGAZINE_PARAM. The possible values of the MD = 1, 2, 3, 4, 5 and 6 denote the NC command types BOOL, CHAR, INT, REAL, STRING and AXIS. The type FRAME cannot be defined here. Type STRING must not be longer than 31 characters. Example: MD 18090: MM_NUM_CC_MAGAZINE_PARAM=1 MD 18091: MM_TYPE_CC_MAGAZINE_PARAM=5 “UserMagazine” can be programmed for parameter $TC_MAPC1. The non-volatile RAM is used. Changing the value can, but does not necessarily, result in reconfiguration of the non-volatile memory.	
Corresponding to...	MD 18090: MM_NUM_CC_MAGAZINE_PARAM MD 18084: MM_NUM_MAGAZINE	
Further references:		

18092 MD number	MM_NUM_CC_MAGLOC_PARAM Number of magazine location data for users/compile cycles	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 10
Changes effective after POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies as of SW 2.	
Meaning:	Number of magazine-location data parameters (of the integer type) that are made available to the user or the Compile Cycle. If this machine data is set, the amount of non-volatile memory required increases by sizeof(int)∗ max. number of magazine locations.	
Special cases, errors,...		
Corresponding to...	MD 18086: MM_NUM_MAGAZINE_LOCATION	
Further references:		

18093 MD number	MM_TYPE_CC_MAGLOG_PARAM[n] Type definition for magazine location-oriented user data	
Default setting: 3	Minimum input limit: 1	Maximum input limit: 6
Changes effective after: Power ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies as of SW 5.2	
Meaning:	**Settings deviating from the standard pre-assignment are not supported by the standard HMI display up to now.** Used to assign individual types to the parameters. The array index n can assume values between 0 and the setting in machine data MD 18092: MM_NUM_CC_MAGLOC_PARAM. The possible values of the MD = 1, 2, 3, 4 and 6 denote the NC command types. 1 BOOL, 2 CHAR, 3 INT, 4 REAL and 6 AXIS The type STRING cannot be used explicitly here, value 5 is treated like value 2. The type FRAME cannot be defined here. Example: MD 18090: MM_NUM_CC_MAGAZINE_PARAM=1 MD 18091: MM_TYPE_CC_MAGAZINE_PARAM=2 “UserMagazineLocation” can be programmed for parameter $TC_MPPC1. The non-volatile RAM is used. Changing the value can, but does not necessarily, result in reconfiguration of the non-volatile memory.	
Corresponding to...	MD 18092: MM_TYPE_CCS_MAGLOG_PARAM	
Further references:		

18094 MD number	MM_NUM_CC_TDA_PARAM Number of tool parameters for users/compile cycles	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 10
Changes effective after POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies as of SW 2	
Meaning:	Number of tool-specific data that can be created for each tool (of type integer) and are available to the user or compile cycle. If this machine data is set, the amount of non-volatile memory required increases by sizeof(double) ∗ max. number of tools.	
Special cases, errors,...		
Corresponding to...	MD 18082: MM_NUM_TOOL	
Further references:		

18095 MD number	**MM_TYPE_CC_TDA_PARAM[n]** Type definition for tool-oriented user data		
Default setting: 4	Minimum input limit: 1	Maximum input limit: 6	
Changes effective after: Power ON	Protection level: 2/2	Unit: –	
Data type: DWORD	Applies as of SW 5.2		
Meaning:	**Settings deviating from the standard pre-assignment are not supported by the standard HMI display up to now.** Used to assign individual types to the parameters. The array index n can assume values between 0 and the setting in machine data MD 18094: MM_NUM_CC_TDA_PARAM. The possible values of the MD = 1, 2, 3, 4, 5 and 6 denote the NC command types. 1 BOOL, 2 CHAR, 3 INT, 4 REAL, 5 STRING and 6 AXIS. The type FRAME cannot be defined here. Type STRING must not be longer than 31 characters. Example: MD 18094: MM_NUM_CC_TDA_PARAM=1 MD 18095: MM_TYPE_CC_TDA_PARAM=5 “UserCuttingEdge” can be programmed for parameter $TC_TPC1. The non-volatile RAM is used. Changing the value can, but does not necessarily, result in reconfiguration of the non-volatile memory.		
Corresponding to...	MD 18094: MM_NUM_CC_TDA_PARAM MD 18082: MM_NUM_TOOL		
Further references:			

18096 MD number	**MM_NUM_CC_TOA_PARAM** Number of TOA data for users/compile cycles		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 10	
Changes effective after POWER ON	Protection level: 2/2	Unit: –	
Data type: DWORD	Applies as of SW 2		
Meaning:	Number of TOA data that can be created for each tool (of type Double) and are available to the user or compile cycle. If this machine data is set, the amount of non-volatile memory required increases by sizeof(double) * max. number of edges.		
Special cases, errors,...			
Corresponding to...	MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA		
Further references:			

18097 MD number	MM_TYPE_CC_TOA_PARAM[n] Type definition for cutting edge-oriented user data		
Default setting: 4	Minimum input limit: 1		Maximum input limit: 6
Changes effective after: Power ON		Protection level: 2/2	Unit: –
Data type: DWORD		Applies as of SW 5.2	
Meaning:	**Settings deviating from the standard pre-assignment are not supported by the standard-MMC display up to now.** Used to assign individual types to the parameters. The array index n can assume values between 0 and the setting in machine data MD 18096: MM_NUM_CC_TOA_PARAM. The possible values of the MD = 1, 2, 3, 4 and 6 denote the NC command types. 1 BOOL, 2 CHAR, 3 INT, 4 REAL, 6 AXIS. The type STRING cannot be used explicitly here, value 5 is treated like value 2. The type FRAME cannot be defined here. Example: MD 18096: MM_NUM_CC_TOA_PARAM=1 MD 18097: MM_TYPE_CC_TOA_PARAM=5 “UserCuttingEdge” can be programmed for parameter $TC_DPC1. The non-volatile RAM is used. Changing the value can, but does not necessarily, result in reconfiguration of the non-volatile memory.		
Corresponding to...	MD 18096: MM_NUM_CC_TOA_PARAM MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA		
Further references:			

18098 MD number	MM_NUM_CC_MON_PARAM Number of monitoring data for users/compile cycles		
Default setting: 0	Minimum input limit: 0		Maximum input limit: 10
Changes effective after POWER ON		Protection level: 2/2	Unit: –
Data type: DWORD		Applies as of SW 2	
Meaning:	Number of monitoring data that are created for each tool (of type integer) and are available to the user or compile cycle. If this machine data is set, the amount of non-volatile memory required is increased by sizeof(int) * max. number of cutting edges.		
Special cases, errors,...			
Corresponding to...	MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA		
Further references:			

18099 MD number	**MM_TYPE_CC_MON_PARAM[n]** Type definition for monitoring-related user data			
Default setting: 3		Minimum input limit: 1		Maximum input limit: 6
Changes effective after: Power ON			Protection level: 2/2	Unit: –
Data type: DWORD			Applies as of SW 5.2	
Meaning:	**Settings deviating from the standard pre-assignment are not supported by the standard-MMC display up to now.** Used to assign individual types to the parameters. The array index n can assume values between 0 and the setting in machine data MD 18098: MM_NUM_CC_MON_PARAM. The possible values of the MD = 1, 2, 3, 4 and 6 denote the NC command types. 1 BOOL, 2 CHAR, 3 INT, 4 REAL and 6 AXIS. The type STRING cannot be used explicitly here, value 5 is treated like value 2. The type FRAME cannot be defined here. Example: MD 18098: MM_NUM_CC_MON_PARAM=1 MD 18099: MM_TYPE_CC_MON_PARAM=2 “UserCuttingEdge” can be programmed for parameter $TC_MOPC1. The non-volatile RAM is used. Changing the value can, but does not necessarily, result in reconfiguration of the non-volatile memory.			
Corresponding to...	MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA MD 18098: MM_NUM_CC_MON_PARAM			
Further references:				

18100 MD number	**MM_NUM_CUTTING_EDGES_IN_TOA** Number of tool cutting edges in TO area			
Default setting: 30		Minimum input limit: 0		Maximum input limit: 600/1500 (SW 5 and higher)
Changes effective after POWER ON			Protection level: 2/7	Unit: –
Data type: DWORD			Applies as of SW 1	
Meaning:	Number of cutting edges possible in the TO area. For each tool edge approx. 250 bytes per TOA block of the battery-backed memory are reserved with this machine data irrespective of the tool type. Tools with type 400–499 edges (=grinding tools) also occupy the location of a tool edge. Example: Define 10 grinding tools with one cutting edge each. Then the min. settings must be made: MM_NUM_TOOL = 10 MM_NUM_CUTTING_EDGES_IN_TOA = 20 see also MM_NUM_TOOL			
Special cases, errors,...	The data in the buffer are lost when the machine data are changed!			
Corresponding to...				
Further references:	Description of Functions: Memory Configuration (S7)			

18102 MD number	**MM_TYPE_OF_CUTTING_EDGE** Type of D number programming		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after POWER ON	Protection level: 2/7	Unit: –	
Data type: DWORD	Applies as of SW 4.1		
Meaning:	Activates the “flat D number management”. The individual values determine the type of D programming (direct or indirect programming): The default value is 0. This setting means that the NCK manages the T and D numbers. A value > 0 is only accepted by the NCK if bit 0 is not set in MM_TOOL_MANAGEMENT_MASK, i.e. tool management (TMG) and tool monitoring (TMO) cannot be active at the same time. Value Meaning 0 No “flat D number management” active 1 Direct and absolute programming of D numbers Values 2, 3 not enabled up to now		
Special cases, errors,...			
Corresponding to...			
Further references:	Description of Functions: Tool Offset (W1)		

18104 MD number	**MM_NUM_TOOL_ADAPTER** Tool adapter in TO area		
Default setting: 0	Minimum input limit: –1	Maximum input limit: 600	
Changes effective after POWER ON	Protection level: 2/7	Unit: –	
Data type: DWORD	Applies as of SW 5		
Meaning:	Contains the number of tool data records available in NCK. This function can only be used if magazine locations are available in the NCK. I.e. the tool management function must be active. In order to activate the setting, bit 7 (=0x80) must be set in MD $MN_MM_TOOL_MANAGEMENT_MASK. Adapter data blocks and the cutting-edge-specific basis / adapter-data blocks are mutually excluding. I.e. when adapter data are defined, then the parameter $TC_DP21, $TC_DP22, $TC_DP23 or their values in NCK are available. Value Meaning –1 Every magazine location is automatically assigned an adapter, i.e. internally there are just as many adapters foreseen as are foreseen by the magazine locations set in MD $MN_MM_NUM_MAGAZINE_LOCATION. 0 No adapter-data definition possible. Cutting-edge-specific parameters $TC_DP21, $TC_DP22 and $TC_DP23 are available in cases where adapters are utilized outside the active TM. >0 Number of adapter data records. By this, adapters can be defined independently of magazine locations. An additional step following definition of the data assigns the adapters to magazine locations.		
Corresponding to...	MD 18080: MM_TOOL_MANAGEMENT_MASK MD 20310: TOOL_MANAGEMENT_MASK MD 18084: MM_NUM_MAGAZINE MD 18086: MM_NUM_MAGAZINE_LOCATION		
Further references:			

18105 MD number	MM_MAX_CUTTING_EDGE_NO Maximum value of D number	
Default setting: 9	Minimum input limit: 1	Maximum input limit: 32000
Changes effective after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 5	
Meaning:	Maximum number of D numbers per cutting edge is not affected by this. The monitoring of the D number assignment associated with this value is only effective for new definitions of D numbers. This means the existing data records are not checked later – if the MD is changed. Advisable setting: $MN_MM_MAX_CUTTING_EDGE_NO equal to $MN_MM_MAX_CUTTING_EDGE_PER_TOOL. If $MN_MM_MAX_CUTTING_EDGE_NO > $MN_MM_MAX_CUTTING_EDGE_PER_TOOL is set, you should familiarize yourself with the difference between offset number D and the tool edge number CE. See also language commands CHKDNO, CHKDM, GETDNO, SETDNO, DZERO. The is not evaluated with the function “flat D number” and accordingly is not meaningful there. The MD can change the memory requirements: A change from “less than equal to” to “greater than” – or vice versa – in the values of both the above mentioned MDs can influence the demand for volatile memory.	
Corresponding to...	MD 18106: MM_MAX_CUTTING_EDGE_PER_TOOL	
Further references:	Description of Functions: Tool Offset (W1)	

18106 MD number	MM_MAX_CUTTING_EDGE_PER_TOOL Max. number of D numbers per tool	
Default setting: 9	Minimum input limit: 1	Maximum input limit: 12
Changes effective after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 5	
Meaning:	Maximum number of cutting edges (D compensation) per tool (per T number) This allows greater security with the data definition. A value of 1 can be set if only tools with one cutting edge are to be used. This will avoid the problem of assigning more than one cutting edge to the tool when data is defined. Logically the same value is set for MM_MAX_CUTTING_EDGE_NO as for MM_MAX_CUTTING_EDGE_PER_TOOL. If MM_MAX_CUTTING_EDGE_NO is set greater than MM_MAX_CUTTING_EDGE_PER_TOOL, you should familiarize yourself with the difference between offset number D and the tool edge number CE. See also language commands CHKDNO, CHKDM, GETDNO, SETDNO, DZERO. The is not evaluated with the function “flat D number” and accordingly is not meaningful there. The MD can change the memory requirements: A change from “less than equal to” to “greater than” – or vice versa – in the values of both the above mentioned MDs can influence the demand for volatile memory.	
Corresponding to...	MD 18105: MM_MAX_CUTTING_EDGE_NO	
Further references:	Description of Functions: Tool Offset (W1)	

18108 MD number	**MM_NUM_SUMCORR** Additive offsets in the TO area			
Default setting: –1		Minimum input limit: –1		Maximum input limit: 9000
Changes effective after POWER ON		Protection level: 2/7		Unit: –
Data type: DWORD			Applies as of SW 5	
Meaning:	Total number of additive offsets in NCK. A setting of –1 means that the number of sum offsets equals the number of cutting edges * number of sum offsets per cutting edge. A value > 0 and < “Number of cutting edges* number of additive offsets per cutting edge” means that per cutting edge, a maximum of the “Number of additive offsets per cutting edge” can – but does not need to – be defined, i.e. the option is given for using the non-volatile memory sparingly. Only the cutting edges defined for the explicit data have a additive offset data block. Non-volatile memory is reserved. The memory requirements for additive offset are doubled if “Setup offset” is also configured and active; see MD $MN_MM__KIND_OF_SUMCORR.			
Corresponding to...	MD 18100: MM_NUM_CUTTING_EDGE_IN_TOA MD 18110: MM_MAX_SUMCORR_PER_CUTTEDGE			
Further references:	Description of Functions: Tool Offset (W1)			

18110 MD number	**MM_MAX_SUMCORR_PER_CUTTEDGE** Maximum number of sum offsets per cutting edge			
Default setting: 1		Minimum input limit: 1		Maximum input limit: 6
Changes effective after POWER ON		Protection level: 2/7		Unit: –
Data type: DWORD			Applies as of SW 1	
Meaning:	The following applies to MM_NUM_SUMCORR >0: This data does not reserve memory, but is used for monitoring purposes only. The following applies to MM_NUM_SUMCORR = –1: This data reserves non-volatile memory.			
Corresponding to...	MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA MD 18108: MM_NUM_SUMCORR			
Further references:	Description of Functions: Memory Configuration (S7)			

18112 MD number	**MM_KIND_OF_SUMCORR** Properties of additive offset in the TO area		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 0x1F	
Changes effective after POWER ON	Protection level: 2/7	Unit: –	
Data type: DWORD	Applies as of SW 5		
Meaning:	Properties of additive offsets in NCK. Bit 0=0 “Additive offsets fine” are saved when the tool data is backed up. Bit 0=1 “Additive offsets fine” are not saved when the tool data is backed up. Bit 1=0 Setup offsets are saved when the tool data is backed up. Bit 1=1 Setup offsets are not saved when the tool data is backed up. Bit 2=0 If the function tool management (TMG) or tool monitoring (TMO) are used, then setting the tool status to “active” has no effect on the “additive offsets fine”/setup offsets already in use. Bit 2=1 When the tool status is set to “active”, the existing additive offsets are set to the value 0. This does not influence the setup offsets. Bit 3=0 If the functions “TMG” = “Adapter” are in use: “Additive offsets fine”/setup offsets are transformed. Bit 3=1 “Additive offsets fine”/setup offsets are not transformed. Bit 4=0 No setup data blocks. Bit 4=1 setup compensation data blocks are created as well. The sum offset is thus the product of the setup offset sum offset fine. Changing the states of bits 0, 1, 2 and 3 does not alter the memory configuration. Changing the status of bit 4 causes the non-volatile memory to be re-configured with the next Power ON operation.		
Corresponding to...	MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA MD 18108: MM_NUM_SUMCORR MD 18110: MM_MAX_SUMCORR_PER_CUTTEDGE MD 18080: MM_TOOL_MANAGEMENT_MASK MD 20310: MC_TOOL_MANAGEMENT_MASK MD 18086: MM_NUM_MAGAZINE_LOCATION MD 18104: MM_NUM_TOOL_ADAPTER		
Further references:	Description of Functions: Tool Offset (W1)		

8.1.4 Channelspecific machine data

20096 MD number	**T_M_ADDRESS_EXT_IS_SPINO** Meaning of the address extension with T, M “tool change code”				
Default setting: 0		Minimum input limit: 0		Maximum input limit: 1	
Changes effective after: Power ON			Protection level: 2/7		Unit:
Data type: Boolean				Applies as of SW 6.1	
Meaning:	Only meaningful if functions “Tool and magazine management” / “Flat D numbers” are inactive. FALSE The contents of the address extension of the NC addresses T and M “change-command number” are not evaluated by the NCK. The PLC determines the meaning of the programmed extension TRUE The address extension of NC addresses T and M “Change command number” are interpreted as a spindle number. The NCK thus handles the extension in a way analogous to the function tool management or TMFD. The programmed D number always refers to the T number of programmed main-spindle numbers.				
Corresponding to...	MD 20090: SPIND_DEF_MASTER_SPIND MD 22550: TOOL_CHANGE_MODE MD 22560: TOOL_CHANGE_M_CODE				
Further references:					

20110 MD number	**RESET_MODE_MASK** Definition of tool length compensation selection after power-up and reset		
Default setting: 0x0	Minimum input limit: 0	Maximum input limit: 0x7FFF	
Changes effective after: RESET	Protection level: 2/7	Unit: HEX	
Data type: DWORD	Applies as of SW 2		
Meaning:	Definition of the basic PLC setting after booting and reset/end of part program with respect to the G codes (especially current level and adjustable zero offset), tool-length offset and transformation by setting the following bits (only the bits in bold are relevant for tool management): **Bit 0 Reset mode** Bit 1 Suppress auxiliary function output on tool selection **Bit 2 Selection of the reset response after Power ON; e.g. of tool offset** **Bit 3 Selection of the reset response at the end of the test mode for active tool offsets. This bit is relevant only if bits 0 and 6 are set.** **It defines the relation for the "current setting for the active tool length compensation";** **- The program which was active at the end of the test operation** **- The program which was active before the test operation was activated** Bit 4 Reserved! Setting is now made in $MC_GCODE_RESET_MODE[..] Bit 5 Reserved! Setting is now made in $MC_GCODE_RESET_MODE[..] **Bit 6 Reset response Active tool length compensation** Bit 7 Reset response Active kinematic transformation Bit 8 Reset response Coupled-motion axes Bit 9 Reset response Tangential follow-up Bit 10 Reset response Synchronous spindle Bit 11 Reset response Revolutional feedrate Bit 12 Reset response Geo-axis replacement Bit 13 Reset response Master value coupling Bit 14 Reset response Basic frame Bits 4 to 11 are evaluated only if bit 0=1. Bit 15 Function for electronic gears, not relevant for tool management. Bit 16=0 After end of program/reset, the number given by the MD SPIND_DEF_MASTER_SPIND is the number of the master spindle Bit 16=1 The programmed value of **"SETM"** is retained after end of program/reset Bit 17=0 After end of program/reset, the toolholder number provided in MD TOOL_MANAGEMENT_TOOLHOLDER is the number of the master toolholder Bit 17=1 The programmed value of **SETMH** is retained after end of program/reset Both bits are only meaningful if bit 0=1 is set as well. Bit value=0 is set so that the behavior applicable up to now is retained with bit 0=1. (The effect of bit 0=0 was and is that the values programmed for SETMTH/SETMS are retained at end of program.)		
Corresponding to...	MD 20120: TOOL_RESET_VALUE MD 20130: CUTTING_EDGE_RESET_VALUE MD 20150: GCODE_RESET_VALUES MD 20152: GCODE_RESET_MODE MD 20140: TRAFO_RESET_VALUE MD 20112: START_MODE_MASK MD 20121: TOOL_PRESEL_RESET_VALUE MD 20118: GEOAX_CHANGE_RESET		
Further references:	Description of Functions: Coordinate Systems (K2)		

20112 MD number	**START_MODE_MASK** Definition of tool length compensation selection after part program start				
Default setting: 0x400		Minimum input limit: 0		Maximum input limit: 0x7FFF	
Changes effective after: RESET			Protection level: 2/7		Unit: HEX
Data type: DWORD				Applies as of SW 3	
Meaning:	Definition of the basic PLC control for part-program start with respect to the G codes (especially current level and adjustable zero offset), tool-length offset, transformation and axis coupling setting the following bits (only the bits in bold are relevant for tool management): **Bit 0 Not assigned: $MC_START_MODE_MASK is evaluated every time the part program is started** Bit 1 Suppress auxiliary function output on tool selection Bit 4 Start response G code Current plane Bit 5 Start response G code Settable zero offset **Bit 6 Start response Active tool length compensation** Bit 7 Start response Active kinematic transformation Bit 8 Start response Coupled-motion axes Bit 9 Start response Tangential follow-up Bit 10 Start response Synchronous spindle Bit 11 Reserved Bit 12 Start response “Geo-axis replacement” Bit 13 Start response “Master value coupling” Bit 14 Start response “Basic frame” Bit 15 Function for electronic gears, not relevant for tool management. Bit 16=0 The current value **SETMS** is retained (depends on the settings in RESET_MODE_MASK) Bit 16=1 At program start, the spindle defined in MD: $MC_SPIND_DEF_MASTER_SPIND becomes the master spindle. Bit 17=0 The current value **SETMH** is retained (depends on the settings in RESET_MODE_MASK) Bit 17=1 At program start, the number defined in MD: $MC_Tool_Management_Toolholder becomes the number of the master toolholder Bit value=0 is set so that the behavior applicable up to now is retained.				
Corresponding to...	MD 20120: TOOL_RESET_VALUE MD 20130: CUTTING_EDGE_RESET_VALUE MD 20150: GCODE_RESET_VALUES MD 20152: GCODE_RESET_MODE MD 20140: TRAFO_RESET_VALUE MD 20112: START_MODE_MASK MD 20121: TOOL_PRESEL_RESET_VALUE MD 20118: GEOAX_CHANGE_RESET				
Further references:	Description of Functions: Coordinate Systems (K2)				

20120 MD number	**TOOL_RESET_VALUE (only without tool management)** Tool whose length compensation is selected during power-up				
Default setting: 0		Minimum input limit: 0		Maximum input limit: 32000	
Changes effective after: RESET			Protection level: 2/7		Unit: –
Data type: DWORD				Applies as of SW 2	
Meaning:	This data is valid only when the tool/magazine managment is not active. Definition of tool of which length compensation is selected during power-up and on Reset/ end of part program as a function of MD 20110 and on start of part program as a function of MD 20112.				
Corresponding to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK				
Further references:	Description of Functions: Coordinate Systems (K2)				

20121 MD number	TOOL_PRESEL_RESET_VALUE Preselect tool on Reset	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 32000
Changes effective after: RESET	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 4.1	
Meaning:	This data is only valid without tool management. Definition of the preselected tool with MD 20310=1. A tool is preselected after power-up and on Reset or end of part program as a function of MD 20110 and on start of part program as a function of MD 20112.	
Corresponding to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK	
Further references:	Description of Functions: Coordinate Systems (K2)	

20122 MD number	TOOL_RESET_NAME Active tool at reset/start and tool management	
Default setting: –	Minimum input limit: –	Maximum input limit: –
Changes effective after: RESET	Protection level: 2/7	Unit: –
Data type: STRING	Applies as of SW 3.2	
Meaning:	This data is valid only if the TM function is active. Definition of the tool with which tool length compensation is selected during power-up and on Reset or end of part program as a function of MD 20110 RESET_MODE_MASK and on start of part program as a function of MD 20112 START_MODE_MASK.	
Corresponding to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK MD 20124: TOOL_MANAGEMENT_TOOLHOLDER MD 20130: CUTTING_EDGE_RESET_VALUE	
Further references:		

20123 MD number	USEKT_RESET_VALUE Effective value of $P_USEKT at RESET	
Default setting: 0x0, ...	Minimum input limit: 0	Maximum input limit: 0xF
Changes effective after: RESET	Protection level: 2/7	Unit:
Data type: DWORD	Applies as of SW 6.1	
Meaning:	Definition of the tool-technology group during booting and for a reset or end of main program as a dependency on MD $MC_RESET_MODE_MASK and for a part-program start in dependency on MD $MC_START_MODE_MASK. This data is only valid for an active tool management (TMMG) and/or active tool-monitoring function.	
Corresponding to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK	
Further references:		

20124 MD number	TOOL_MANAGEMENT_TOOLHOLDER Toolholder number	
Default setting: 0,0,0,...	Minimum input limit: 0	Maximum input limit: 16
Changes effective after: Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 3.2.	
Meaning:	This data is relevant only when the tool management function is active. Definition of whether toolholder no. or spindle no. is given in order to defined the location of use for a a tool to be loaded. The tool management must know on which tool holder the tool is to be loaded. If the MD is greater than 0, the spindle numbers $TC_MPP5 are interpreted as toolholder numbers. The automatic address extension of T and from M06 is then the value for this MD and no longer the value of MD 20090 SPIND_DEF_MASTER_SPIND. With machines where there are several toolholders without a designated master spindle, the MD serves as the default value in order to determine the toolholder for the tool shall be loaded. Tool holder n is declared the master tool holder with SETMTH(n). Tools which are to be loaded in a buffer location of the spindle type and which have the value $TC_MPP5=n correct the tool path. Tools with a value not equal to n have no effect on the offset. The command SETMTH is used to declare the toolholder defined in the MD as the master toolholder again. When defining the magazine locations of internal magazines, spindle locations –$TC_MPP1=2=spindle-location can be assigned a location type index ($TC_MPP5). This assignes a concrete tool holder to the location.	
Corresponding to...	MD 20090 SPIND_DEF_MASTER_SPIND MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK MD 20310: MC_TOOL_MANAGEMENT_MASK MD 18080: MM_TOOL_MANAGEMENT_MASK	
Further references:		

20126 MD number	TOOL_CARRIER_RESET_VALUE Operative toolholder on Reset	
Default setting: 0	Minimum input limit: 0.0	Maximum input limit: plus
Changes effective after: Reset	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 4.1	
Meaning:	Definition of the toolholder with which the tool length compensation is selected during booting and for a reset or an end of the part program as a dependency on the machine data $MC_RESET_MODE_MASK and for a main program start as a dependency on the machine data $MC_START_MODE_MASK.	
Corresponding to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK	
Further references:	Description of Functions: Tool Offset (W1)	

20128 MD number	**COLLECT_TOOL_CHANGE** Collect tool changes during block search				
Default setting: 1		Minimum input limit: –		Maximum input limit: –	
Changes effective after: Immediately			Protection level: 2/7		Unit: –
Data type: BOOLEAN				Applies as of SW 4.3	
Meaning:	This MD is only relevant when tool management is active. It determines whether the tool change M code defined in MD 22560: TOOL_CHANGE_M_CODE will be collected in the block search with calculation. TRUE: Tool change M code is collected FALSE: Tool change M code is not collected The tool determined in the search run is displayed and treated as the current tool. The T-number output is not affected by this. The tool compensation data determined on the NCK side are effective. There is on change in the magazine data, etc. Without tool management the tool change M code is not collected if it is not assigned to any auxiliary function group.				
Corresponding to...	MD 22560: TOOL_CHANGE_M_CODE				
Further references:					

20130 MD number	**CUTTING_EDGE_RESET_VALUE** Cutting edge effective at reset				
Default setting: 0		Minimum input limit: 0		Maximum input limit: 32000	
Changes effective after: RESET			Protection level: 2/7		Unit: –
Data type: DWORD				Applies as of SW 2	
Meaning:	Definition of the tool cutting edge with which the tool length compensation is selected during booting and for a reset or an end of the part program as a dependency on the machine data $MC_RESET_MODE_MASK and for a main program start as a dependency on the machine data $MC_START_MODE_MASK. When tool management is active and bits 0 and 6 set in $MC_RESET_MODE_MASK, the last offset of the tool which was active on power-off – generally the tool in the spindle – is operative after power-on.				
Corresponding to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK				
Further references:	Description of Functions: Coordinate Systems (K2)				

20132 MD number	**SUMCORR_RESET_VALUE** Additive sum effective at reset		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 6	
Changes effective after: RESET	Protection level: 2/7	Unit: –	
Data type: DWORD	Applies as of SW 5		
Meaning:	Definition of the additive offset with which the additive offset is selected during booting and for a reset or an end of the part program as a dependency on the machine data $MC_RESET_MODE_MASK and for a main program start as a dependency on the machine data $MC_START_MODE_MASK. Machine data $MC_MM_NUM_SUMCORR determines the maximum meaningful value which can be entered.		
Corresponding to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK		
Further references:	Description of Functions: Tool Offset (W1)		

20140 MD number	**TRAFO_RESET_VALUE** Active transformation at RESET		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 8	
Changes effective after: RESET	Protection level: 2/7	Unit: –	
Data type: BYTE	Applies as of SW 2		
Meaning:	Definition of the transformation data set selected during booting and for a reset or an end of the part program. In conjunction with machine data $MC_RESET_MODE_MASK and for a part program start in conjunction with machine data $MC_START_MODE_MASK		
Corresponding to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK		
Further references:	Description of Functions: Axes, coordinate systems,... (K2)		

20150 MD number	**GCODE_RESET_VALUES[n]** Initial setting of G group		
Default setting :{2, 0, 0, 1, 0...}	Minimum input limit: 0.0	Maximum input limit: plus	
Changes effective after: RESET	Protection level: 2/7	Unit: –	
Data type: BYTE	Applies as of SW 1		
Meaning:	Definition of the G code which is to become effective at boot or reset/end of part program – dependent on machine data $MC_RESET_MODE_MASK – and at part program start – dependent on machine data $MC_START_MODE_MASK. The G code index defined in coenum.hh must be specified in the respective groups as preset value.		
Corresponding to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK		
Further references:	(K1, G2)		

20152 MD number	**GCODE_RESET_MODE[n]** Reset behavior of G groups	
Default setting: –	Minimum input limit: 0	Maximum input limit: 1
Changes effective after: RESET	Protection level: 2/7	Unit: –
Data type: BYTE	Applies as of SW 4.4	
Meaning:	This MD is only evaluated if bit 0 is set in $MC_RESET_MODE_MASK. For each entry in MD $MN_GCODE_RESET_VALUES you determine whether at reset/part program end the setting is to be applied again as set in $MN_GCODE_RESET_VALUES (MD=0), or the currently active setting is to be maintained (MD=1).	
Corresponding to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK	
Further references:	Description of Functions: Axes, coordinate systems,... (K2)	

20270 MD number	**CUTTING_EDGE_DEFAULT** Basic setting of tool cutting edge after tool change without programming	
Default setting: 1	Minimum input limit: –2	Maximum input limit: 32000
Changes effective after: Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 5.2	
Meaning:	If no cutting edge is programmed after a tool change, then the edge number preset in CUTTING_EDGE_DEFAULT is applied. Wert = 0: No cutting edge is initially active after a tool change. Cutting-edge selection only takes at D programming. Value = 1: MD_SLMAXCUTTINGEDGENUMBER, number of cutting edge (up to SW 4 = 9) Value = –1: Tool edge number of old tool also applies for new tool Value = –2: Cutting edge (compensation) of the old tool remains active until D is programmed.	
Corresponding to...		
Further references:	Description of Functions: Tool Offset (W1)	

20272 MD number	**SUMCORR_DEFAULT** Basic setting of additive offset without programming	
Default setting: 0	Minimum input limit: –1	Maximum input limit: 6
Changes effective after: Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 5	
Meaning:	The MD defines the number of the additive offset for the cutting edge that is active when a new cutting-edge compensation is activated without a programmed DL value. Machine data $MN_MAX_SUMCORR_PERCUTTING_EDGE determines the maximum meaningful value which can be entered. Value Meaning > 0 Number of additive offset = 0 No additive offset active with D programming = –1 The additive sum number for the previously programmed D is used.	
Corresponding to...	MD 20270: CUTTING_EDGE_DEFAULT	
Further references:	Description of Functions: Tool Offset (W1)	

Notice

The output of the DL number is controlled by the MD AUXFU_DL_SYNC_TYPE.

20310 MD number	**TOOL_MANAGEMENT_MASK** Channel-specific activation of tool management functions				
Default setting: 0x0, ...		Minimum input limit: 0		Maximum input limit: 0xFFFFF	
Changes effective after: Power ON			Protection level: 2/7		Unit: HEX
Data type: DWORD				Applies as of SW 2	
Meaning:	MD = 0: Tool management inactive Bit0=1: Tool management active The tool-management functions are enabled for the current channel. Bit 1=1: Tool management monitoring functions active The functions for monitoring tools (tool life and workpiece count) are enabled. Bit2=1: OEM functions active The memory for user data can be utilized (see also MD 18090 to 18098) Bit3=1: Consider adjacent location active **Bits 0 to 3** must be set identically to MD 18080 MM_TOOL_MANAGEMENT_MASK.				
Meaning:	Bit 4=1: The PLC has the option to request another T preparation with modified parameters. Ackonowledgement status “2”, “7” and “103” is enabled with this bit. This causes the tool selection to be recalculated in the NCK.				
	Bits 5 to 8 Bit 5 and bit 7 refer to the main spindle Bit 6 and bit 8 refer to the secondary spindles Bit 5 = 1 The command output is considered completed if the **internal transport acknowledgement + the transport acknowledgement** are present, i.e. if the command was received by the basic PLC program (see Section 3.16.2) Bit 19=1 additionally allows the block change to be prevented (**main run**) until the required acknowledgements are received. Bit 7 = 1 The command output is not considered completed until the **end acknowledgement** from the PLC is received, i.e. the command was acknowledged by the PLC user program with status “1” (see Section 3.16.2) Bit 19=1 additionally allows the block change to be prevented (**main run**) until the required acknowledgements are received. Bit 5 and bit 7 (alternatively bit 6 and bit 8) are mutually exclusive! Only the following combinations are permitted: Bit 5 0 1 0 Bit 7 0 0 1 With the default setting, i.e. bit 5 to 8 = 0, synchronization is performed in the block in which a cutting edge was first selected. Setting these bits delays block processing.				
Meaning:	Bit 9: Reserved for testing purposes Can also be used by the machine manufacturer in the test phase, as long as the PLC program does not yet change tools				
Meaning:	Bit 10=1: M06 is delayed until the preparation is taken over by the PLC user program. The change command is only output when the preparation acknowledgement is received. This can be, for example status “1” or “105”. Bit 10=0 The change command is output without delay, immediately after the preparation command.				

20310	TOOL_MANAGEMENT_MASK	
MD number	Channel-specific activation of tool management functions	
Meaning:	Bit 11=1:	The tool preparation command (PLC command numbers = 2, 4, 5) is carried out even if the same tool preparation command has taken place! (Commands 4, 5 contain the tool preparation) Example: (tool change takes place with M6 (PLC command number=3): T=“TL1”; Tool preparation M6; Tool change T=“TL2”; 1st tool preparation after M6 (for the same toolholder) ; is always output to PLC T=“TL2”; 2nd tool preparation, is only output as command to PLC if bit 11 = 1 ; This tool preparation counts as the first one if the status of the tool has changed since the previous tool preparation such that it can no longer be used. A possible reason for this would be e.g. asynchronous unloading of a tool. This tool preparation them attempts to select a replacement tool.
	Bit 11=0:	The preparation command can be output only once for each tool.
Meaning:	Bit 12=1:	The tool preparation command (PLC command numbers = 2, 4, 5) is carried out even if the tool is already positioned in the spindle/toolholder. T=“TL1”; Tool preparation M6; Tool change T=“TL1”; Tool is already placed on the toolholder ; 1st tool preparation after M6 (for the same toolholder) ; is only output to PLC if bit 12 = 1 ; A tool that cannot be used (e.g. disabled due to tool monitoring) on the toolholder does not count as if placed on the toolholder. This tool preparation them attempts to select a replacement tool. T=“TL2”; 2nd tool preparation – the following rules applies for bit 11 for output
	Bit 12=0:	The preparation command is not executed if the tool is already inserted in the spindle.
	Bit 13=1:	On Reset the commands are transferred from the diagnostics buffer to the passive file system (TCTRA xx.MPF under part program). This file is required by the Hotline. The tool sequences are only stored in the diagnostics buffer on systems with sufficient memory (NCU572, NCU573).
	Bit 14=1:	Reset mode Tool and offset selection according to the settings in MD: $MC_RESET_MODE_MASK and $MC_START_MODE_MASK. For information on how to do this, refer to Section 3.2.22.
	Bit 14=0:	No reset mode
Meaning:	Bit 15=1:	The tool is not returned if several preparation commands have been issued (Tx–>Tx).
	Bit 15=0:	Tool is returned from any defined buffers.
	Bit 16=1:	T=Location number is active
	Bit 16=0:	T=“Tool name”
	Bit 17=1:	Tool life decrementation can be started/stopped via the PLC in channel DB 2.1...DBx 1.3.
	Bit 18=1:	Activation of monitoring “last tool of tool group” Bit 18 extends the search for a suitable tool, especially if there are many disabled replacement tools.
	Bit 18=0:	No monitoring for “Last tool in tool group”
	Bit 19=1:	The synchronizations defined by bits 5...8 are relative to the main run block, i.e. there is no block change until the required acknowledgements are received Bit 19 in conjunction with bits 5, 6, 7, 8 set delays the block processing.
	Bit 19=0:	The synchronizations defined by bits 5...8 are relative to the tool management command output, i.e. there is no block change delay

20310 MD number	TOOL_MANAGEMENT_MASK Channel-specific activation of tool management functions
Meaning:	Bit 20=0: The commands generated on PLC signal “program testing active” are not output to the PLC. NCK acknowledges the commands itself. Magazine and tool data are not changed. Bit 20=1: The commands generated on PLC signal “program testing active” are output to the PLC. Depending on the type of acknowledgement, tool/magazine data can be changed in the NCK. If the acknowledgement parameters for the “target magazine” are set to the same values as the “source magazine”, the tool is not transported and thus no data modified in the NCK.
	Bit 21=0: Default setting: Ignore the tool status “W” at tool selection Bit 21=1: Tools in status “W” cannot be selected by another tool change, tool preparation command.
	Bit 22=1 “Tool subgroups” function $TC_TP11[x] is the grouping or selection parameter (see Section 5.3.1, parameter $TC_TP11).
	Bit 23=0 Default setting The tool management selects the tool optimally securely in the main run, i.e. the interpreter may have to wait for the end of the tool selection for offset selection. Bit 23=1 For simple applications The interpreter selects the tool itself, i.e. no synchronization is required with the main run for offset selection. (If the tool becomes no longer useable after selection but before loading, an uncorrectable alarm may result.) Bit 24=0 Default setting If the PLC commands 8 and 9 (asynchronous transfer) want to move a tool to a location that is reserved for another tool, this is rejected and an alarm is issued. Bit 24=1 If the PLC commands 8 and 9 are to move a tool to a location that is reserved for another tool with “Reserved for tool from buffer” (bit values=“H4”), this is possible. This location reservation is then removed before the movement is executed (“Reserved for new tool to be loaded” (bit value=“H8”) remains effective).
Corresponding to...	MD 18080: MM_TOOL_MANAGEMENT_MASK MD 20320: TOOL_TIME_MONITOR_MASK MD 20122: MC_TOOL_RESET_NAME MD 20110: MC_RESET_MODE_MASK MD 20124: MC_TOOL_MANAGEMENT_TOOLHOLDER MD 22560: TOOL_CHANGE_M_CODE
Further references:	

20320 MD number	TOOL_TIME_MONITOR_MASK Time monitoring for tool in spindle	
Default setting: 0x0	Minimum input limit: –	Maximum input limit: –
Changes effective after: Power ON	Protection level: 2/7	Unit: HEX
Data type: DWORD	Applies as of SW 2	
Meaning:	Activation of tool time monitoring function for spindle 1....x. As soon as the path axes are moved (not with G00), the time monitoring data for the tool loaded in the appropriate spindle are updated. Bit 0...x–1: Monitoring of active tool in spindle 1...x	
Corresponding to...		
Further references:	Description of Functions: Memory Configuration (S7)	

22550 MD number	**TOOL_CHANGE_MODE** New tool offset for T or M function	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 1
Changes effective after: Power ON	Protection level: 2/7	Unit: –
Data type: BYTE	Applies as of SW 1.1	
Meaning:	This machine data determines the mode of tool change **MD: TOOL_CHANGE_MODE = 0** The new tool data becomes effective directly when T or D is programmed. This setting is used mainly for turning machines with tool revolver. If there is no D programmed in the block with T, the tool offset which is defined in $MC_CUTTING_EDGE_DEFAULT becomes effective. The function “Manual tools” is not enabled for this case. **MD: TOOL_CHANGE_MODE = 1** The new tool is prepared for changing with the T function. This setting is used mainly on milling machines with a tool magazine, in order to bring the new tool into the tool change position without interrupting the machining process. With the M function set in MD: TOOL_CHANGE_MODE the old tool is removed from the spindle and the new tool is loaded into the spindle. According to DIN 66025, this tool change must be programmed with the M function M06. If there is no D programmed in the block with M, the tool offset which is defined in $MC_CUTTING_EDGE_DEFAULT becomes effective.	
Corresponding to...	MD 22560: TOOL_CHANGE_M_CODE	
Further references:	Description of Functions: Tool Offset (W1)	

22560 MD number	**TOOL_CHANGE_M_CODE** M function for tool change	
Default setting: 6	Minimum input limit: 0	Maximum input limit: 99999999
Changes effective after: Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 1	
Meaning:	This machine data is effective only when TOOL_CHANGE_MODE = 1. If the T function is only used to prepare a new tool for a tool change (this setting is used mainly on milling machines with a tool magazine, in order to bring the new tool into the tool change position without interrupting the machining process), another M function must be used to trigger the tool change. The M function entered in TOOL_CHANGE_M_CODE triggers the tool change (remove old tool from spindle and insert new tool in spindle). This tool change is required to be programmed with M function M06, in accordance with DIN66025.	
Corresponding to...	MD 22550: TOOL_CHANGE_MODE	
Further references:	Functional description for tool offset (W1)	

22562 MD number	TOOL_CHANGE_ERROR_MODE Response when errors occur at tool change

Default setting: 0x0	Minimum input limit: 0	Maximum input limit: 0x1F
Changes effective after: Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies as of SW 5.1	

Meaning:

Bit 0=0 Standard response: Stop on faulty NC block
Bit 0=1: If the error occurs in the block containing the tool change preparation command, the alarm activated by the preparation command (T) is delayed until the program run reaches the point at which the associated tool change command (M06) is interpreted. Only then is the alarm output that is triggered by the preparation command. The operator can make corrections in this block. When execution of the program is continued, the faulty NC block is interpreted again and the preparatory command is internally executed again automatically.
This machine data is only relevant only if the setting MD 22550: TOOL_CHANGE_MODE = 1 is used.

Bit 1 is only meaningful if tool management is active.
Bit 1=0 Standard response: During the tool-change preparation only those tools are recognized whose data are assigned to a magazine.
Bit 1=1 Manual tools can also be loaded at change.
A tool will also be loaded at change if its data is registered in the NCK, but not assigned to a magazine location. The data is then assigned to the programmed toolholder.
The user is prompted to place tools into the toolholder or remove tools from it.
The function is only executed if TOOL_CHANGE_MODE = 1.

Bit 2 Qualifying the offset programming
Bit 2=0 Active D no. > 0 and active T no. = 0 results in offset 0
Active D no. > 0 and active D no.=0 result in additive offset 0
Bit 2=1 Active D no. > 0 and active T no. = 0 generates an alarm message
Active D no. > 0 and active D no.=0 generates an alarm message

Bits 3 and 4 are only meaningful if tool management is active.
Function:
Control of behavior of Init block generation at program start if disabled tool is positioned in the spindle and needs to be activated.
See also: MD 20112: START_MODE_MASK,
MD 20110: RESET_MODE_MASK
Note:
At RESET the response of “keep disabled tool on spindle active” is not affected.

Bit 3=0 Default: If the tool on the spindle is blocked: Generate a tool-change command that requests a spare tool. If there is no replacement, an alarm is produced.
Bit 3=1 The blocked state of the spindle tool is ignored. The tool is active. The following part program should then be formulated so that no parts are machined with the blocked HARMONIZE tool.

Bit 4=0 Default: The attempt is made to activate the spindle tool or the replacement tool
Bit 4=1 If the tool on the spindle is blocked, Initsat TO is programmed in the start.
The following statements are given for the combination of bit 3 and bit 4:
0 / 0: Response as before, automatic change at NC Start is the disabled tool is in the spindle
1 / 0: Is not automatically changed
0 / 1: A TO is generated automatically for a disabled tool in the spindle at NC Start
1 / 1: No statement

22562 MD number	TOOL_CHANGE_ERROR_MODE Response when errors occur at tool change	
	Bit 5	Reserved
	Bit 6=0	Default: with TO or D0 only TO or D0 are programmed. In other words, the MD $MC_CUTTING_EDGE_DEFAULT, $MC_SUMCORR_DEFAULT define the value of D, DL when TO is programmed. Example: $MC_CUTTING_EDGE_DEFAULT=1, $MC_SUMCORR_DEFAULT=2, $MC_TOOL_CHANGE_MODE=0 (tool change with T programming) N10 TO; T no. 0 has active number D1 and DL=2 which results in offset zero. If bit 2 is additionally set: Programming of a) TO; for tool deselection b) D0; for offset deselection generates an alarm, if a) At least one of MD $MC_CUTTING_EDGE_DEFAULT, $MC_SUMCORR_DEFAULT is not equal to zero (TO D0 DL=0 is the correct programming). b) The MD $MC_SUMCORR_DEFAULT is not equal to zero (D0 DL=0 is the correct programming).
	Bit 6=1	Controls the NCK behavior with programming of (x, y, z all greater than zero), if at least one of MD $MC_CUTTING_EDGE_DEFAULT, $MC_SUMCORR_DEFAULT is not equal to zero. a) Tx Dy -> TO with TO, D0 or D0 DL=0 is automatically programmed in the NCK; i.e. values not equal to zero for MD $MC_CUTTING_EDGE_DEFAULT, $MC_SUMCORR_DEFAULT are processed as if the value were equal to zero. b) Tx Dy -> TO Dy, or TO DL =z, or TO Dy DL=z, or TO D0 DL=z explicitly programmed values of D, DL are not influenced. c) Dy DL=z -> D0 with D0, DL=0 is automatically programmed in the NCK; i.e. values not equal to zero for MD $MC_SUMCORR_DEFAULT are processed as if the value were equal to zero. d) Dy DL=z -> D0 DL=z explicitly programmed values of DL are not affected. If bit 2 is additionally set: you only need to program TO/D0 for tool/offset deselection for no alarm to be issued. The statements relative to $MC_SUMCORR_DEFAULT or DL are only valid if the additive offset function is active (see $MN_MM_TOOL_MANAGEMENT_MASK, bit 8).
	Bit 7=0	Programming Tx checks whether a tool with the T number x is known in the TO unit of the channel. If it is not, processing stops in this block and alarm 17190 is issued.
	Bit 7=1	Only if tool basic functionality is active ($MC_TOOL_MANAGEMENT_MASK, bit 0,1=0) and ($MN_MM_TYPE_OF_CUTTING_EDGE=0): If Tx is programmed, any unknown Tx is first ignored and the alarm for the preparation command (Tx) is ignored until D selection is interpreted in the program execution. Only then is alarm 17191 output that was triggered by the preparation command. This means that in this block, the operator could use the D selection to make corrections. When execution of the program is continued, the faulty NC block is interpreted again and the preparation command is internally executed again automatically. (If the programmer wants to program Cutting-Edge-Default=0 or =-2 or D0 for programmatical reasons, otherwise the D from Cutting-Edge-Default is deselected at tool change.) This variant can be required if you want to programm “Tool number=Location” (turret as toolholder) without tool management. The turret can only be positioned at a location for which there is no tool defined (yet). If bit 0=1 is set, this bit is irrelevant. This behavior is compatible with software versions prior to 6.5.13.
Further references:	Description of Functions: Tool Offset (W1)	

20090 MD number	**SPIND_DEF_MASTER_SPIND** Initial setting of master spindle in channel		
Default setting: 1, 1, 1, 1,...	Minimum input limit: 1	Maximum input limit: 15	
Changes effective after: Power ON	Protection level: 2/7	Unit: –	
Data type: BYTE	Applies as of SW 1		
Meaning:	Definition of master spindle in channel. The number of the spindle is set. Example: 1 corresponds to spindle S1. When S is programmed, the current master spindle is automatically addressed. The SETMS(n) command can be programmed to declare the spindle number as the master spindle. SETMS declares the spindle defined in the MD to be the master spindle again.		
Corresponding to...			
Further references:	Description of Functions: Spindles (S1)		

28085 MD number	**LINK_TOA_UNIT** Allocation of a TO unit to a channel		
Default setting: 1, 2, 3, 4, 5, ...	Minimum input limit: 1	Maximum input limit: 10	
Changes effective after: Power ON	Protection level: 2/7	Unit:	
Data type: DWORD	Applies as of SW 2		
Meaning:	The area TO includes all tools, magazine, ... data blocks known to the NCK. The maximum number of units in the TO area match the maximum number of channels. If LINK_TOA_UNIT = default, then each channel is individually assigned a TO unit. The channel is assigned the TO unit i when LINK_TOA_UNIT = i . It is thus possible to assign one TO unit to several channels. *Notice* The upper limit value does not imply that the value is always meaningful or without conflict. If one channel (the first) is active and the other not on a system with a total of 2 channels, the MD on channel 1 can be formally set to a value of 2. However, the NCK cannot work with this setting because it would mean that channel 1 possesses no data blocks for tool offsets since a channel with Id=2 does not exist. The NCK detects this conflict during power-on or a warm restart and reacts by independently changing the (incorrect) setting to the default setting for the MD.		
Corresponding to...			
Further references:	Description of Functions: Memory Configuration (S7)		

8.1.5 Machine data for function replacement

10715 MD number	**M_NO_FCT_CYCLE** M function for cycle call				
Default setting: -1		Minimum input limit: -1		Maximum input limit: -	
Changes effective after POWER ON			Protection level: 2/4		Unit: -
Data type: DWORD				Applies as of SW 5.2	
Meaning:	M number via which a subroutine is called. The name of the subroutine is stored in $MN_M_NO_FCT_CYCLE_NAME. If programmed in a part program with the defined by $MN_M_NO_FCT_CYCLE, then the subroutine defined in M_NO_FCT_CYCLE_NAME is started at the end of block. If the M function is re-programmed in the subroutine, then there is no longer any replacement by a subroutine call. $MN_M_NO_FCT_CYCLE acts both in the Siemens mode G290 as well as in the external language mode G291. A subroutine call must not be superimposed on M functions with predetermined meaning. Alarm 4150 is generated in case of a conflict: - M0 to M5, - M17, M30, - M40 to M45, - M function for selecting spindle/axis mode according to $MC_SPIND_RIGID_TAPPING_M_NR (default M70) - M functions for nibbling/punching as configured in $MC_NIBBLE_PUNCH_CODE if activated by $MC_PUNCHNIB_ACTIVATION. - for applied external language ($MN_MM_EXTERN_LANGUAGE) M19, M96-M99. Exception: the M functions defined by $MC_TOOL_CHANGE_M_CODE for the tool change. The subroutines configured with $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be effective in one block (part-program line) at the same time, i.e. max. one M/T function replacement can be effective per block. Neither an M98 nor a modal subroutine call may be programmed in the block with the M-function replacement. It is also illegal to program a subroutine return jump or end of part program. Alarm 14016 is generated if these conventions are not observed.					
Corresponding to...					
Further references:	ISO dialects for Sinumerik (FBFA)				

10716 MD number	**M_NO_FCT_CYCLE_NAME** Name for tool-changing cycle with M functions from MD $MN_NO_FCT_CYCLE				
Default setting: –		Minimum input limit: –		Maximum input limit: –	
Changes effective after POWER ON			Protection level: 2/4		Unit: –
Data type: STRING				Applies as of SW 5.2	
Meaning:	The cycle name is stored in the machine data. This cycle is called if the the M function was programmed from the machine $MN_M_NO_FCT_CYCLE. If the M function is programmed in a motion block, then the cycle is executed after the motion. $MN_M_NO_FCT_CYCLE acts both in the Siemens mode G290 as well as in the external language mode G291. If a T number is programmed in the calling block, the programmed T number can be queried in the cycle under the variable $P_TOOL. The subroutines configured with $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be effective in one block at the same time, i.e. max. one M/T function replacement can be effective per block. Neither an M98 nor a modal subroutine call may be programmed in the block with the M function replacement. It is also illegal to program a subroutine return jump or end of part program. Alarm 14016 is generated if these conventions are not observed.				
Corresponding to...					
Further references:	ISO dialects for Sinumerik (FBFA)				

10717 MD number	**T_NO_FCT_CYCLE_NAME** Name for tool-changing cycle with T number				
Default setting: –		Minimum input limit: –		Maximum input limit: –	
Changes effective after POWER ON			Protection level: 2/4		Unit: –
Data type: STRING				Applies as of SW 5.2	
Meaning:	This machine data defines the name of a tool cycle that is called when T is programmed. If the T number is programmed in a motion block, then the cycle is executed after the motion. The programmed T number can be scanned in the cycle with $C_T. $MN_T_NO_FCT_CYCLE_NAME acts both in the Siemens mode G290 as well as in the external language mode G291. The subroutines configured with $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be effective in one block at the same time, i.e. max. one M/T function replacement can be effective per block. Neither an M98 nor a modal subroutine call may be programmed in the block with the T function replacement. It is also illegal to program a subroutine return jump or end of part program. Alarm 14016 is generated if these conventions are not observed.				
Corresponding to...					
Further references:	ISO dialects for Sinumerik (FBFA)				

10718 MD number	**M_NO_FCT_CYCLE_PAR** M function replacement with parameters				
Default setting: –		Minimum input limit: –1		Maximum input limit: –	
Changes effective after POWER ON			Protection level: 7/2		Unit: –
Data type: DWORD				Applies as of SW	
Meaning:	If a M function replacement was configured with $MN_M_NO_FCT_CYCLE[n] / $MN_M_NO_FCT_CYCLE_NAME[n], then $MN_M_NO_FCT_CYCLE_PAR can be used for specifying parameter passing for one of these M functions per system variable as is the case for the T function replacement. The parameters stored in the system variables always refer to the part program line in which the M function to be replaced is programmed. The following system variables are available: $C_ME : Address extension of substituted M function $C_T_PROG: TRUE if address T has been programmed $C_T: Value of address T (integer) $C_TE: Address extension of address T $C_TS_PROG: TRUE if address TS has been programmed $C_D_PROG: TRUE if address D has been programmed $C_D: Value of address D $C_DL_PROG: TRUE if address DL has been programmed $C_DL: Value of address DL				
Corresponding to...					
Further references:	ISO dialects for Sinumerik (FBFA)				

10719 MD number	**T_NO_FCT_CYCLE_MODE** Parameter settings of T function substitution				
Default setting: 0		Minimum input limit: 0		Maximum input limit: 7	
Changes effective after POWER ON			Protection level: 7/2		Unit: –
Data type: DWORD				Applies as of SW	
Meaning:	Processing of the substitution program is parameterized in this MD for the tool selection/ tool offset selection. Bit 0 = 0: The D or DL number is passed to the substitution program (default value) Bit 0 = 1: The D or DL number is not passed to the substitution program if the following conditions are fulfilled: $MC_TOOL_CHANGE_MODE = 1 programming of D/DL with T or M function with which the tool-changing cycle is called, in one part program line Bit 1 = 0: Execution of substitution subroutine at end of block (default value) Bit 1 = 1: Execution of substitution subroutine at start of block Bit 2 = 0: Execution of substitution subroutine according to the setting for bit 1 Bit 2 = 1: Execution of substitution subroutine at start of block and end of block				
Corresponding to...					
Further references:	ISO dialects for Sinumerik (FBFA)				

11717 MD number	**D_NO_FCT_CYCLE_NAME** Subroutine name for D function replacement				
Default setting: –1		Minimum input limit: –		Maximum input limit: –	
Changes effective after POWER ON			Protection level: 7/2		Unit: –

11717 MD number	D_NO_FCT_CYCLE_NAME Subroutine name for D function replacement
Data type: STRING	Applies as of SW
Meaning:	Cycle name for D function replacement routine If a D function is programmed in a part program block, the subroutine defined with $MN_D_NO_FCT_CYCLE_NAME is called in accordance with the machine data $MN_T_NO_FCT_CYCLE_NAME, $MN_T_NO_FCT_CYCLE_MODE and $MN_M_NO_FCT_CYCLE_PAR. The programmed D number can be queried in the cycle via the system variables $C_D / $C_D_PROG. $MN_D_NO_FCT_CYCLE_NAME can only run in Siemens mode (G290). A maximum of one M/T/D function replacement can be effective for each part program line. A modal subroutine call may not be programmed in the block with the D function replacement. It is also not permissible to program a subroutine return jump or end of part program. Alarm 14016 is generated if these conventions are not observed.
Corresponding to...	
Further references:	ISO dialects for Sinumerik (FBFA)

8.1.6 Machine data for the Siemens user data

The numbers of the Siemens machine data are listed in the following. These data are defined by Siemens and must not be used by customers. No detailed description of them is given for this reason.

18200

18201

18202

18203

18204

18205

18206

18207

18208

18209

Notice

A detailed description of machine data 18091, 18093, 18095, 18097 and 18099 has been provided, but these MD may be used only if they are set to their respective defaults.

Notes

Signal description PLC interface

9

Overview of data blocks

The table below shows an overview of the data blocks used for data management. DB 71 to DB 73 are the tool management interfaces.

DB 71	For loading/unloading points
DB 72	For spindle as change position
DB 73	For turret as change position
DB 74	Internal data block of basic program for tool management

1. The interfaces for loading magazines are organized in such a way in DB 71 that a separate interface area is defined for every configured loading magazine. The interface area for loading point 1 generally has the task of loading into the spindle. It also receives commands for relocating and positioning tools in any location.
2. DB 72 includes an independent interface area for every spindle defined in the tool management system.
3. DB 73 includes an independent interface area for every turret in the magazine configuration. The turret numbers are counted contiguously from the lowest to the highest magazine number.

All interfaces are designed for receiving tool-management command (load, tool change, ...). Basic program blocks FC 7 and FC 8 are used to communicate the current positions of tools.

One of the interface is updated by NCK bia the basic program in accordance with a command (e.g. by operating the function “Load” or by a part-program function like “Tool change”).

Notice

If data for magazine, buffer or load/unload-position is changed in the installation branch, then:

---

1. Press softkey Create PLC data (HMI Avanced) or change the assignment of DB 4 in the PLC program and
2. Delete data blocks DB 71 to DB 74 and perform a cold restart of the PLC.

9.1 Interface for loading/unloading magazine

DB71 Data block	Signals of load/unload points NCK ->PLC interface							
Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Interfaces							
DBB 0	INT 8	INT 7	INT 6	INT 5	INT 4	INT 3	INT 2	INT 1
DBB 1	INT 16	INT 15	INT 14	INT 13	INT 12	INT 11	INT 10	INT 9
DBB 2, 3								
DBB n + 0				NC program positions magazine	Position	Relocate	Unload	Load
DBB n + 1	Unassigned							
DBB n + 2	Assigned channel (8bit‑Int)							
DBB n + 3	Tool management number (8bit‑Int)							
DBD n + 4	Reserved							
DBD n + 8	Reserved							
DBD n + 12	Reserved							
DBW n + 16	Identifier for load/unload point (Int), (fixed value 9999)							
DBW n + 18	Location no. of load/unload point (Int)							
DBW n + 20	Magazine no. (source) for loading/relocation/positioning (Int)							
DBW n + 22	Location no. (source) for loading/relocation/positioning (Int)							
DBW n + 24	Magazine no. target for loading/relocation/positioning (Int)							
DBW n + 26	Location no. target for loading/relocation/positioning (Int)							
DBW n + 28 HMI to PLC								Load/ unload without moving magazine
DBW n + 29	Reserved							

Initial addresses of load/unload locations:

Load/unload point 1: n = 4
2: n = 34
3: n = 64
4: n = 94

Example calculation of address DBW n+24 (magazine no. target)

n = (m–1) * len + 4   m = location no. of loading station/point
len = 30 (length of a loading point)

m =2 ; len = 30 n = (2–1) * 30 + 4 ==> n = 34

DBW (34 + 24) = DBW 58

Address for magazine no. target of 2nd load point is DBW 58.

Load point 1 is intended for loading/unloading in all spindles. This must be taken into account when load interface assignments are made (applies only to HMI Embedded; implemented automatically on HMI Advanced). Load point 1 is also used to relocate/position tools in any location (e.g. buffer location).

DB71 DBX 0.0 – 0.15	**Active status of interface 1–16**	
Edge evaluation	Signal(s) updated: **Conditional**	Signals valid from SW version: **2**
Signal state 1	The active interface has a valid data record. A task bit has been set in DBB (n+0). There are 16 interfaces. Address “n” must always be calculated for the active interface.	
Signal state 0	Operation for this interface has ended. Is reset by FC 8.	
Additional references		

DB71 DBX(n+0).0	**Command: Load**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state 1	Load operation for a tool is initiated. The magazine location into which the tool is to be loaded is defined in DBW (n+24) and DBW (n+26). The loading point in question is the location number of the load point. It also appears in DBW (n+18).	
Corresponding to...	DB71 DBW(n+16) and (n+18) or (n+24) and (n+26)	
Additional references		

DB71 DBX(n+0).1	**Command: Unload**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state 1	Unload operation for a tool is initiated. The magazine location from which the tool is to be unloaded is defined in DBW (n+20) and DBW (n+22). The number of the unload point is defined in DBW (n+18).	
Corresponding to...	DB71 DBW(n+16) and (n+18) or (n+20) and (n+22)	
Additional references		

DB71 DBX(n+0).2	Command: Relocation	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state 1	Relocate operation for a tool is initiated. From magazine/location (n+20, n+22= source) to magazine/location (n+24, n+26= target)	
Corresponding to...		
Additional references		

Notice

The bits in DBB (n+0) (load, unload,....) are not updated by the basic program until a new task exists for this interface. They are current only if the corresponding interface bit in DBB0 is set to “1”. However, the bits can be reset by the user if necessary.

DB71 DBX(n+0).3	Command: Positioning to loading point	
Edge evaluation	Signal(s) updated:	Signal(s) valid from SW: **3.2**
Signal state 1	A magazine location is to be positioned at the loading point (magazine no. 9999). The magazine location to be moved to the loading point is defined in DB71.DBW n+20 and n+22. The loading point is stored in DB71.DBW n+18.	
Corresponding to...		

DB71. DBB(n+2)	Assigned channel	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Number of channel for which active interface applies	
Corresponding to...		

DB71. DBB(n+3)	**Tool management no.**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Associated tool-management number; corresponds to the number of the TO unit within a TO area	
Corresponding to...		

DB71. DBW(n+16)	**Identifier for load/unload point (fixed value 9999)**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	The identifier for the loading/unloading point is fixed as the value 9999.	
Corresponding to...		
Additional references		

DB71. DBW(n+18)	**Location no. of the loading/unloading point**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW version **2**
Meaning	The location no. of the loading/unloading is displayed.	
Corresponding to...		
Additional references		

DB71. DBW(n+20)	**Magazine no. (source) for unloading/relocating/ positioning**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Unload: Magazine from which the tool is to be unloaded Relocate: Magazine from which the tool is taken Position: Magazine to be positioned	
Corresponding to...	DBW(n + 22)	

DB71. DBW(n+22)	Location no. (source) for unloading/relocating/ positioning	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Unload: Location from which the tool is to be unloaded Relocate: Location from which the tool is taken Position: Location that shall be positioned at the loading point DBW(n+18)	
Corresponding to...	DBW(n+20)	

DB71. DBW(n+24)	Magazine no. (target) for loading/relocating/ positioning	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Load: Magazine into which tool is to be loaded Relocate: Magazine into which the tool is to be placed Position: Magazine at which the tool must be positioned Tool remains at original location Only meaningful for interface 1. If values other than 0 are entered here, the data define the magazine or location for positioning (language command POSM).	
Corresponding to...	DBW(n + 26)	

DB71. DBW(n+26) MD number	Location no. (target) for loading/relocating/ positioning	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Load: Location into which tool is to be loaded Relocate: Location into which the tool is to be placed Position: Location at which the tool must be positioned Tool remains at original location Only meaningful for interface 1. If values other than 0 are entered here, the data define the magazine or location for positioning (language command POSM).	
Corresponding to...	DBW(n+24)	

DB71. DBX(n+28) MD number	**Load/unload without moving magazine**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **6**
Meaning	HMI / Jobshop sets/deletes this signal when requested by the operator. If the bit is active, there must be no traversing motion of the magazine, only a mechanical unlocking/locking of the location. The load/unload command must be acknowledged after the action. With positioning and relocating request, this signal is not valid for a traversing motion.	

9.2 Interface for spindle as change position

DB72 Data block	Spindle as change point Interface NCK->PLC							
Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DBB 0	INT 8	INT 7	INT 6	INT 5	INT 4	INT 3	INT 2	INT 1
DBB 1	INT 16	INT 15	INT 14	INT 13	INT 12	INT 11	INT 10	INT 9
DBB 2, 3								
DBB n + 0	Tool remains in spindle	Detach manual tool	Attach manual tool	OldTool in BL No. (n+42)	T0	Prepare change	Change tool (initiated by: M06)	Obligatory change
DBB n + 1	Unassigned							
DBB n + 2	Assigned channel (8bit-Int)							
DBB n + 3	Tool management number (8bit-Int)							
DBD n + 4	$P_VDITCP[0] User parameter 0 (DWord)							
DBD n + 8	$P_VDITCP[1] User parameter 1 (DWord)							
DBD n +12	$P_VDITCP[2] User parameter 2 (DWord)							
DBW n + 16	Buffer identifier (Int), fixed value 9998) equals “Target position for new tool”							
DBW n +18	Relative location (target) in buffer magazine (Int)							
DBW n +20	Magazine no. (source) for new tool (Int)							
DBW n +.22	Location no. (source) for new tool (Int)							
DBW n + 24	Magazine no. (target) for old tool (Int)							
DBW n + 26	Location no. (target) for old tool (Int)							
DBW n + 28	Tool new: Location type (Int)							
DBW n + 30	Tool new: size left (Int)							
DBW n + 32	Tool new: size right (Int)							
DBW n + 34	Tool new: size top (Int)							
DBW n + 36	Tool new: size bottom (Int)							
DBW n + 38	Tool status for new tool							
	Tool was in use	Tool with fixed loc. code		Prewarn. limit reached	Tool calibration		Tool enabled	Active tool
DBW n + 40	Tool new: Internal T no. of NCK (Int)							
DBW n + 42	If DBX (n+0.4) = 1, then the buffer location of the old tool must be entered here							
DBW n + 44	Source magazine of new tool (from SW version 6.4)							
DBW n + 46	Source location of new tool (from SW version 6.4)							

Initial addresses of spindle:	Spindle 1: n= 4 Spindle 2: n= 52 Spindle 3: n= 100
n = (m–1)* len + 4	m = location no. of change position len = 48

Notice

If only M06 is programmed, only free parameters (from SW version 6), channel, tool management number and the bit for "Perform change" are updated.

DB72. DBX 0.0 – 0.15	**Active status of interface 1–16**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	Associated interface has a valid block, a tool change request of tool preparation has been initiated.	
Signal state **0**	Operation for this interface has ended. Is reset by FC 8.	
Corresponding to...		

DB72. DBX(n+0).0	**Command code: Obligatory change**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	The new tool is fixed-location-coded	
Signal state **0**		
Corresponding to...	Position of participating tools	
Additional references		

DB72. DBX(n+0).1	**Command code: Perform change with M06**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	M06 command was programmed for tool change, the tool change can now take place.	
Signal state **0**		

DB72. DBX(n+0).2	**Command code: Prepare change**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	Prepare new tool for change. If necessary, move location for old tool to spindle.	
Signal state **0**		
Corresponding to...		
Additional references		

DB72. DBX(n+0).3	**Command code: T0**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	Indicates that T0 has been programmed (no-load traversing of spindle)	
Signal state **0**		
Corresponding to...		
Additional references		

DB72. DBX(n+0).4	**Command code: Old tool in buffer**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	The buffer number of the tool to be changed is written in DB72.DBW (n+42)	
Signal state **0**		
Corresponding to...		
Additional references		

DB72. DBX(n+0).5	Command code: Attach manual tool	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	A manual tool is to be loaded. Which tool is to be loaded is displayed on the HMI.	
Signal state **0**		
Corresponding to...		
Additional references		

DB72. DBX(n+0).6	Command code: Detach manual tool	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	The tool is to be changed via manual operation.	
Signal state **0**		
Corresponding to...		
Additional references		

DB72. DBX(n+0).7	Command code: Tool remains in spindle	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	The bit is set at change from spindle to spindle. Initiated e.g. by reset and start mode or block search.	
Signal state **0**		
Corresponding to...		
Additional references		

Notice

The bit in DBB (n+0).2 (prepare change) is **not** reset by the system with a change command. The bits in DBB(n+0)... are current only if the corresponding interface bit in DBB0 is set to “1”. However, the bits can be reset by the user if necessary.
If DBX(n+0).1 and DBX(n+0).2 are present at the same time, it means that T and M06 were programmed in one block.

DB72. DBB(n+2)	Assigned channel	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Number of channel for which active interface applies	
Corresponding to...		
Additional references		

DB72. DBB(n+3)	Tool management no.	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Associated tool management number (TO area)	
Corresponding to...		
Additional references		

DB72. DBD(n+4)	User parameter 0 (DInt)	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	If you need to send a value to the PLC via the part program, the transfer can be programmed with $P_VDITCP[0]=(value). Values are transferred at T call. From **SW version 6** the parameters are also transferred with M06.	
Corresponding to...		
Additional references		

DB72. DBD(n+8)	User parameter 1 (DInt)	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	If you need to send a value to the PLC via the part program, the transfer can be programmed with $P_VDITCP[1]=(value);.	
Corresponding to...		
Additional references		

DB72. DBD(n+12)	User parameter 2 (DInt)	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	If you need to send a value to the PLC via the part program, the transfer can be programmed with $P_VDITCP[2]=(value);.	
Corresponding to...		
Additional references		

DB72. DBW(n+16)	Buffer magazine no. (fixed value 9998) target position for new tool	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Magazine no. 9998 (buffer magazine); Target magazine for new tool	
Corresponding to...		
Additional references		

DB72. DBW(n+18)	Location in buffer magazine (spindle)	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Location number of buffer magazine to which the new tool is to be loaded. This is normally the spindle. The location number defined for this particular buffer during start-up is output.	
Corresponding to...		
Additional references		

DB72. DBW(n+20)	Magazine no. (source) for new too to be loaded at change	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	No. of magazine the new spindle tool came from (source)	
Corresponding to...	DBW(n+22)	
Additional references		

DB72. DBW(n+22)	**Location no. (source) for new tool**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	No. of location the new spindle tool came from (source)	
Corresponding to...	DBW(n+20)	
Additional references		

DB72. DBW(n+24)	**Magazine no. (target) for old tool to be removed at change**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Number of magazine in which the tool to be removed at change will be placed.	
Corresponding to...	DBW(n+26)	
Additional references		

DB72. DBW(n+26)	**Location no. (target) for old tool**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Magazine location for tool that is unloaded at change.	
Corresponding to...		
Additional references		

DB72. DBW(n+28)	**Tool new: Location type**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	The location type of the new spindle tool is entered here.	
Corresponding to...	Tool size: Left, right, top, bottom	
Additional references		

DB72. DBW(n+30)	Tool new: Size left (Int)	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Specification of tool size **on left** in half locations for the new spindle tool.	
Corresponding to...		
Additional references		

DB72. DBW(n+32)	Tool new: Size right (Int)	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Specification of tool size **on right** in half locations for the new spindle tool.	
Corresponding to...		
Additional references		

DB72. DBW(n+34)	Tool new: Size top	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Specification of tool size at **top** in half locations for the new spindle tool.	
Corresponding to...		
Additional references		

DB72. DBW(n+36)	Tool new: Size bottom	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Specification of tool size at **bottom** in half locations for the new spindle tool.	
Corresponding to...		
Additional references		

DB72. DBW(n+38)	**Tool status for new tool**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	**Bit 0**: Active tool **Bit 1**: Tool enabled **Bit 2**: Tool disabled **Bit 3**: Measured tool **Bit 4**: Prewarning limit reached **Bit 6**: Tool is fixed-location-coded **Bit 7**: Tool was in use	
Corresponding to...		
Additional references		

DB72. DBW(n+40)	**Tool new: Internal T no. of NCK**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Display of internal T no. of NCK for the new spindle tool.	
Corresponding to...		
Additional references		

DB72. DBW(n+42)	**Buffer location of old tool**	
Edge evaluation	Signal(s) updated:	Signal(s) valid from SW:
Meaning	If DB72.(n+0.4) = 1, the buffer location of the old tool must be entered here. This can be any buffer (also a gripper).	
Corresponding to...		
Additional references		

DB72. DBW(n+44)	**Original magazine of new tool**	
Edge evaluation	Signal(s) updated:	Signal(s) valid from SW: **6.4**
Additional references		

DB72. DBW(n+46)	Original location of new tool	
Edge evaluation	Signal(s) updated:	Signal(s) valid from SW: **6.4**
Additional references		

9.3 Interface for tool turrets as change position

DB73 Data block	Turret as change position Interface NCK->PLC							
Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DBB 0	INT 8	INT 7	INT 6	INT 5	INT 4	INT 3	INT 2	INT 1
DBB 1	INT 16	INT 15	INT 14	INT 13	INT 12	INT 11	INT 10	INT 9
DBB 2, 3								
DBB n + 0					T0		Perform change (initiation: T NO.)	Obligatory change
DBB n + 1	Unassigned							
DBB n + 2	Assigned channel (8bit-Int)							
DBB n + 3	Tool management number (8bit-Int)							
DBD n + 4	$P_VDITCP[0] User parameter 0 (DWord)							
DBD n + 8	$P_VDITCP[1] User parameter 1 (DWord)							
DBD n + 12	$P_VDITCP[2] User parameter 2 (DWord)							
DBW n + 16	Reserved							
DBW n + 18	Reserved							
DBW n + 20	Magazine no. of turret (Int)							
DBW n + 22	Location no. of new tool (Int)							
DBW n + 24	Reserved							
DBW n + 26	Location no. of old tool (Int)							
DBW n + 28	Tool new: loc. type (Int)							
DBW n + 30	Tool new: size left (Int)							
DBW n + 32	Tool new: size right (Int)							
DBW n + 34	Tool new: size top (Int)							
DBW n + 36	Tool new: size bottom (Int)							

DB73 Data block	Turret as change position Interface NCK–>PLC							
DBW n + 38	Tool status for tool							
	Tool was in use	Tool with fixed loc. code		Prewarn. limit reached	Tool calibration		Tool enabled	Active tool
DBW n + 40	Tool new: Internal T no. of NCK (Int)							
DBW n + 42	Source location of new tool in this circular magazine (from SW version 6.4)							

Initial addresses of turrets: Turret 1: n= 4
Turret 2: n= 48
Turret 3: n= 92

n = (m–1)* len + 4
m = Location no. of the point of change
len = 44

Example for change position 3: n = (3–1)*n 44+ 4= 2*44 + 4= 88 + 4= 92

DB73 – DBX 0.0 – 0.15	**Active status of interface 1–16**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	Associated interface has a valid data block	
Signal state **0**	Operation for this interface has ended. Is reset by FC 7.	
Additional references		

DB73. DBX(n+0).0	**Command code: Obligatory change**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**		
Signal state **0**		
Corresponding to...	Position of involved tools	
Additional references		

DB73. DBX(n+0).1	Command code: Perform change	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Signal state **1**	Execute tool change	
Signal state **0**		
Additional references		

DB73. DBB(n+0).3	T0	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Indicates that T0 was programmed.	
Additional references		

DB73. DBB(n+2)	Assigned channel	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Number of channel from which the T word was programmed.	
Additional references		

DB73. DBB(n+3)	Tool management no.	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Associated tool management number (TO area) of channel	
Corresponding to...		
Additional references		

Notice

The bits in DBB (n+0) (obligatory change, execute change,...) are **not** reset by the system. They are current only if the corresponding interface bit in DBB0 is set to “1”. However, the bits can be reset by the user if necessary.

DB73. DBD(n+4)	User parameter 0 (DInt)	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	If you need to send a value to the PLC via the part program, this can be achieved by programming $P_VDITCP[0]=(value). Parameters 0–2 are passed with the T command.	
Corresponding to...		
Additional references		

DB73. DBD(n+8)	User parameter 1 (DInt)	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	If you need to send a value to the PLC via the part program, the transfer can be programmed with $P_VDITCP[1]=(value);.	
Corresponding to...		
Additional references		

DB73. DBD(n+12)	User parameter 2 (DInt)	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	If you need to send a value to the PLC via the part program, the transfer can be programmed with $P_VDITCP[2]=(value);.	
Corresponding to...		
Additional references		

DB73. DBW(n+16)	Reserved	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning		

DB73. DBW(n+18)	Reserved	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning		

DB73. DBW(n+20)	**Magazine no. of new tool**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Magazine number of the new tool to be used for machining.	
Corresponding to...	DBW(n+22)	
Additional references		

DB73. DBW(n+22)	**Location no. of new tool to be loaded at change**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Location number of the new tool to be used for machining.	
Corresponding to...	DBW(n+20)	
Additional references		

DB73. DBW(n+24)	**Reserved**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning		
Corresponding to...		

DB73. DBW(n+26)	**Location no. of old tool to be unloaded**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Location number of the old tool (used up to now for machining)	
Corresponding to...		
Additional references		

DB73. DBW(n+28)	**Tool new: Location type**
Meaning	The location type of the new tool is entered here.
Corresponding to...	Tool size: Left, right, top, bottom
Additional references	

DB73. DBW(n+30)	**Tool new: Size left (Int)**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Specification of new tool size **on left** in half locations.	
Corresponding to...		
Additional references		

DB73. DBW(n+32)	**Tool new: Size right (Int)**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Specification of new tool size **on right** in half locations.	
Corresponding to...		
Additional references		

DB73. DBW(n+34)	**Tool new: Size top**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Specification of new tool size **at top** in half locations.	
Corresponding to...		
Additional references		

DB73. DBW(n+36)	**Tool new: Size bottom**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Specification of new tool size **at bottom** in half locations.	
Corresponding to...		
Additional references		

DB73. DBW(n+38)	**Tool status for new tool**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	**bit 0**: Active tool **bit 1**: Tool enabled **bit 2**: **bit 3**: Measure tool **bit 4**: Prewarn lim. reached **bit 6**: Tool is fixed-location-coded **bit 7**: Tool was in use	
Corresponding to...		
Additional references		

DB73. DBW(n+40)	**Tool new: Internal T no. of NCK**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Meaning	Display of internal T no. of NCK for the new tool. Tool management variables can be read/written via FB2/FB 3 using this T no.	
Corresponding to...		
Additional references		

DB73. DBW(n+42)	**Source location of new tool in this circular magazine (from SW version 6.4)**	
Edge evaluation	Signal(s) updated: **Conditional**	Signal(s) valid from SW: **2**
Additional references		

9.4 Interface NC channels

Signals are also contained in the channel data blocks for tool management functions.
The data relevant for tool management is in bold formatting.

DB21–30 Data block	Signals to/from NC CHANNEL PLC–>NCK interface							
Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DBB 1	Activate program test	PLC action ended	CLC override	CLC Stop	**Time monitoring active (tool management)**	Synchronized action OFF	Enable protection zones	Activate referencing
DBB 29	**Tool block not effective**	**Disable wear monitoring**	**De-activate workpiece counter**	Activate PTP traversal	Activate fixed feed 4	Activate fixed feed 3	Activate fixed feed 2	Activate fixed feed 1
Cyclic signals from NC channel								
DBB 317	**Tool missing**	PTP traversal active						External language mode active
Change signals tool management functions								
DBB 344					**Last replacement tool from tool group**	**Transition to new replacement tool**	**Tool limit value reached**	**Tool prewarning limit reached**
Transferred tool management functions								
DBD 348	**T number for tool management prewarning limit (DInt)**							
DBD 352	**T number for tool limit value (DInt)**							
DBD 356	**T number of new replacement tool (DInt)**							
DBD 360	**T number of last replacement tool (DInt)**							

DB21. DBX 1.3	The user can start and stop tool life monitoring time using PLC signal “Time monitor active”. The effectiveness of this control is set via MD 20310 bit 17.

DB21. DBX 29.5	Switches workpiece count monitoring ON/OFF

DB21. DBX 29.6	Switches wear monitoring ON/OFF

DB21. DBX 29.7	**VDI signal “Tool disable ineffective”** (bit value=1) means the NCK does not process the tool status “Disabled” during tool search. **VDI signal “Tool disable effective”** (bit value=0) means the NCK processes the tool status “Disabled” during tool search.

DB21. DBX 317.7	Display in PLC that the programmed tool is missing.

DB21.DBX 344.0–344.3	**Modification signals of the tool management functions**	
Edge evaluation	Signal(s) updated: Jobcontrolled by NCK	Signal(s) valid from SW: **5.1**
Meaning	A T number for tool prewarning limit, limit value, new replacement tool, last replacement tool has been output with a value at the interface at the beginning of an OB1 cycle together with the associated modification signal. In this case, the change signal indicates that the appropriate value is valid.	
Corresponding to...		
Additional references		

9.5 Interface magazine configuration

DB4	HMI –> PLC interface	
Address	**Meaning**	**Data type**
DBB 64	Number of magazines including buffer magazines and loading magazines	BYTE
<	Beginning of rerun loop; number of reruns from DB4.DBB64	
DBW 65 (70, 75, ...)	Magazine number	INT
DBB 67 (...)	Magazine type	BYTE
DBW 68 (...)	Number of locations	INT
>	Rerun lool end	
Address = (contents DBB64*5) + 65	Number of spindles	BYTE

Notes

Alarms

10

Alarm no.	Brief Description
6402	Tool change not possible, magazine number does not exist
6403	Tool change not possible, specified magazine location does not exist
6404	Tool change not possible because tool not available or cannot be used
6405	Command has invalid PLC acknowledgement parameter
6406	PLC acknowledgement missing
6407	Tool is to be set down at a location that does not meet the requirements for loading
6410	One cutting edge of the monitored tool has reached a warning limit
6411	One cutting edge of the monitored tool has reached a warning limit
6412	One cutting edge of the monitored tool has reached a monitoring limit
6413	One cutting edge of the monitored tool has reached a monitoring limit
6421	No location available for the tool in the tool-holding magazine
6422	No tool-motion command possible because magazine not defined
6423	No tool-motion command possible because no location in the magazine
6424	No tool-motion command possible because tool not available or cannot be used
6425	No tool-motion command possible because tool cannot be be put down at the specified location in the magazine
6430	Workpiece counter: overflow in table of monitored cutting edges
6431	Function not permitted because TOOLMAN / TOOLMAN monitoring not activated
6432	Function cannot be executed because no tool assigned to spindle
6433	System variable not available for active tool management.
6441	Not permitted to write $P_USEKT.
6450	Tool change not possible because magazine-location number is not valid.
6451	No buffer magazine defined.
6452	Toolholder number / spindle number not defined.
6453	No relationship defined between toolholder number / spindle number and buffer magazine.
6454	Neither spindle nor buffer location has a distance relationship.
6924	Neither spindle nor buffer location has a distance relationship.

Alarm no.	Brief Description
17001	No more memory for tool magazine data
17160	No tool selected
17180	Illegal D number
17181	D number not known
17182	Illegal additive offset number
17188	The D number given in the channel's TO units is not unique
17189	D number is not unique
17191	Unknown tool identifier
17192	No further replacement tools possible
17194	No suitable tool found
17202	Cannot delete magazine data
17212	Manual tool must be changed
17214	Remove manual tool from spindle/toolholder
17216	Manual tools must be changed
17220	Tool does not exist
17224	It is not possible on this system to select tool offsets for tools of the specified tool type.
17230	Duplo no. already assigned
17240	Invalid tool definition
17250	Invalid magazine definition
17260	Invalid magazine location definition
17262	Incorrect tool-adapter assignment
20150	PLC terminates the interrupted command
20160	PLC can terminate only incorrectly aborted commands
22066	Tool motion not possible because specified tool is not in magazine
22067	Tool change not changed because no tool ready for use in the tool group
22068	No tool ready for use in the tool group
22069	No tool ready for use in the tool group
22070	Change tool into magazine. Repeat data backup
22071	Tool has the status "active" in an "inactive" wear group
400601	Configuration of loading points faulty.
400602	Configuration of spindles faulty.
400603	Configuration of turrets faulty.

Alarm no.	Brief Description
400604	Set change with M06 in machine data
410151	Magazine data for tool management missing in PLC

Machine data MD 11410 SUPPRESS_ALARM_MASK can be set bit-wise to suppress specific alarms.

Bit	Alarm number
2	16924
4	17189
5	22071
7	22070
8	6411, 6413
9	6410, 6412

10.1 Alarm descriptions

Alarm no.	
6402	Channel %1 tool change not possible because magazine no. %2 not available
Explanation	%1 = channel ID, %2 = magazine number The desired tool change is not possible. The magazine with the specified number is not available.
Reaction	Alarm display. Interface signals are set NC Start disable NC stop for alarm
Remedy	– Check whether the magazine data is correctly defined. – Check whether the magazine is connected to the required spindle via a distance relationship
Program continuation	Cancel the alarm with the RESET button and start the part program again.

Alarm no.	
6403	Channel %1 tool change not possible because magazine no. %2 on magazine %3 not available.
Explanation	%1 = channel ID %2 = magazine number, %3 = magazine location number The desired tool change is not possible. The specified magazine location is not contained in the specified magazine.
Reaction	Alarm display Interface signals are set NC Start disable NC stop for alarm
Remedy	– Check whether the magazine data is correctly defined.
Program continuation	Cancel the alarm with the RESET button and start the part program again.

Alarm no.	
6404	Channel %1 tool change not possible. Tool %2 not available or cannot be used
Explanation	%1 = channel ID, %2 = string (identifier) The desired tool change is not possible. The specified tool does not exist or cannot be used.
Reaction	Alarm display Interface signals are set NC Start disable NC stop for alarm
Remedy	– Check whether the part program is written correctly. – Check whether the magazine data is correctly defined. – Check whether there is a replacement tool which can be used for the specified tool.
Program continuation	Cancel the alarm with the RESET button and start the part program again.

Alarm no.	
6405	Channel %1 command %2 has an invalid PLC acknowledgement parameter %3 identification %4
Explanation	%1 Channel ID, %2 = command no. %3 = PLC acknowledgement parameter, %4 = error identification The specified command has been answered by the PLC with an invalid acknowledgement in the current combination. The following assignments are defined for "command no.": 1 Move tool, load or unload magazine 2 Prepare tool change 3 Execute tool change 4 Prepare tool change and execute with T command 5 Prepare tool change and execute with M command 7 Terminate aborted tool command 8 Check tool motion with reservation 9 Check tool motion 0 Transportation acknowledgment The tool change defined by the command cannot be executed. The magazine location specified in the invalid parameter does not exist in the magazine. The error code (%4) explains the alarm in more detail: 0 Not defined 1 Status not allowed now or undefined status received from PLC 2 Source and/or destination magazine no. / location no. not known 3 Not defined 4 For tool motion command, the destination magazine no. and/or destination location no. are not the final destination 5 Not defined 6 For tool change, source and/or destination magazine no. / location no. not known 7 PLC command with inconsistent data: either magazine addresses inconsistent in VDI or NCK command not the same as the PLC acknowledgement, or both 8 Not defined 9 Not defined 10 It is not defined to reserve buffer storage for an asynchronous tool motion.
Reaction	Alarm display Interface signals are set NC Start disable NC stop for alarm
Remedy	Notify authorized personnel / service Faulty PLC communication: correct the PLC program.
Program continuation	Cancel the alarm with the RESET button and start the part program again.

Alarm no.	
6406	Channel %1 PLC acknowledge for command %2 is missing
Explanation	%1 = channel ID, %2 = command no. There is still no acknowledgement from the PLC for the tool change. The NCK cannot continue processing until it receives this acknowledgement for the specified command number. Possible values are described under alarm 6405
Reaction	Alarm display Interface signals are set NC Start disable
Remedy	Notify authorized personnel / service – Faulty PLC communication: correct the PLC program. – It is possible to release NCK from the wait condition with the PLC command 7. This aborts the waiting command.
Program continuation	Cancel the alarm with the RESET button and start the part program again.

Alarm no.	
6407	Channel %1 tool %2 cannot be placed in magazine %3 on location %4. Invalid magazine definition!
Explanation	%1 = channel ID, %2 = string (identifier), %3 = magazine number, %4 = magazine location number A tool change request or a verification request was issued to put the tool in a location which does not satisfy the prerequisites for filling. The following causes for the error are possible: – Location is blocked or not free – Tool type does not match the location type – Tool possibly too large, adjacent locations are not free
Reaction	Alarm display Interface signals are set NC Start disable NC stop for alarm
Remedy	– Check whether the magazine data is correctly defined (especially the location type) – Check whether the tool data is correctly defined (especially the location type)
Program continuation	Cancel the alarm with the RESET button and start the part program again.

Alarm no.	
6410	TO unit %1 tool %2 with duplo no. %3 has reached a tool warning limit
Explanation	%1 = TO unit, %2 = tool identifier (name), %3 = duplo number Indication that at least one cutting edge of the timer or quantity-monitored tool has reached its warning limit. The alarm is triggered via the OPI interface (HMI, PLC). The channel context is not defined. The TO unit is therefore specified.
Reaction	Alarm display Interface signals are set
Remedy	For information only. The user must decide what to do.
Program continuation	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6411	Channel %1 tool %2 with duplo no % 3 has reached tool warning limit
Explanation	%1 = channel number %2 = tool identifier (name), %3 = duplo number Indication that at least one cutting edge of the timer or workpiece quantity-monitored tool has reached its warning limit. Limit is detected in the context of the channel. The alarm originates during NC program execution. The channel context is defined.
Reaction	Alarm display Interface signals are set
Remedy	For information only. The user must decide what to do.
Program continuation	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6412	TO unit %1 tool %2 with duplo no.%3 has reached tool monitoring limit
Explanation	%1 = TO unit, %2 = tool identifier (name), %3 = duplo number Indication that at least one cutting edge of the timer or quantity-monitored tool has reached its monitoring limit. The alarm is triggered via the OPI interface (HMI, PLC). The channel context is not defined, therefore the TO unit is specified.
Reaction	Alarm display Interface signals are set
Remedy	For information only. The user must decide what to do.
Program continuation	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6413	Channel %1 tool %2 with duplo no % 3 has reached tool monitoring limit
Explanation	%1 = channel number, %2 = tool identifier (Name), %3 = duplo number Indication that at least one cutting edge of the timer or quantity-monitored tool has reached its monitoring limit. Limit is detected in the context of the channel. The alarm originates during NC program execution. The channel context is defined.
Reaction	Alarm display Interface signals are set
Remedy	For information only. The user must decide what to do.
Program continuation	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6421	Channel %1 tool motion not possible. There is no empty location for tool %2 duplo no. %3 in magazine %4.
Explanation	%1 = channel ID, %2 = string (identifier), %3 = duplo number, %4 = magazine number The desired tool motion command – triggered from the HMI or PLC – is not possible. The tool cannot be moved into the specified tool magazine. There is no location available for this tool.
Reaction	Alarm display Interface signals are set NC Start disable
Remedy	– Check whether the magazine data have been defined correctly (e.g. the magazine must not be disabled). – Check whether the tool data are correctly defined (for example, the tool location type must match the location types allowed in the magazine). – Check whether there is still room in the magazine to add another tool; there may not be due to operating procedures. – Check whether a location type hierarchy is defined and whether it, for example, does not allow insertion of a type ’A’ tool in a free location with type ’B’.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6422	Channel %1 tool motion not possible. Magazine no. %2 not available!
Explanation	%1 = channel ID, %2 = magazine number The desired tool motion command – triggered from the HMI or PLC – is not possible. The magazine with the specified number is not available.
Reaction	Alarm display Interface signals are set NC Start disable
Remedy	– Check whether the magazine data is correctly defined. – If the PLC issued the command for motion: check whether the PLC program is correct. – If the HMI issued the command for motion: check whether the HMI command was assigned correct parameters.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6423	Channel %1 tool motion not possible. Magazine location no. %2 in magazine %3 not available.
Explanation	%1 = channel ID, %2 = magazine location number, %3 = magazine number The desired tool motion command – triggered from the HMI or PLC – is not possible. The specified magazine location is not contained in the specified magazine.
Reaction	Alarm display Interface signals are set NC Start disable
Remedy	– Check whether the magazine data is correctly defined.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6424	Channel %1 tool motion not possible. Tool %2 not available or cannot be used
Explanation	%1 = channel ID, %2 = string (identifier) The desired tool motion command – triggered from the HMI or PLC – is not possible. The specified tool is not defined.
Reaction	Alarm display Interface signals are set NC Start disable
Remedy	– Check whether the magazine data is correctly defined. – Check whether the move command has been correctly parameterized.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6425	Channel %1 tool %2 cannot be placed in magazine %3 on location %4. Invalid magazine definition
Explanation	%1 = channel ID, %2 = string (identifier), %3 = magazine number, %4 = magazine location number The desired tool motion command – triggered from the HMI or PLC – is not possible. A movement request was issued to put the tool in a location which does not satisfy the prerequisites for filling. The following causes for the error are possible: – Location is disabled or not free – Tool type does not match the location type. – Tool possibly too large, adjacent locations are not free.
Reaction	Alarm display Interface signals are set NC Start disable
Remedy	– Check whether the magazine data is correctly defined. – Check whether there is still room in the magazine to add another tool; there may not be due to operating procedures. – Check whether a location type hierarchy is defined and whether it, for example, does not allow insertion of a type ’A’ tool in a free location with type ’B’.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6430	Workpiece counter: overflow in table of monitored cutting edges
Explanation	No more cutting edges can be entered in the workpiece counter table. The total number of cutting edges that can be noted for the workpiece counter is the same as the total number of possible cutting edges in the NCK. In other words, the limit is reached if each tool uses each cutting edge precisely once for a workpiece. If several workpieces are made on several spindles simultaneously, it is possible to note cutting 18100 MM_NUM_CUTTING_EDGES_IN_TOA for the totaling counter for all of the workpieces. If the alarm is pending then this means that the cutting edges that are about to be used will no longer be monitored by the workpiece counter for such a time until the table has been cleared again, e.g. by the NC command SETPIECE or the appropriate order from the HMI, programmable controller (PI service).
Reaction	Alarm display Interface signals are set NC Start disable
Remedy	– Decrement workpiece counter overlooked? Then program SETPIECE in the part program, or add the correct command in the PLC program. – If the part program or the programmable controller is correct, then more memory should be set for the tool cutting edge by the machine data $MM_NUM_CUTTING_EDGES_IN_TOA (only possible for those with access authorized).
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6431	Function not allowed. Tool management/tool-management monitoring not activated
Explanation	Occurs when a data management function is called which is not available because tool management is deactivated. For example, the language commands GETT, SETPIECE, GETSELT, NEWT, DELT.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Please inform the authorized personnel/service department. – Verify how the NC control shall be configured. Is tool management or ToolMan monitoring necessary but not yet activated? – Is a part program used that has been designed for NC control with tool management/ToolMan monitoring? Either operate the part program with the matching NC controls or modify the part program. – Activate tool management/ToolMan monitoring by setting the appropriate machine data. See $MM_TOOL_MANAGEMENT_MASK, $MC_TOOL_MANAGEMENT_MASK. – Check whether the required option is set accordingly.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6432	Function cannot be executed. No tool assigned to spindle
Explanation	An attempt was made to perform an operation that requires a tool to be located on the spindle. This can be the workpiece count monitoring function, for example.
Reaction	Alarm display Interface signals are set
Remedy	– Select another function, another spindle, position tool on spindle.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6433	Channel %1 block %2 variable %3 not available with tool management
Explanation	%1 = channel number, %2 = block number, Label, %3 = source symbol The system variable specified in %3 is not available with active tool manage-ment. The function GETSELT should be used with $P_TOOLP.
Reaction	Alarm display Interface signals are set NC Start disable
Remedy	ProgramChanging
Continue program	Clear the alarm with the delete key.

Alarm no.	
6441	Writing of $P_USEKT not allowed.
Explanation	An attempt was made to write into the value of $P_USEKT. This is not possible because the programming T="location number" with automatic setting of $P_USEKT is active.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Verify how the NC control should be configured (bit 16 and bit 22 in TOOLS_MANAGEMENT_MASK) – Either operate the part program with the matching NC control or modify the part program.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
6450	Channel %1 tool change not possible. Invalid magazine location number %2 in the buffer magazine
Explanation	%1 = channel number, %2 magazine location number The desired tool change is not possible. The specified magazine location is a toolholder/spindle or is empty. Only buffer-storage numbers that are not for toolholders/spindles may be programmed using the NC command TCI; i.e. the location number of a gripper for example, is not allowed.
Reaction	Alarm display Interface signals are set
Remedy	Check whether the magazine data ($TC_MPP1) is correctly defined. Check that the parameters of the program command responsible are correctly assigned.
Continue program	

Alarm no.	
6451	Channel %1 tool change not possible. No buffer magazine defined.
Explanation	%1 channel number The desired tool change is not possible. No buffer magazine defined.
Reaction	Alarm display Interface signals are set
Remedy	Check whether the magazine data is correctly defined.
Continue program	

Alarm no.	
6452	Channel %1 tool change not possible. The toolholder no. / spindle no. = %2 has not been defined.
Explanation	%1 = channel number, %2 toolholder / spindle no. The desired tool change is not possible. The toolholder number / spindle number is not defined.
Reaction	Alarm display Interface signals are set
Remedy	Check whether the toolholder no./spindle no. and magazine data are correctly defined. (See $TC_MPP1, $TC_MPP5)
Continue program	

Alarm no.	
6453	Channel %1 tool change not possible. No assignment between toolholder/ spindle no. = %2 and buffer location %3
Explanation	%1 = channel number, %2 toolholder / spindle no. %3 buffer location The desired tool change is not possible. No relation has been defined between the toolholder/spindle number and the buffer location (locNo)
Reaction	Alarm display Interface signals are set
Remedy	Check whether the magazine data ($TC_MLSR) is correctly defined. Check whether the program command causing the error (e.g. TCI) has been programmed correctly.
Continue program	

Alarm no.	
6454	Channel %1 tool change not possible. There is no distance relationship available.
Explanation	%1 channel number The desired tool change is not possible. Neither spindle nor buffer location have a distance relationship.
Reaction	Alarm display Interface signals are set
Remedy	Check whether the magazine data ($TC_MDP) is correctly defined. Check whether the program command causing the error (e.g. TCI) has been programmed correctly.
Continue program	

Alarm no.	
16924	Channel %1 Caution: program test alters tool management data
Explanation	%1 = channel number Tool data is altered during program testing. You cannot automatically correct the tool data again on termination of program test mode. This alarm prompts you to create a backup of the tool data which must be copied back in when you have finished testing the program.
Reaction	Alarm display
Remedy	Please inform the authorized personnel/service department. – Save tool data on HMI and reimport data after “ProgtestOff”.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
17001	Channel %1 block %2 no more memory for tool / magazine data
Explanation	%1 = channel number, %2= block number, label The available memory for defining adapter data has been used up. If the alarm occurs when you are writing one of the $TC_ADPT parameters, you have tried to define more adapter data records than permitted by the setting in MD MM_NUM_TOOL_ADAPTER. Number of additive/setup offsets: $MN_MM_NUM_SUMCORR if $MN_MM_NUM_SUMCORR = –1 set, then the following applies number of additive offsets = $MN_MM_NUM_CUTTING_EDGES_IN_TOA * $MN_MAX_SUM-CORR_PER_CUTTING_EDGE
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	Please inform the authorized personnel/service department. – Maschinendaten ändern – Modify NC program, i.e. reduce the number of variables with discrepancies
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
17160	Channel %1 block %2 no tool selected
Explanation	%1 = channel number, %2= block number, label An attempt has been made to access the current tool offset data via the system variables: $P_AD[n] Contents of the parameter (n: 1 – 25) $P_TOOL Active D number (cutting-edge number) $P_TOOLL[n] Active tool length (n: 1 – 3) $P_TOOLR Active tool radius although no tool was previously selected.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable

Alarm no.	
Remedy	Program or activate a tool offset in the NC program before using the system variables. Example: N100 G.. ... T5 D1 ...LF The channel-specific machine data: 22550: **TOOL_CHANGE_MODE** New tool offset for M function 22560: **TOOL_CHANGE_M_MODE** M function with tool change are set to define whether activating a tool offset in the block is carried out with the T word or whether the new offset values is only computed with the M word for the tool change.
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
17180	Channel %1 block %2 illegal D number
Explanation	%1 = channel number %2 = block number, label In the displayed block, access is made to a D number (tool edge number) that is not initialized and therefore is not available.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	Check tool call in the NC part program: – Correct tool edge number programmed? If no tool edge number is specified, then D1 is automatically active. – Tool parameters P1 – P25 defined? The dimensions of the tool edge must have been entered previously either through the operator panel or through the V.24 interface.
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
17181	Channel %1 block %2 T no.= %3, D no.= %4 does not exist
Explanation	%1 = channel number, %2 = block number, label, %3 = T number, %4 = D number A D number has been programmed that the NCK does not recognize. As standard, the D number refers to the given T number. If the “flat D number” function is activated, T = 1 is output.

Alarm no.	
Reaction	Alarm display Interface signals are set Correction block with reorganization
Remedy	In case of a programming error, eliminate the error with a correction block and continue the program run. If the data block is missing, then load the NCK with a data block for the specified T/D values (via HMI, with overstore) and continue program.
Continue program	Cancel the alarm with NC START and continue processing.

Alarm no.	
17182	Channel %1 block %2 illegal sum correction number
Explanation	%1 = channel number, %2= block number, label An attempt was made to access a non-defined total offset of the current tool edge.
Reaction	Alarm display Interface signals are set Correction block with reorganization
Remedy	Access the additive offset memory with $TC_SCUP*, $TC_CEP*, check the additive offset selection Dlx or tool selection Ti or offset selection Dz.
Continue program	Cancel the alarm with NC START and continue processing.

Alarm no.	
17188	Channel %1 D number %2 defined in tool T no. %3 and %4
Explanation	%1 = channel number, %2 = compensation number D, %3 = T number for first tool, %4 = T number for second tool The specified D number %2 in the TO unit of channel %1 is not unique. The specified T numbers %3 and %4 each have an offset with number %2. If tool management is active: The specified T numbers belong to tool groups with different identifiers.
Reaction	Alarm display Interface signals are set
Remedy	– Ensure uniqueness of the D numbering within the TO units – Do not use the causal instruction if uniqueness is not needed in the following. See also command DZERO.
Continue program	The alarm is for information purposes. You can suppress the alarm output by setting bit 4 in MD $MN_SUPPRESS_ALARM_MASK.

Alarm no.	
17189	Channel %1 D number %2 of tools defined on magazine/location %3 and %4
Explanation	%1 = channel ID, %2 = D number, %3 = magazine no./ magazine location no. – “/” as separator, %4 = magazine no./ magazine location no. – “/” as separator Only possible with active tool management The specified D number %2 in the TO unit of channel %1 is not unique. The tools in the specified magazine locations %3 and %4 each have an offset with the number %2. In addition, if tool management is active: the specified T numbers belong to tool groups with different identifiers.
Reaction	Alarm display Set interface signals
Remedy	– Ensure uniqueness of D numbering within the TO units, e.g. by renaming the D numbers – Do not use the causal instruction CHKDM if uniqueness is not needed in the following. The alarm is for information purposes. It can be suppressed by setting bit 4 of MD 11410 SUPPRESS_ALARM_MASK.
Continue program	Alarm display with cause of the alarm disappears. No further operator action required.

Alarm no.	
17191	Channel %1 block %2 T= %3, does not exist, program %4
Explanation	A tool identifier which the NCK does not recognize was programmed.
Reaction	Alarm display Interface signals are set Correction block with reorganization
Remedy	%1 = channel number, %2 = block number, label, %3 = T number or T identifier, %4 = program name If the program pointer is at an NC block that contains the specified T identifier: If the program is incorrect, remedy the error with a correction block and continue the program. Create a data record if the data block is missing. I.e. load the data block for the tool with all defined D numbers to the NCK (via HMI with Overstore) and then continue the program. If the program pointer is at an NC block that does not contain the specified T identifier: The error occurred at an earlier point in the program where the T command appeared, but the alarm was not output until the change command was detected. If the program contains an error, e.g. T5 programmed instead of T55, the current block can be corrected with a correction block; i.e. if it contains only M06, then it can be corrected to T55 M06. The incorrect T5 line remains in the program until it is terminated by a RESET or end of program. In complex program structures with indirect programming, it may not be possible to correct the program. In this case, you can only intervene locally with an overstore block – with T55 in the example. Create a data record if the data block is missing. I.e. load the data block for the tool with all defined D numbers to the NCK (via HMI, with Overstore), program T with Overstore and then continue the program.
Continue program	Cancel the alarm with NC START and continue processing.

Alarm no.	
17192	TO units %1 invalid tool naming of %2, duplo no. %3. No further replacement tools in %4 possible
Explanation	%1 = TO units number, %2 = tool identifier, %3 = duplo number of the tool to be renamed, %4 = group identifier only possible for an active tool management The tool with the specified tool identifier, duplo number cannot accept the group identifier. Reason: The maximum permissible number of replacement tools has already been defined. The name entered for the tool has assigned or changed the assignment of the tool to a tool group which already includes the maximum permissible number of replacement tools for this particular machine.

Alarm no.	
Reaction	Alarm display Interface signals are set
Remedy	Define fewer replacement tools. Unload replacement tools that are no longer required and delete their data in the NCK. Request other settings for the maximum number from the machine manufacturer.
Continue program	Alarm display with cause of the alarm disappears. No further operator action required.

Alarm no.	
17193	Channel %1 block %2 the active tool is no longer on toolholder no./spindle no. %3, program %4
Explanation	%1 = channel number, %2 = block number, label, %3 = toolholder no., spindle no., %4 = program name The tool on the specified toolholder/spindle on which the last tool change was performed as the master toolholder or master spindle, has been replaced. Example: N10 SETHTH(1) N20 T=“Wz1” ;Tool change on master toolholder 1 N30 SETMTH(2) N40 T1=“Wz2” ;Toolholder1 is only a secondary_toolholder. ;Replacing the tool does not cause offset deselection. N50 D5 ;New offset selection. There is currently no active tool to which D can refer.
Reaction	Alarm display Interface signals are set
Remedy	– Modify program: – Set the required spindle as the main spindle or the toolholder as the master toolholder. – Then reset any main spindles or master toolholders.
Continue program	Cancel the alarm with NC START and continue processing.

Alarm no.	
17194	Channel %1 block %2 no suitable tool found
Explanation	%1 = channel number, %2= block number, label – An attempt was made to access a tool which has not been defined. – The specified tool does not permit access. – A tool with the desired properties is not available.

Alarm no.	
Reaction	Alarm display Interface signals are set Correction block with reorganization
Remedy	– Check access to tool: – Are the parameters of the command correctly programmed? – Does the status of the tool prevent access?
Continue program	Cancel the alarm with NC START and continue processing.

Alarm no.	
17200	Channel % 1 block % 2 tool data cannot be deleted
Explanation	%1 = channel number, %2 = block number, label An attempt has been made to delete from the part program the tool data for a tool currently being processed. Tool data for tools involved in the current machining operation may not be deleted. This applies both for the tool preselected with T or that has been changed in place of another, and also for tools for which the constant grinding wheel peripheral speed or tool monitoring is active.
Reaction	Alarm display Interface signals are set Correction block with reorganization
Remedy	Check access to tool offset memory by means of $TC_DP1[t,d] = 0 or deselect tool.
Continue program	Cancel the alarm with NC START and continue processing.

Alarm no.	
17202	Channel %1 block %2 magazine data cannot be deleted
Explanation	%1 = channel number, %2 = block number, label You have attempted to delete magazine data at a time when they cannot be deleted. The data for a magazine which currently has the status “Tool is moving” cannot be deleted. A tool adapter currently assigned to a magazine location cannot be deleted. A tool adapter cannot be deleted if machine data $MN_MM_NUM_TOOL_ADAPTER is set to –1.
Reaction	Alarm display Interface signals are set Correction block with reorganization
Remedy	If your attempt to delete a magazine is rejected: Make sure that the relevant magazine does not have the “Tool is moving” status when you enter the Delete command. If the attempt to clear a tool adapter fails, then it has to be removed from the magazine location or from the magazine locations by clearing the data.
Continue program	Cancel the alarm with NC START and continue processing.

Alarm no.	
17212	Channel %1 tool management: Load manual tool %3, duplo no. %2 onto spindle/toolholder %4
Explanation	%1 = channel number, %2 = duplo number, %3 = tool identifier, %4 = toolholder (spindle) number Indication that the specified manual tool must be brought to the specified toolholder or spindle before the program is continued. A manual tool is a tool whose data are registered in the NCK, but which is not assigned to a magazine location. As a result, it is not fully accessible for the purpose of automatic tool changes by the NCK or other operations on the machine.
Reaction	Alarm display
Remedy	– Assure that the specified tool is placed on the toolholder. The alarm is automatically canceled once the tool change ON command has been acknowledged by the PLC.
Continue program	Alarm display with cause of the alarm disappears. No further operator action required.

Alarm no.	
17214	Channel %1 tool management: Remove manual tool %3 from spindle/toolholder %2
Explanation	%1 = channel number, %2 = toolholder (spindle) number %3 = tool identifier, Indication that the specified manual tool must be removed from the specified toolholder or spindle before the program is continued. A manual tool is a tool whose data are registered in the NCK, but which is not assigned to a magazine location. As a result, it is not fully accessible for the purpose of automatic tool changes by the NCK or other operations on the machine.
Reaction	Alarm display
Remedy	– Assure that the specified tool is removed from the toolholder. After the PLC has acknowledged the tool change ON command, the alarm is automatically deleted. Manual tools can only be used efficiently if this is supported by the PLC program.
Continue program	Alarm display with cause of the alarm disappears. No further operator action required.

Alarm no.	
17216	Channel %1 tool management: remove manual tool from spindle/toolholder %4 and load manual tool %3, duplo no. %2.
Explanation	%1 = channel number, %2 = duplo number, %3 = tool identifier, %4 = toolholder (spindle) number Indicates that the specified manual tool must be loaded in the specified toolholder or spindle before the program is continued and that the manual tool located there must be removed. A manual tool is a tool whose data are registered in the NCK, but which is not assigned to a magazine location. As a result, it is not fully accessible for the purpose of automatic tool changes by the NCK or other operations on the machine.
Reaction	Alarm display
Remedy	– Make sure that the manual tools are exchanged. The alarm is automatically canceled once the tool change ON command has been acknowledged by the PLC. Manual tools can only be used efficiently if this is supported by the PLC program.
Continue program	Alarm display with cause of the alarm disappears. No further operator action required.

Alarm no.	
17220	Channel %1 block %2 tool not existing
Explanation	%1 = channel number, %2= block number, label The attempt has been made to access a tool by means of a T number that has not (yet) been defined, e.g. if tools shall be placed at magazine locations by programming $TC_MPP6 = “toolNo”. This is possible only when both the magazine location and the tool given by “toolNo” have been defined.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct the NC program
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
17224	Channel %1 block %2 tool T/D=%3 – tool type %4 is not permitted
Explanation	%1 = channel number, %2 = block number, label %3 = refused T/D no., %4 = refused tool type It is not possible on this system to select tool offsets for tools of the specified tool type. The multitude of tool types can be restricted by the machine manufacturer or by the individual control models. Only use tools with types permitted by this system. Check whether an error occured in definition of the tool.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct NC program or tool data
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
17230	Channel %1 block %2 Duplo no. already assigned
Explanation	%1 = channel number, %2= block number, label The attempt was made to write the duplo number of a tool using a duplo number that has already been assigned to another tool (different T number).
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct the NC program
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
17240	Channel %1 block %2 illegal tool definition
Explanation	%1 = channel number, %2= block number, label The attempt was made to change a tool datum that would subsequently destroy the data consistency or would lead to contradictions in the definition.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct the NC program
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
17250	Channel %1 block %2 illegal magazine definition
Explanation	%1 = channel number, %2= block number, label The attempt was made to change a magazine datum that would subsequently destroy the data consistency or would lead to contradictions in the definition.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct the NC program
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
17260	Channel %1 block %2 illegal magazine location definition
Explanation	%1 = channel number, %2= block number, label The attempt was made to change a magazine datum that would subsequently destroy the data consistency or would lead to contradictions in the definition.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct the NC program
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
17262	Channel % 1 block% 2 illegal tool adapter operation
Explanation	%1 = channel number, %2= block number, label This alarm is generated if you attempt to define or cancel the assignment between a tool adapter and a magazine location and the selected location already has another tool adapter and/or is already holding a tool or, if you are canceling the assignment, there is still another tool in the location. If machine data $MC_MM_NUM_SUMCORR has the value –1, adapters cannot be generated by a write operation to an adapter which is not yet defined. While the machine data has this value, you can only write adapter data to adapters which have already been (automatically) assigned to magazine locations.
Reaction	Alarm display Interface signals are set Correction block with reorganization
Remedy	– Assign a maximum of one adapter to a magazine location – The magazine location must not contain a tool. – Machine data $MC_MM_NUM_SUMCORR has value –1: If the alarm is generated when you are writing one of the system parameters $TC_ADAPTx (x=1,2,3,T), then you must change the write operation to ensure that it includes only adapter data which is already assigned to magazine locations.
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
20150	Channel %1 tool management: PLC terminates interrupted command
Explanation	%1 = channel number Indication that the PLC has terminated an interrupted command (with alarm output) from the tool management – tool change.
Reaction	Alarm display Interface signals are set
Remedy	For information only.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
20160	Channel %1 tool management: PLC can only terminate commands interrupted due to an error
Explanation	%1 = channel number Indication that the PLC wanted to interrupt an active command from the tool management tool change; or that there is no command active for abort. NCK refuses because the channel status is either “active” (cancel is then not allowed), or “reset” (then there is nothing to cancel).
Reaction	Alarm display Interface signals are set
Remedy	For information only.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
22066	Channel %1 tool management: tool motion not possible since there is no tool %2 with duplo no. %3 in magazine %4
Explanation	%1 = channel number, %2 = string (identifier), %3 = duplo number, %4 = magazine number The desired tool change is not possible. The specified tool is not contained in the specified magazine. (NCK cannot contain tools that are not assigned to a magazine. No operations (movement, change) can be performed by these tools.
Reaction	NC Start disable Alarm display Interface signals are set NC stop for alarm
Remedy	Please inform the authorized personnel/service department. – Ensure that the specified tool is in the desired magazine or program a different tool that shall then be substituted – Check whether the machine data $MC_RESET_MODE_MASK, $MC_START_MODE_MASK, $MC_TOOL_RESET_NAME match the current definition data.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
22067	Channel %1 tool management: tool change not possible, since no tool ready for use in tool group %2
Explanation	%1 = channel number, %2 = string (identifier) The desired tool change is not possible. The specified tool group does not contain a “ready to use” replacement tool which could be loaded at change. The tool monitoring function may have set all potentially suitable tools to the “disabled” status. Alarm 22067 is generated in a situation where it is no longer possible to intervene in a correcting manner.
Reaction	NC Start disable Alarm display Interface signals are set NC stop for alarm
Remedy	– Ensure that the specified tool group contains a tool that is ready for use when tool change is requested. – This can be achieved, for example, by replacing disabled tools – or by manually releasing a disabled tool. – Check whether the magazine data is correctly defined. Have all intended tools in the group been defined with the specified identifier and loaded?
Continue program	Clear alarm with the RESET key. Restart part program.

Alarm no.	
22068	Channel %1 tool management: no tool ready for use in tool group %3
Explanation	%1 = channel number, %2 = block number, label, %3 = string (identifier) The specified tool group does not contain a “ready to use” replacement tool which could be loaded at change. The tool monitoring function may have set all potentially suitable tools to the “disabled” status. The alarm can occur in conjunction with alarm 14710. In this specific situation, NCK attempts to replace the disabled tool located on the spindle with an available replacement tool (which does not exist in this error condition). The user must resolve this conflict, for example, by removing the tool located on the spindle from the spindle by issuing a motion command (e.g. through HMI operation).
Reaction	NC Start disable Alarm display Interface signals are set
Remedy	– Ensure that the specified tool group contains a tool that is ready for use when tool change is requested. – This can be achieved, for example, by replacing disabled tools – or by manually releasing a disabled tool. – Check whether the magazine data is correctly defined. Have all intended tools in the group been defined with the specified identifier and loaded?
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
22069	Channel %1 block %2 tool management: No tool available in tool group %3, program %4
Explanation	%1 = channel number, %2 = block number, label, %3 = string (identifier), %4 = program name The specified tool group does not contain a “ready to use” replacement tool which could be loaded at change. The tool monitoring function may have set all potentially suitable tools to the “disabled” status. Parameter %4 = program name facilitates the identification of the program containing the programming command (tool selection) that caused the error. This can be a subprogram or cycle, etc., which can no longer be identified from the display.
Reaction	Alarm display Interface signals are set Correction block with reorganization
Remedy	– Ensure that the specified tool group contains a tool that is ready for use when tool change is requested, e.g. by: – Replacing disabled tools, – Manually releasing a disabled tool. – Check whether the magazine data is correctly defined. Have all intended tools in the group been defined with the specified identifier and loaded?
Continue program	Cancel the alarm with NC START and continue processing.

Alarm no.	
22070	TO unit %1 Please change tool T= %2 into magazine. Repeat data back-up.
Explanation	%1 = TO unit, %2 = T number of the tool Only issued if tool management is active. A data backup of the tool/machine data has been started. The system has detected that the buffer magazine still contains one or more tools. During backup, these tools lose the information assigning them to a magazine and a location in the magazine. It is therefore useful at the time of data backup to have filed all the tools in the magazine. If the above scenario does not apply, you have re-imported data with magazine locations set to the “reserved” status. You may have to reset this status manually. In the case of tools with a fixed-location coding, the loss of information about their location in the magazine is equivalent to a general empty location search on any subsequent change back to the magazine.
Reaction	Interface signals are set. Alarm display.

Alarm no.	
Remedy	Make sure that there are no tools stored in the buffer magazine before you start to back up data. Repeat the data backup after removing the tools from the buffer magazine.
Continue program	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
22071	TO unit %1 tool %2 duplo no. %3 is active but not in the current wear group
Explanation	%1 = TO units, %2 = T number of the tool, %3 = duplo number The “Wear grouping” function is active. The setting “Set tool to active status” which applies when a new wear grouping is activated is also selected. This setting can also be programmed with language command SETTA or started via Analog Functions on the OPI. It has been detected that more than one tool from the tool group has the “active” status. The tool which has the “active” status in an “inactive” wear grouping is named in the alarm. The alarm is for information purposes. It can be suppressed by setting bit 5 of MD 11410 SUPPRESS_ALARM_MASK.
Reaction	Alarm display Set interface signals
Remedy	Before you start the machining operation, make sure that the “active” status is not set for any of the tools in the magazine. You can do this by programming command SETTIA.
Program continuation	Clear the alarm with the cancel key. No further operator action required.

Alarm no.	
400601	
Explanation	The magazine data in the PLC is incorrect. Tool management start-up is faulty if the tool management option is activated.
Reaction	Alarm display PLC STOP
Remedy	Delete DB71 – DB74 and load correct tool management configuration via HMI or correct the settings in DB4.

Alarm no.	
400602	
Explanation	The magazine data in the PLC is incorrect. Tool management start-up is faulty if the tool management option is activated.
Reaction	Alarm display PLC STOP
Remedy	Delete DB71 – DB74 and load correct tool management configuration via HMI or correct the settings in DB4.

Alarm no.	
400603	
Explanation	The magazine data in the PLC is incorrect. Tool management start-up is faulty if the tool management option is activated.
Reaction	Alarm display PLC STOP
Remedy	Delete DB71 – DB74 and load correct tool management configuration via HMI or correct the settings in DB4.

Alarm no.	
400604	Set change with M06 in machine data
Explanation	Change is possible only with M06 for the magazine type used (box, chain). Check for invalid settings when using turret magazines.
Reaction	Alarm display PLC STOP
Remedy	The value is 1 in channel-specific machine data 22550 tool_CHANGE_MODE

Alarm no.	
410151	Magazine data for tool management missing in PLC
Explanation	No magazine data available in the PLC. Start-up not complete although tool management option is active.
Reaction	Alarm display
Remedy	Press the “Create PLC data” softkey via HMI Advanced during start-up of the tool management. Set the data in data block DB4 starting at DBB64 for HMI Embedded.

Notes

PLC sample programs 11

This section gives practical examples which illustrate how function blocks can be adapted to suit a variety of realistic configurations.

These sample programs are stored in file wzv_bsp.exe in catalog Bsp_prog in the SINUMERIK 810D/840D Toolbox.

11.1 FB 90: QUIT_WZVacknowledgments to TOOLMAN

Description of functions

FB QUIT_WZV supports the user in the acknowledgement of TOOLMAN (TOOLMAN) jobs as well as signaling position changes by tools to tool management and updating the PLC point of tool change when using integrated tool management.

32 user interfaces are available for transfer tasks to the tool management (WZV) in the instance DB FB QUIT_WZV. Data is transferred to the NCK in FB QUIT_WZV using call FC TM_TRANS (FC 8). The parameters of FC TM_TRANS are defined as a variable in FB QUIT_WZV and must be assigned a value for each user interface. The symbolic names of the variables have the same names as the formal parameters of FC TM_TRANS. See the Description of Function basic PLC program in Chapter 4 of the Block description FC 8 for more information about the parameters.

Specifically, the following variables must be assigned values in the branch target list of each user interface:

- TaskIdent
- TaskIdentNo
- NewToolMag
- NewToolLoc
- OldToolMag
- OldToolLoc
- Status

If the tool transfer from the magazine to the spindle is via temporary storage (e.g. gripper), then the variables

– NewToolMag_ChangeI_S1
– NewToolLoc_Change_S1
– OldToolMag_Change_S1
– OldToolLoc_Change_S1

must also be assigned values when changing to spindle 1.

For spindle 2, these variables have the same name with the ending '_S2'. If the tool change operation is reset or aborted, then these variables must be used to assign the FC TM_TRANS parameters.

With a 1 signal at a user interface (DIB 0 – DIB 3), FC TM_TRANS is called with the parameter values programmed in the branch target list.

If the task is completed successfully (FC TM_TRANS Ready = 1), the user interface bit is reset by FB QUIT_WZV. If the task or transfer of FC TM_TRANS produces an error, error bit DIX 4.0 in the instance DB is set to 1 signal and the output parameter error of FC TM_TRANS is available as error number in DIW 6.

The user interface is reset in the event of an error. Further tasks are only processed after the error bit has been reset (by the user). For the meaning of the error numbers, please refer to Description of Functions, Basic PLC Program, Chapter 4, Description of Block FC 8 under the parameter 'Error'.

If several tasks are present simultaneously, the user interfaces (UI) are processed according to the following priority:

1. UI 25 → UI 32
2. UI 17 → UI 24
3. UI 9 → UI 16
4. UI 1 → UI 8

You must enter the actual magazine position of the tool change point in accordance with the selected FB-QUIT in the instance DB starting at DIW 10.

Declaration

```
FUNCTION_BLOCK FB 90
// no parameters
// user interface in the instance DB
```

Block call

```
CALL FB 90, DB xxx;          // xxx No. of instance DB
```

User interface

The user interface is stored in the instance DB from DIB 0 to DIB 46.
Bytes 47 to 64 are internal variables of FB QUIT_WZV, which can be read out for support during installation if required. The variables ASS_alt (UI_old), ASS_Aenderung (UI_change) and ASS_aktiv (UI_active) have the same assignment as ASS_neu (UI_new) (DBB 0 to DBB 3).

DB Instance	User Interface							
Byte	Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIB 0	UI 8	UI 7	UI 6	UI 5	UI 4	UI 3	UI 2	UI 1
DIB 1	UI 16	UI 15	UI 14	UI 13	UI 12	UI 11	UI 10	UI 9
DIB 2	UI 24	UI 23	UI 22	UI 21	UI 20	UI 19	UI 18	UI 17
DIB 3	UI 32	UI 31	UI 30	UI 29	UI 28	UI 27	UI 26	UI 25
DIB 4								Error
DIB 5	–							
DIB 6	Error number							
DIB 8	–							
DIB 10	ActPosChangePosMag1							
DIB 12	ActPosChangePosMag2							
DIB 14	ActPosChangePosGr1							
DIB 16	ActPosChangePosGr2							
DIB 18	–							
DIB 20	TaskIdent							
DIB 21	TaskIdentNo							

DB Instance	User Interface
DIW 22	NewToolMag
DWI 24	NewToolLoc
DWI 26	OldToolLoc
DWI 28	Status
DWI 30	NewToolMag_Change_S1
DWI 32	NewToolLoc_Change_S1
DWI 34	OldToolMag_Change_S1
DWI 36	OldToolMag_Change_S1
DWI 38	OldToolLoc_Change_S1
DWI 40	NewToolMag_Change_S2
DWI 42	NewToolLoc_Change_S2
DWI 44	OldToolMag_Change_S2
DWI 46	OldToolLoc_Change_S2

<table>
<tr><th>DB Instance</th><th colspan="8">User Interface</th></tr>
<tr><td>DID 48</td><td colspan="8">UI_old</td></tr>
<tr><td>DID 52</td><td colspan="8">UI_change</td></tr>
<tr><td>DID 56</td><td colspan="8">UI_active</td></tr>
<tr><td>DIB 60</td><td></td><td></td><td></td><td></td><td></td><td></td><td>Ready</td><td>Start</td></tr>
<tr><td>DIB 61</td><td colspan="8">–</td></tr>
<tr><td>DIW 62</td><td colspan="8">Error FC 8</td></tr>
<tr><td>DIB 64</td><td colspan="8">StepNo</td></tr>
</table>

Abort/reset

If a task in progress such as “Load tool” , “Unload tool”, “Prepare change” or “Execute change” is aborted by the NC Reset or Emergency Stop signal, the PLC must acknowledge the task with FC TM_TRANS, status 3, if the task has not been completed. A task acknowledgement with status 3 is acknowledged negatively by the tool management with error no. 6405. This behavior is taken into account in FB QUIT_WZV in the error evaluation of FC TM_TRANS. No error is output here.

Configuration/Startup

When supplying parameters for the FC TM_TRANS, it is important to ensure that the correct magazine locations are assigned for the parameters NewToolMag/Loc and OldToolMag/Loc on each status change or end-of-job acknowledgement. The same applies for TaskIdent and TaskIdentNo. The tool management checks each parameter against FC TM_TRANS on acknowledgement. If an incorrect value is detected by the tool management, the NC assumes the STOP state and NC error 6405 “Channel %1 command %2 has invalid PLC acknowledgement parameter %3” appears. If such a faulty condition occurs, then the variables of the parameters of the FC TM_TRANS can be read in the PLC status and checked.

The status of the variables ASS_aktiv (DIB 44 DIB 47) shows which was the last task to be processed. The assignment of ASS_aktiv is identical to the ASS interface (DIB 0–DIB 3).

Power OFF / Restart

If the NCK is disconnected from the mains during a job or a NCK reset is executed, then the user-interface bits have to be deleted by the user.
Also, the following variables in the DB instance must be deleted in OB 100:

```
Open DB xxx;   // Open DB instance FB QUIT_WZV
L 0;
T DBD 48;      // UI_old
```

```
T DBD 52;    // UI_change
T DBD 56;    // UI_active
T DBB 60;    // Start and ready FC 8
```

As of version 2.0 of FB 90, the variables in FB 90 are deleted on restart. The instruction section can be omitted in OB100.

11.1.1 Sample Programs

Sample Programs

As an example of the use of FB QUIT_WZV, five different magazine configurations are programmed in FB 90. The setting for the user interface bits in FB 90 is programmed in FC 90. The blocks are contained in files QUIT_1.awl – QUIT_2.awl.

The following magazine types have been implemented as program examples:

- Chain magazine with one spindle as a pick-up magazine
- Chain magazine with one dual gripper and one spindle
- Chain magazine with two grippers and one spindle
- Two chain magazines with one spindle
- Chain magazine with two spindles.

11.1.2 Chain magazine with one spindle as a pick-up magazine

Description

FB QUIT_WZV is programmed as FB 90 in QUIT_1.awl for the following magazine configuration:

Magazine no. c	Location no.	Meaning
9999	1	Spindle loading point
9999	2	Magazine loading point
9998	1	Spindle
1	1	Magazine location 1
1	2	Magazine location 2
1		Magazine location .

Magazine no. c	Location no.	Meaning
1	n–1	Magazine location n–1
1	n	Magazine location n

The tool is changed by moving directly from the magazine to the spindle (pick-up magazine). If a tool is already located in the spindle it is returned to the magazine before the new tool is placed in the magazine. Loading is performed either via the loading point magazine or the loading point spindle.
For this configuration, 17 transfer job requests from the PLC to the tool management are programmed in FB QUIT_WZV. These requests are triggered by the user via the user interfaces UI 1 – UI 20.
With an asynchronous job request, tool position changes outside a programmed sequence, e.g. for movements in JOG, can be sent to the tool management after a tool change has been aborted.

The following job requests are implemented in FB 90 and triggered in FC 90 in the example in QUIT_1.awl:

OCV	Function
1	Acknowledgement load tool completed, magazine loading point
2	Abort / reset load tool, magazine loading point
3	Acknowledgement unload tool completed, magazine loading point
4	Abort / reset unload tool, magazine loading point
5	Acknowledgement load tool completed, spindle loading point
6	Abort / reset load tool, spindle loading point
7	Acknowledgement unload tool completed, spindle loading point
8	Abort / reset unload tool, spindle loading point
9	Acknowledgement prepare change completed
10	Abort / reset prepare change
11	Spindle change status → magazine tool change
12	Magazine change status → spindle tool change
13	Abort / reset change
14	–
15	Acknowledgement relocate (from HMI)
16	Asynchronous relocation spindle → magazine
17	Asynchronous relocation spindle → spindle
18	–
19	–
20	Actual position change magazine location change point
21	–

OCV	Function
22	Acknowledgement positioning at loading point
23	Abort / reset positioning at loading point

... ...

32	–

The actual position for job requests 16, 17 and 20 is taken from DB instance DBW 10 in FB 90. The actual position address can be changed by the user.

NOTICE!

With asynchronous relocation the magazine location state “Z” (reserved for tool in buffer) is not taken into account. This means that with asynchronous relocation from magazine to spindle, the identifier “Z” is not set and with asynchronous relocation from spindle to magazine that the identifier “Z” is not reset in the old location.

In this case, “Z” must be set and cleared with FB 3 (write NC variable).
With NC SW 3.2 and later, magazine location status “Z” is transferred with Task-Ident 5 for asynchronous relocation.

11.1.3 Chain magazine with one dual gripper and one spindle

Description

FB QUIT_WZV is programmed as FB 90 in QUIT_2.awl for the following magazine configuration:

Magazine no.	Location no.	Meaning
9999	1	Spindle loading point
9999	2	Magazine loading point
9998	1	Spindle
9998	1	Dual gripper, gripper 1
9998	1	Dual gripper, gripper 2
1	1	Magazine location 1
1	2	Magazine location 2

Magazine no.	Location no.	Meaning
1	.	Magazine location .
1	n	Magazine location n

The tool is moved via the dual gripper from the tool change position in the magazine to the spindle. The tools are simultaneously changed to and from the magazine and the spindle. Before the tool is changed, the gripper on the magazine side is gripper 2 and the gripper on the spindle side is gripper 1.
With this definition only two relocation commands are necessary.

Loading is performed either via the loading point magazine or the loading point spindle.
For this configuration, 19 transfer job requests from the PLC to the tool management are programmed in FB QUIT_WZV. These requests are triggered by the user via the user interfaces UI 1–UI 20.
With an asynchronous job request, tool position changes outside a programmed sequence, e.g. for movements in JOG, can be sent to the tool management after a tool change has been aborted.

The following job requests are implemented in FB 90 and triggered in FC 90 in the example in QUIT_2.awl:

OCV	Function
1	Acknowledgement load tool completed, magazine loading point
2	Abort / reset load tool, magazine loading point
3	Acknowledgement unload tool completed, magazine loading point
4	Abort / reset unload tool, magazine loading point
5	Acknowledgement load tool completed, spindle loading point
6	Abort / reset load tool, spindle loading point
7	Acknowledgement unload tool completed, spindle loading point
8	Abort / reset unload tool, spindle loading point
9	Acknowledgement prepare change completed
10	Abort / reset prepare change
11	Status change spindle →gripper 1 and magazine → gripper 2 tool change
12	Status change magazine → magazine and gripper 2 → spindle tool change
13	Abort / reset change
14	–
15	Acknowledgement relocate (from HMI)
16	Asynchronous relocation gripper 1 → spindle
17	Asynchronous relocation gripper 1 → magazine
18	Asynchronous relocation gripper 2 → spindle

OCV	Function
19	Asynchronous relocation gripper 2 → magazine
20	Actual position change magazine location change point
21	–
22	Acknowledgement positioning at loading point
23	Abort / reset positioning at loading point
...	...
32	–

The actual position for job requests 17, 19 and 20 is taken from DB instance DIW 10 in FB 90. The actual position address can be changed by the user.

NOTICE!
With asynchronous relocation the magazine location state “Z” (reserved for tool in buffer) is not taken into account. This means that with asynchronous relocation from magazine to spindle, the identifier “Z” is not set and with asynchronous relocation from spindle to magazine that the identifier “Z” is not reset in the old location. In this case, “Z” must be set and cleared with FB 3 (write NC variable). With NC SW 3.2 and later, magazine location status “Z” is transferred with Task-Ident 5 for asynchronous relocation.

11.1.4 Chain magazine with two grippers and one spindle

Description

FB QUIT_WZV is programmed as FB 90 in QUIT_3.awl for the following magazine configuration:

Magazine no.	Location no.	Meaning
9999	1	Spindle loading point
9999	2	Magazine loading point
9998	1	Spindle
9998	2	Gripper 1
9998	3	Gripper 2
1	1	Magazine location 1
1	2	Magazine location 2
1	.	Magazine location .

Magazine no.	Location no.	Meaning
1	n	Magazine location n–1
1	n	Magazine location n

The tool is relocated from the tool change position in the magazine into the spindle via gripper 1 or gripper 2 and from the spindle into the magazine via gripper 2. Tools can only be loaded via the loading point of the magazine.

For this configuration, 20 transfer job requests from the PLC to the tool management are programmed in FB QUIT_WZV.
These tasks must be initiated by the user via the user interfaces UI 1–UI 20.
With an asynchronous job request, tool position changes outside a programmed sequence, e.g. for movements in JOG, can be sent to the tool management after a tool change has been aborted.

The following job requests are implemented in FB 90 and triggered in FC 90 in the example in QUIT_3.awl:

OCV	Function
1	Acknowledgement load tool completed, magazine loading point
2	Abort / reset load tool, magazine loading point
3	Acknowledgement unload tool completed, magazine loading point
4	Abort / reset unload tool, magazine loading point
5	Acknowledgement prepare change completed
6	Abort / reset prepare change
7	Status change magazine → gripper 1 tool change
8	Status change magazine → gripper 2 tool change
9	Status change spindle → gripper 2 tool change
10	Status change gripper 1 → spindle tool change
11	Status change gripper 2 → Magazine tool change
12	Abort / reset change
13	Acknowledgement relocate (from HMI)
14	Asynchronous relocation gripper 1 → magazine
15	Asynchronous relocation gripper 2 → magazine
16	Asynchronous relocation gripper 1 → spindle
17	Asynchronous relocation gripper 2 → spindle
18	Asynchronous relocation spindle → gripper 1
19	Asynchronous relocation spindle → gripper 2
20	Actual position change magazine location change point
21	–

OCV	Function
22	Acknowledgement positioning at loading point
23	Abort / reset positioning at loading point
...	...
32	–

The actual position is read from the DB instance DIW 10 for job 20 in FB 90.
The actual magazine position for gripper 1 is read from the DB instance DIW14 (UI 14) and the actual magazine position for gripper 2 from DB instance DIW16 (UI 15). The addresses of the actual positions can be changed by the user.
Neither loading nor unloading spindles has been programmed in FB QUIT_WZV. With a user-programmable UI, this function can be programmed by the user. Jump target lists IFC 1–IFC 3 can be used as an example.

NOTICE!
With asynchronous relocation the magazine location state “Z” (reserved for tool in buffer) is not taken into account. This means that with asynchronous relocation from magazine to spindle, the identifier “Z” is not set and with asynchronous relocation from spindle to magazine that the identifier “Z” is not reset in the old location. In this case, “Z” must be set and cleared with FB 3 (write NC variable).
With NC SW version 3.2 and higher, magazine location status “Z” is transferred with Task-Ident 5 for asynchronous relocation.

11.1.5 Two chain magazines with one spindle as a pick-up magazine

Description

FB QUIT_WZV is programmed as FB 90 in QUIT_4.awl for the following magazine configuration:

Magazine no.	Location no.	Meaning
9999	1	Spindle loading point
9999	2	Magazine loading point
9998	1	Spindle
1	1	Magazine location 1
1	2	Magazine location 2
1	.	Magazine location .
1	n–1	Magazine location n–1
1	n	Magazine location n

Magazine no.	Location no.	Meaning
1	1	Magazine location 1
1	2	Magazine location 2
1	.	Magazine location .
1	n	Magazine location n–1
1	n	Magazine location n

The tool is moved directly from magazine 1 or magazine 2 to the spindle (pick-up magazine). If a tool is already located in the spindle it is returned to magazine 1 or magazine 2 before the new tool is placed in the magazine.
Loading is performed either via the loading point magazine or the loading point spindle. For this configuration, 22 transfer job requests from the PLC to the tool management are programmed in FB QUIT_WZV. These requests are triggered by the user via the user interfaces UI 1–UI 22.
With an asynchronous job request, tool position changes outside a programmed sequence, e.g. for movements in JOG, can be sent to the tool management after a tool change has been aborted.

The following job requests are implemented in FB 90 and triggered in FC 90 in the example in QUIT_4.awl:

OCV	Function
1	Acknowledgement load tool completed, magazine loading point
2	Abort / reset load tool, magazine loading point
3	Acknowledgement unload tool completed, magazine loading point
4	Abort / reset unload tool, magazine loading point
5	Acknowledgement load tool completed, spindle loading point
6	Abort / reset load tool, spindle loading point
7	Acknowledgement unload tool completed, spindle loading point
8	Abort / reset unload tool, spindle loading point
9	Acknowledgement prepare change completed
10	Abort / reset prepare change
11	Status change spindle → magazine tool change
12	Magazine change status → spindle tool change
13	Abort / reset change
14	–
15	Acknowledgement relocate (from HMI)
16	Asynchronous relocation spindle → magazine 1
17	Asynchronous relocation magazine 1 → spindle
18	Asynchronous relocation spindle → magazine 2

OCV	Function
19	Asynchronous relocation magazine 2 → spindle
20	Actual position change magazine location change point magazine 1
21	Actual position change magazine location change point magazine 2
22	Acknowledgement positioning at loading point
23	Abort / reset positioning at loading point
...	...
32	–

The actual position is read from the DB instance DIW10 or DIW12 for job 20 in FB 90. The addresses of the actual positions can be changed by the user.

NOTICE!
With asynchronous relocation the magazine location state “Z” (reserved for tool in buffer) is not taken into account. This means that with asynchronous relocation from magazine to spindle, the identifier “Z” is not set and with asynchronous relocation from spindle to magazine that the identifier “Z” is not reset in the old location. In this case, “Z” must be set and cleared with FB 3 (write NC variable).
With NC SW 3.2 and later, magazine location status “Z” is transferred with Task-Ident 5 for asynchronous relocation.

11.1.6 Chain magazine with two spindles.

Description

FB QUIT_WZV is programmed as FB 90 in QUIT_5.awl for the following magazine configuration:

Magazine no.	Location no.	Meaning
9999	1	Spindle loading point 1
9999	2	Spindle loading point 2
9999	2	Magazine loading point
9998	1	Spindle 1
9998	2	Spindle 2
1	1	Magazine location 1
1	2	Magazine location 2
1	.	Magazine location .

Magazine no.	Location no.	Meaning
1	n	Magazine location n–1
1	n	Magazine location n

The tool is moved directly from the magazine to spindle 1 or spindle 2 (pick-up magazine). If a tool is already located in the spindle it is returned to the magazine before the new tool is placed in the magazine.

Spindle 1 is assigned to channel 1 and spindle 2 to channel 2. Therefore a tool function or tool change programmed in channel 1 is outputted in DB 72 in UI 1 and the new tool is placed on spindle 1.
Therefore, a tool call or tool change programmed in channel 2 is outputted in DB 72 in UI 2 and the new tool is placed on spindle 2. Tools can only be loaded via the loading point of the magazine.

For this configuration, 20 transfer job requests from the PLC to the tool management are programmed in FB QUIT_WZV. These tasks must be initiated by the user via the user interfaces UI 1 – UI 20.
With an asynchronous job request, tool position changes outside a programmed sequence, e.g. for movements in JOG, can be sent to the tool management after a tool change has been aborted.

The following job requests are implemented in FB 90 and triggered in FC 90 in the example in QUIT_5.awl:

OCV	Function
1	Acknowledgement load tool completed, magazine loading point
2	Abort / reset load tool, magazine loading point
3	Acknowledgement unload tool completed, magazine loading point
4	Abort / reset unload tool, magazine loading point
5	Acknowledgement prepare change completed spindle 1
6	Abort / reset prepare change spindle 1
7	Acknowledgement prepare change completed spindle 2
8	Abort / reset prepare change spindle 2
9	Status change spindle 1 → magazine
10	Status change magazine → spindle 1
11	Abort / reset change spindle 1
12	Status change spindle 2 → magazine
13	Status change magazine → spindle 2
14	Abort / reset change spindle 2
15	Acknowledgement relocate (from HMI)
16	Asynchronous relocation spindle 1 → magazine

OCV	Function
17	Asynchronous relocation magazine → spindle 1
18	Asynchronous relocation spindle 1→ magazine
19	Asynchronous relocation magazine → spindle 2
20	Actual position change magazine
21	–
22	Acknowledgement positioning at loading point
23	Abort / reset positioning at loading point
...	....
32	–

The actual position for job request 20 in FB 90 is taken from DB instance DIW 10. The actual position address can be changed by the user.
Neither loading nor unloading spindles has been programmed in FB QUIT_WZV. With a user-programmable UI, this function can be programmed by the user. Jump target lists IFC 1–IFC 3 can be used as an example.

NOTICE!
With asynchronous relocation the magazine location state “Z” (reserved for tool in buffer) is not taken into account. This means that with asynchronous relocation from magazine to spindle, the identifier “Z” is not set and with asynchronous relocation from spindle to magazine that the identifier “Z” is not reset in the old location. In this case, “Z” must be set and cleared with FB 3 (write NC variable).
With NC SW 3.2 and later, magazine location status “Z” is transferred with Task-Ident 5 for asynchronous relocation.

11.2 FB 91: LE_SUCH search for empty location for tool in buffer

Description of functions

A search for an empty location in the magazine for a tool in the buffer can be made with FB LE_SUCH.
Every FB 91 call must be assigned a separate instance DB from the user area.
When FB 91 is called an empty location is searched in the magazine for a tool in the buffer on a positive edge change at control input Start.

The location in the temporary storage is also given to the function block by the input parameters MagNr_ZW and LocNr_ZW. The magazine number where theempty located shall be searched is given by the parameter magazine no.

Successful execution of the job is displayed by means of a logic “1” in status parameter Done. The empty location is output via output parameter MagNr_Empty and LocNo_Empty. Any errors that occurred are indicated by Error and State.

The search for empty locations is a routine executed over several PLC cycles. The block can be called up in cyclic mode only. FB 2 is called twice and FB 4 once in FB 91. These blocks are called with a multi-instance DB in FB91.

Important

FB 91 can perform the empty location search only if basic program parameter NCKomm has been set to “1” (in OB100: FB 1, DB 7).

Declaration

```
FUNCTION_BLOCK FB 91

VAR_INPUT


Start :        BOOL;
MagNr_ZW :     INT;
LocNr_ZW :     INT;
MagNr :        INT;


END_VAR

VAR_OUTPUT

Active:        BOOL;
Done :         BOOL;
Error :        BOOL;
State :        WORD;
MagNr_Empty:   INT;
```

LocNr_Empty: INT;

END_VAR

Description of formal parameters

The table below lists all the formal parameters for block LE_SUCH.

Signal	Type	Type	Value range	Remarks
Start	E	Bool		Start empty location search.
MagNr_ZW	E	Int	1..	Magazine number of the buffer
LocNr_ZWv	E	Int	1..	Location number of buffer
MagNr	E	Int	1..	Magazine number of the magazine where the empty locations shall be searched.
Active	A	Bool		Empty location search running
Done	O	Bool		Empty location found. Signal is active for one PLC cycle.
Error	O	Bool		Empty location search was acknowledged negative or could not be executed. Signal is active for one PLC cycle. Fault no. is stored in State.
State	O	Word		See error identifiers
MagNr_Empty	O	Int		Magazine number for empty location
LocNr_Empty	O	Int		Location number for empty location

Furthermore, the search for an empty location can be influenced with the following signals in the instance DB of the FB91:

TNr_write = 1:
The T number of the tool for the search for an empty location is in TNr_FB2.
MagNr_ZW/LocNo_ZW are not evaluated.
MMCSEM =1:
No setting of semaphores in the PI service TMFDPL.

Error identifiers

If it is not possible to execute a request in the empty location search, this is indicated in status parameter Error with ’logical 1’. The error cause is coded at the block output State:

State	Meaning	Note
1	Error while reading T number (FB 2) from MagNr_ZW and LocNo_ZW.	The fault detection of the FB2 is stored in the variable StateFB2_TNrGesp in the instance DB.
2	The logical T number of the magazine location is zero.	Check whether a tool is at the magazine location of the buffer.
3	Error in the PI service searching for an empty location (FB4).	The error ID of FB 4 is stored in the variable StateFB4Gesp.
4	Error on acknowledgement parameter of PI service read TMFDPL with FB 2.	The error ID of FB 2 is stored in the variable State FB 2_ParGesp instance DB.
5	Search for empty location terminated with error	No empty location available in the magazine
6	Invalid step number	Internal error in FB.
7	Error while reading variable numMagPlaceParams with FB 2.	Restart required.
8	Error FB4 PI service MMCSEM	Semaphore for PI service TMFDPL on 1st event. Another job may be active (HMI)

Pulse diagram

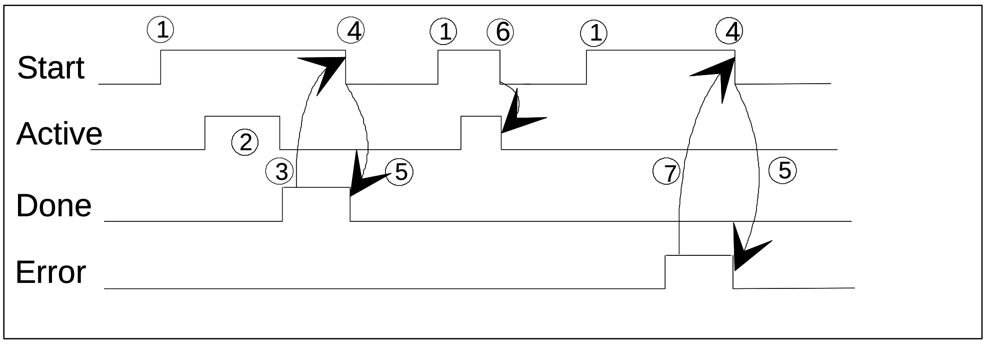


Bild 11-1 Timing diagram for FB 91

1. Activation of function
2. Empty location search active
3. Positive acknowledgment: empty location found
4. Reset of function activation signal after receipt of acknowledgement by user, signal change by FC
5. If function activation signal is reset before receipt of acknowledgement, the output signals are not updated; not relevant once the function is running
6. Negative acknowledgment: Error occurred. Error code in the output parameter State

Call example

```
U       DB21.DBX 204.0;      // M80 signal
S       M 150.0;             // Start empty location search
CAll FB91,DB 91(
Start: M 150.0,              // Start empty location search
MagNr_ZW : 9998,             // Magazine no.= buffer
LocNr_ZW :2,                 // Magazine loc. 2 = gripper

MagNr :         1,           // Magazine no. for empty location = 1
Active:         M 150.1,     // Empty location search active
Done :          M 150.2,     // Empty location found
Error:          M 150.3,     // Error in empty loc. search
State :         MW 152,      // Fault number
MagNr_Empty: MW 154,         // Magazine number for empty location
LocNr_Empty: MW156);         // Location number for empty location

U  M 150.2;                  // Empty location found
O  M 150.3;                  // Error in empty loc. search
R  M 150.0;                  // Start empty location search
U  M 150.3;
S  M 160.0;                  // Error in empty location search
```

Blocks to be loaded

FB 91, FB 2, FB 4, DB 91, DB 119

11.3 FB 92: GET_LOC read magazine location and tool data

Description of functions

The magazine location data of a magazine location and the tool data of a tool can be read with FB GET_LOC.
Every FB 92 call must be assigned a separate instance DB from the user area.
Depending on the signal at input GetWkz, calling FB 92 reads the data on a positive edge change at control input Req. If input GETWKZ carries a 1-signal the magazine location data and tool data is read. If GETWKZ = 0 only the magazine location data is read.
The magazine location is transferred to the FB via input parameters MagNr and LocNr. Successful execution of the function is indicated at status parameter NDR with logical “1”. Any errors that have occurred are output via Error and State.

Specifically, the following data are read:

- Magazine location data (TP):
  - Location state
- General tool data (TD):
  - Size to left in half locations
  - Size to right in half locations
  - Size upwards in half locations
  - Size downwards in half locations
  - Magazine location type
  - Tool status

The data is stored in the instance DB. A detailed description of the data is to be found in the Description Lists in Chapter 4, Variables, and in the Description of Functions Tool Management in the Section Programming.
The Read process is a routine executed over several PLC cycles. The block can be called up in cyclic mode only.

Notice

FB 2 is called twice in FB 92. These blocks are called with a multi-instance DB in FB92.

Declaration

```
FUNCTION_BLOCK FB 92

VAR_INPUT
Req :        BOOL;
GetWkz:      BOOL;
MagNr :      INT;
LocNr :      INT;

END_VAR

VAR_OUTPUT
NDR:         BOOL;
Error:       BOOL;
State :      WORD;

END_VAR
```

Explanations of formal parameters

The table below lists all the formal parameters for block GET_LOC.

Signal	Type	Type	Value range	Remarks
NDR	E	Bool		Start Read state
GetWkz	E	Bool		0 signal: Read magazine location data 1 signal: Read magazine location and tool data
MagNr	E	Int	1..	Magazine number
LocNr	E	Int	1..	Location number
Done	O	Bool		Operation successfully executed.
Error	O	Bool		Task was acknowledged negatively or could not be executed. Fault no. stored in State.
State	O	Word		See error messages

Furthermore, the Read job can be influenced with the following signals in the instance DB of the FB92:

TNr_write = 1: The T number of the tool for tool data is in TNr (DIW28). MagNr/ LocNo are not interpreted.
Only tool data is read.

Error identifiers

If it was not possible to execute a job, the failure is indicated by “logic 1” on status parameter error. The error cause is coded at the block output State:

State	Meaning	Note
1	Error on reading magazine location data (FB 2).	The error ID of FB 2 is stored as the variable StateFB2_TNrGesp. in the instance DB.
2	The logical T number of the magazine location is zero.	Check whether a tool is located in the specified magazine location.
3	Error on reading tool data (FB 2).	The fault detection of the FB2 is stored as the variable StateFB2_WZGesp in the instance DB.
6	Invalid step number	Internal error in FB.
7	Error while reading variable numMagPlaceParams with FB 2.	Restart required.

Data interface

DB instance	
Byte	Description of the data read
DIW 28	Logical T number
DIW 30	Location state
DIW 32	Size to left in half locations
DIW 34	Size to right in half locations
DIW 36	Size upwards in half locations
DIW 38	Size downwards in half locations
DIW 40	Magazine location type
DIW 42	Tool status

Pulse diagram

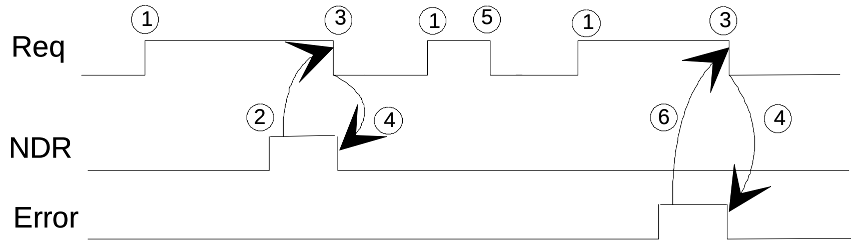


Bild 11-2 Timing diagram for FB 92

1. Activation of function
2. Positive acknowledgment: Receive new data
3. Reset function activation after receipt of acknowledgment
4. Signal change by means of FB
5. If function activation signal is reset before receipt of acknowledgement, the output signals are not updated; not relevant once the function is running
6. Negative acknowledgment: Error occurred. Error code in the output parameter State

Call example

```
U          DB21.DBX 204.1;     // M81 signal
S          M 160.0;            // Start Read states
CAll FB92,DB 92(
Req :      M 160.0,            // Start Read states
GetWkz:    true,               // Read magazine location and tool data
MagNr: 9998,                   Magazine no.= temporary storage
LocNr:2,                       // Magazine loc. 2 = gripper
NDR :      M 160.1,            // Task executed
Error:     M 160.2,            // Reading error
State :    MW 162);            // Error number

U          M 160.1;            // Data read
O          M 160.2;            // Reading error
R          M 160.0;            // Start empty location search
U          M 160.2;
S          M 160.7;            // Error in reading tool data
```

Blocks to be loaded

FB92, FB2, DB92, DB119

11.4 FB 93: PUT_LOC write magazine location and tool data

Description of functions

The magazine location status of a magazine location and the tool status of a tool can be written with FB PUT_LOC.
Every FB 93 call must be assigned a separate instance DB from the user area.
Depending on the signal at input PutWkz, calling FB 93 writes the data on a positive edge change at control input Req. If input PutWkz carries a 1 signal the tool status is written, if PutWkz = 0, the magazine location status is written.
The magazine location is transferred to the FB via input parameters MagNr and LocNr. Successful execution of the job is displayed by means of a logic "1" in status parameter Done. Any errors that have occurred are output via Error and State.
The status data are entered in the instance DB. A detailed description of the data is to be found in the Description Lists in Chapter 4, Variables, and in the Description of Functions Tool Management in the Section Programming.
The Write process is a routine executed over several PLC cycles. The block can be called up in cyclic mode only.
FB 2 is called once and FB 3 twice in FB 93. These blocks are called with a multi-instance DB in FB 92.

Notice

FB 93 can execute the read operations only if basic program parameter NCKomm has been set to "1" (in OB100: FB 1, DB 7).

Declaration

```
FUNCTION_BLOCK FB 93

VAR_INPUT

Req :        BOOL;
PutWkz:      BOOL;
MagNr :      INT;
LocNr :      INT;
END_VAR

VAR_OUTPUT

Done:        BOOL;
Error:       BOOL;
State :      WORD;

END_VAR
```

Explanations of formal parameters

The table below lists all the formal parameters for block PUT_LOC.

Signal	Type	Type	Value range	Remarks
Done	E	Bool		Start write status
PutWkz	E	Bool		0 signal: Write magazine location state 1 signal: Write tool status
MagNr	E	Int	1..	Magazine number
LocNr	E	Int	1..	Location number
Done	O	Bool		Operation successfully executed.
Error	O	Bool		Task was acknowledged negatively or could not be executed. Fault no. stored in State.
State	O	Word		See error messages

Furthermore, the Write job can be influenced with the following signals in the instance DB of the FB 93:

TNr_write = 1: The T number of the tool for Write tool data is in T-Nr(DIW32). MagNr/LocNo are not evaluated

Error identifiers

If it was not possible to execute a job, the failure is indicated by "logic 1" on status parameter error. The error cause is coded at the block output State:

State	Meaning	Note
1	Error on reading magazine location data (FB 2).	The error ID of FB 2 is stored as the variable StateFB2_TNrGesp. in the instance DB.
2	The logical T number of the magazine location is zero.	Check whether a tool is located in the specified magazine location.
3	Error while writing magazine location data (FB 3).	The fault detection of the FB3 is stored as the variable StateFB3_LocGesp in the instance DB.
4	Tool data writing error (FB 3).	The fault detection of the FB3 is stored as the variable StateFB3_WZGesp in the instance DB.
6	Invalid step number	Internal error in FB.
7	Error while reading variable numMagPlaceParams with FB 2.	Restart required.

Data interface

DB instance	User Interface
Byte	Description of the data
DIW 32	Logical T number (read by the FB)
DIW 34	Location state (read by FB)
DIW 36	Location state (data to tool management, block TP, parameter P5)
DIW 38	Tool status (data to tool management, block TD, toolState)

Pulse diagram

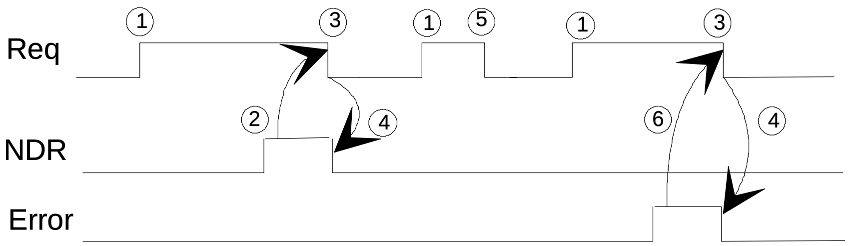


Bild 11-3 Timing diagram FB 93

1. Activation of function
2. Positive acknowledgment: variables have been written
3. Reset function activation after receipt of acknowledgment
4. Signal change by means of FB
5. If function activation signal is reset before receipt of acknowledgement, the output signals are not updated; not relevant once the function is running
6. Negative acknowledgment: Error occurred. Error code in the output parameter State

Call example

```
U          DB21.DBX 204.2;     // M82 signal
S          M 164.0;            // Start Read states
CAll FB93,DB 93(
Req :      M 164.0,            // Start Read states
GetWkz:      true,             // Read magazine location and tool data
MagNr: 1,                      // Magazine no.= Magazine 1
LocNr:10,                      // Magazine location 10
Done:      M 164.1,            // Task executed
Error:     M 164.2,            // Reading error
State :    MW 166);            // Error number

U          M 164.1;            // Data read
O          M 164.2;            // Reading error
R          M 164.0;            // Start empty location search
U          M 164.2;
S          M 164.7;            // Error in reading tool data.
```

Blocks to be loaded

FB93, FB2, DB93, DB119

Notes

Abbreviations

A

ASUB	Asynchronous Subroutine
BLS	Block search
C1 .. C4	Channel 1 to Channel 4
CC	Compile cycle or OEM or user area
CUTOM	Cutter Radius Compensation: Tool radius compensation
DB	Data Block in the PLC
DBB	Data Block Byte in the PLC
DBW	Data Block Word in the PLC
DBX	Data Block Bit in the PLC
DDE	Dynamic Data Exchange:
DW	Data Word
ENC	Encoder: Actual value encoder
EPROM	Erasable Programmable Read Only Memory
FB	Function Block
FC	Function Call: Function block in the PLC
FM–NC	Function Module Numerical Control
GUD	Global User Data: Global user data
HEX	Abbreviation for Hexadecimal Number
HMI	Human Machine Interface
IBN	Installation and Start-up
INC	Increment: Increment
INI	Initializing Data: Initializing data
ISO Code	Special punchtape code, number of punched holes per character always even
K Bus	Communications Bus
MCS	Machine Coordinate System (Machine)
MD	Machine Data
MDA	Manual Data Automatic: Manual input
MMC	Man Machine Communication, see HMI

MSD	Main Spindle Drive
NC	Numerical Control: Numerical Control
NCK	Numerical Control Kernel: NC kernel with block preparation, travel range, etc.
OA	Open Architecture
OB	Organization Block in the PLC
OEM	Original Equipment Manufacturer: manufacturer whose products are marketed under a different name.
OP	Operation Panel: Operating setup
OPI	Operator Panel Interface
PI	Program Invocation: Programming Instance
PLC	Programmable Logic Controller: Programmable logic control
TCA	ToolChangeAbsolute
TCI	ToolChangeIntermediateLocation
TL	Tool
TLC	Tool Length Compensation
TM	Tool Management
TMBF	**T**ool **M**anagement **B**asic **F**unction
TMFD	**T**ool **M**anagment **F**lat **D** numbers
TMMG	**T**ool **M**anagement **M**a**g**azines
TMMO	**T**ool **M**anagement **Mo**nitoring function
TO	Tool Offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
TOOLGNT	TOOLGroupNumber OfTools
TOOLGT	TOOLGroupToolNumber
TRC	Tool Radius Compensation
USEKT	UseKindOfTool
V.24	Serial Interface RS–232 (definition of the interchange circuit between DTE and DCE)
VDI	Virtual Device Interface: virtual interface
WCS	Workpiece Coordinate System (Work)

Terminology

B

Important terms are listed in alphabetical order. The symbol > precedes terms which are explained under a separate entry in this list.

A

Access authorization

Programs and other data are protected internally by a system of access rights based on seven levels:

- Three password levels for system manufacturers, machine manufacturers and users and

Four keyswitch settings which can be evaluated via the PLC (depending on the keyswitch hardware).

Alarms

All messages and alarms are displayed on the operator panel in plaintext with date and time as well as the appropriate symbol for the reset criterion. Alarms and messages are displayed separately.

1. Alarms and messages in the part program:
   Alarms and messages can be displayed in plain text directly from the part program.
2. Alarms and messages from PLC
   Alarms and messages for the machine can be displayed in plain text from the PLC program. No additional function block packages are required for this purpose.

Approach machine fixed-point

Approach motion towards one of the predefined –> fixed machine points.

Archiving

Exporting files and/or directories to an external storage device.

Asynchronous subroutine

A part program that can be started asynchronously (or independently) of the current program status by means of an interrupt signal (e.g. “High-speed NC input” signal) (SW package 4 and higher).

Auxiliary functions,

Auxiliary functions can be used to transfer –> parameters to the –> PLC in –> part programs, where they trigger reactions which are defined by machine manufacturers.

Axes

- In accordance with their functional scope, the CNC axes are subdivided into:
- Axes: interpolating path axes

Auxiliary axes: non-interpolating feed and positioning axes with an axis-specific feed rate. Auxiliary axes are not involved in the actual machining, and include for example tool feeders and tool magazines.

Axis identifier

Axes are referred to in accordance with DIN 66217 (for a right-handed rectangular –> coordinate system) with the letters X,Y, Z.
Rotary axes rotating around X,Y, Z > are referred to as A, B, C.
Additional axes, which are parallel to those specified, can be identified with other letters.

Axis/spindle replacement

An axis/a spindle is permanently assigned to a specific channel via machine data. Using program commands it is possible to release an axis/spindle and assign it to another channel.

B

Backup

Copies of the contents of storage medium (hard disk) are stored to an external memory device for the purpose of backing up and/or archiving data.

Basic coordinate system

Cartesian coordinate system which is mapped by transformation onto the machine coordinate system.
In the –> part program, the programmer uses the axis names of the basic coordinate system. The basic coordinate system exists in parallel to the –> machine coordinate system when no –> transformation is active. The difference between the systems relates to the axis identifiers.

Block

“Block” is the term given to any files required for creating and processing programs.

Block

A section of a –> part program terminated with a line feed. A distinction is made between –> main blocks and –> subblocks.

Block search

The block search function allows any point in the part program to be selected, at which machining must start or be continued. The function is provided for the purpose of testing part programs or continuing machining after a program abort.

Booting

Loading the system program after power on.

C

Channel

A channel is characterized by being able to run independently of other channels or a –> part program. A channel exclusively controls the axes and spindles assigned to it. Part programs runs of various channels can be coordinated by –> synchronization.

Channel structure

The channel structure makes it possible to process the –> programss of individual channels simultaneously and asynchronously.

Compensation memory

Data range in the control, in which the tool offset data are stored.

Contour monitoring

The following error is monitored within a definable tolerance band as a measure of contour accuracy. Overloading of the drive, for example, may result in an unacceptably large following error. In such cases, an alarm is output and the axes are stopped.

Cycle

Subroutine for executing a repetitive machining process on the workpiece.

Cycle support

The available cycles are listed in the “Cycle support” menu in the “Program” operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plain text.

D

D number

Number for the tool offset memory.

Data block

1. A data unit on the –> PLC which can be accessed by –> HIGHSTEP programs.
2. –> NC data unit: Data modules contain data definitions for global user data. These data can be initialized directly when they are defined.

Data word

A data unit, two bytes in size, within a –> PLC data block.

Dimensions specification, metric and inches

Position and pitch values can be programmed in inches in the machining program. The control is set to a basic system regardless of the programmable dimensional specification (G70/G71).

E

Editor

The editor makes it possible to create, edit, extend, join, and import programs/ texts/program blocks.

F

File type

Possible types of files, e.g. part programs, zero offsets, R parameters, etc.

Fixed machine point

A point defined uniquely by the machine tool, such as the reference point.

Fixed-point approach

Machine tools can approach fixed points such as a tool change point, loading point, pallet change point, etc. in a defined way. The coordinates of these points are stored in the control. Where possible, the control traverses these axes in –>rapid traverse.

Frame

A frame is an arithmetic rule that transforms one Cartesian coordinate system into another Cartesian coordinate system. A frame contains the components work offset, rotation, scaling, mirroring.

I

Identifier

Words in compliance with DIN 66025 are supplemented by identifiers (names) for variables (arithmetic variables, system variables, user variables), for subroutines, for keywords and for words with several address letters. These supplements have the same meaning as the words with respect to block format. Identifiers must be unique. It is not permissible to use the same identifier for different objects.

Increment

Travel path length specification based on number of increments. The number of increments can be stored as a –> setting data or selected with keys labeled with 10, 100, 1000, 10 000.

K

Keylock switch

The keyswitch is the operating mode switch of the CPU. The keylock switch is operated by a removable key.
The keylock switch on the –> machine control panel has 4 settings, to which functions are assigned by the operating system of the control. Further, the keylock switch has three differently colored keys, which can be removed in the specified positions.

L

Language

The user guidance display texts and the system messages are available in five system languages:
German, English, French, Italian, and Spanish.
The user can select two of the listed languages at a time in the control (Startup operating area).

M

Machine axes

Physically existent axes on the machine tool.

Machine control panel

An operator panel on a machine tool with operating elements such as keys, rotary switches, etc., and simple indicators such as LEDs. It is used to directly influence the machine tool via the PLC.

Machine coordinate system

A coordinate system, which is related to the axes of the machine tool.

Machine zero

A fixed point on the machine tool, which can be referenced by all (derived) measuring systems.

Machining channel

Via a channel structure, parallel sequences of movements, such as positioning a loading gantry during machining, can shorten unproductive times. Here, a CNC channel must be regarded as a separate CNC control system with decoding, block preparation and interpolation.

Macro techniques

Individual instructions in the programming language can be linked to create one instruction. This condensed instruction sequence is called by a user-defined name in the CNC program and the macro command executed in accordance with the individual instructions.

Magazine

The following categories of magazine are utilized in the tool management system:

- Real magazine
  Actual magazine for storing tools, the NCK is capable of managing several real magazines.
- Internal magazine
  All other positions in which a tool may be located are handled logically in the NCK as a magazine (or magazine location). There are only two types of internal magazines: the load magazine and the buffer magazine.
- Virtual magazine
  This term is used in the MMC environment to refer to all the real and internal magazines of one TO unit.
- Active magazine
  Magazine which is linked to a spindle and from which a tool change can be executed.
- Background magazine
  A magazine which is linked to a previous magazine via system parameter $TC_MAP5. Generally speaking, tool changes involve the relocation of tools.

Main block

A block prefixed by ":" containing all the parameters required to start execution of a –> part program.

Main program

–> Part program identified by a number or name in which further main programs, subroutines or –> cycles can be called.

Main run

The part program blocks which have been decoded and prepared in the "preprocessing" run are executed in the "main run".

MDI

Control operating mode: Manual Data Input: In the MDI mode, individual program blocks or block sequences with no reference to a main program or subprogram can be input and executed immediately afterwards through actuation of the NC start key.

Messages

All messages programmed in the part program and all –> alarms recognized by the system are displayed on the operator panel in plain text. Alarms and messages are displayed separately.

Mirroring

Mirroring reverses the signs of the coordinate values of a contour, with respect to an axis. It is possible to mirror with respect to more than one axis at a time.

N

NC

Numerical Control: It incorporates all the components of the machine tool control system: –> NCK, –> PLC, –> MMC, –> COM.
Note: CNC (computerized numerical control) would be more appropriate for the SINUMERIK 840D or FM–NC controls: MARS and Merkur controls.

NCK

Numerical Control Kernel: Component of the NC control which executes –> part programs and essentially coordinates the movements on the machine tool.

NRK

Numeric Robotic Kernel (operating system of the –> NCK)

O

OEM

The scope for implementing individual solutions (OEM applications) for the SINUMERIK 840D has been provided for machine manufacturers, who wish to create their own operator interface or integrate process-oriented functions in the control.

Operating mode

An operating concept on a SINUMERIK control. The operating modes –> JOG, –> MDI and –> Automatic are defined.

Oriented spindle stop

Stops the workpiece spindle with a specified orientation angle, e.g., to perform an additional machining operation at a specific position.

Oriented tool retraction

RETTOOL: If machining is interrupted (e.g., when a tool breaks), a program command can be used to retract the tool in a user-specified orientation by a defined distance.

Override

Manual or programmable control feature, which enables the user to override programmed feedrates or speeds in order to adapt them to a specific workpiece or material.

P

Part program

A sequence of instructions to the NC control which combine to produce a specific –> workpiece by performing certain machining operations on a given –> blank.

PLC

Programmable Logic Control: –> Programmable logic control. Component of the –> NC control: Programmable controller for processing the control logic of the machine tool.

PLC program memory

- SINUMERIK 840D The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory. The PLC user memory can be expanded up to 128 KB with memory expansions.
- SINUMERIK 810D: The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory on the CPU 314. The user memory in the basic configuration of the S7–CPU314 is 64 KB in size and can be optionally expanded to 128 KB.

R

R parameters

Calculation parameter. The programmer of the –> part program can assign or request the values of the R parameter as required.

Reference point

Point on the machine tool with which the measuring system of the –> machine axes is referenced.

Reference point approach

If the position measuring system used is not an absolute-value encoder, then a reference point approach operation is required to ensure that the actual values supplied by the measuring system are in accordance with the machine coordinate values.

Replacement tool

A tool group generally contains several tools. For tool change purposes, only the identifier is specified in the part program. The tool with the “active” status is generally selected as the new tool. But if this is disabled, then one of the other twin tools, i.e. the replacement tool, is selected instead. –> Replacement tool

Replacement tool, tool group

Replacement tools have the same identifier and only differ in the duplo number. The replacement tools assigned to one identifier are also referred to as a tool group.

REPOS

1. Repositioning on the contour using operator input
   The REPOS function can use the direction keys to reposition at the point of interruption.
2. Repositioning on the contour by program
   The program commands provide various approach strategies: Approach point of interruption, approach start of block, approach end of block, approach a point on the path between start of block and interruption.

S

Safety functions

The control includes continuously active monitoring functions which detect faults in the –> CNC, the programmable controller (–> PLC) and the machine so early that damage to the workpiece, tool or machine rarely occurs. In the event of a fault, the machining operation is interrupted and the drives stopped. The cause of the malfunction is logged and output as an alarm. At the same time, the PLC is notified that a CNC alarm is pending.

Setting data

Data, which communicates the properties of the machine tool to the NC control, as defined by the system software.

Softkey

A key whose name appears on an area of the screen. The selection of keys displayed is adapted dynamically to the operating situation. The freely assignable function keys are assigned defined functions in the software.

Spindles

- Spindle = toolholder
  Toolholder is generally the location for the machining tool. However, the term “spindle” is frequently used in this general context.
- Main spindle = master spindle
  This is the spindle with the number defined by machine data
  MD $MC_SPIND_DEF_MASTER_SPIND. Language command SETMS(n) can be programmed to declare the spindle with number n as the master spindle. A channel has exactly one master spindle.
- Secondary spindle
  This term refers to all spindles that are not the master spindle.

Standard cycles

Standard cycles are provided for machining operations, which are frequently repeated:

- Cycles for drilling/milling applications
- for turning technology

The available cycles are listed in the “Cycle support” menu in the “Program” operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plain text and can be supplied with values.

Subblock

Block prefixed by “N” containing information for a machining step, such as a position parameter.

Subprogram

A sequence of instructions of a –> part program which can be called repetitively with various defining parameters. –> Cycles are a type of subprogram.

Synchronization

Instructions in –> part programs for coordination of the operations in different –> channels at specific machining points.

Synchronized actions

1. Auxiliary function output
   During the workpiece machining, technology functions (–> auxiliary functions) can be issued from the CNC program to the PLC. These auxiliary functions are used for example to control additional equipment for the machine tool, such as quills, grabbers, clamping chucks etc.
2. Fast auxiliary function output
   For switching functions which are time-critical, the confirmation times for the –> auxiliary functions are minimized, and unnecessary stopping points in the machining process can be avoided.

Synchronized axes

Synchronized axes take the same time to traverse their path as the –> geometry axes take for their path.

System variable

A variable which exists although it has not been programmed by the –> part program programmer. It is defined by a data type and the variable name preceded by the character **$**.
See also –> user-defined variable.

T

Tool Nose Radius Compensation

Contour programming assumes that the tool is pointed. Because this is not actually the case in practice, the curvature radius of the tool used must be communicated to the control which then takes it into account. The curvature center is maintained equidistantly around the contour, offset by the curvature radius.

Tool offset

By programming a T function (5 decades, integer) in the block, you can select the tool. Every T number can be assigned up to 12 cutting edges (D addresses). The number of tools to be managed in the control is set at the configuration stage.

Tool radius compensation

In order to be able to program a desired –> workpiece contour directly, the control must traverse a path equidistant to the programmed contour, taking into account the radius of the tool used (G41/G42).

Transformation

Programming in a Cartesian coordinate system, execution in a non-Cartesian coordinate system (e.g., with machine axes as rotary axes).

U

User-defined variable

Users can define variables in the –> part program or data block (global user data) for their own use. A definition contains a data type specification and the variable name. See also –> system variable.

User interface

The user interface (UI) is the display medium for a CNC control in the form of a screen. It is laid out with eight horizontal and eight vertical softkeys.

User memory

All programs and data, such as part programs, subprograms, comments, tool offsets, and work offsets/frames, as well as channel and program user data can be stored in the shared CNC user memory.

User program –> Part program

V

Variable definition

A variable definition includes the specification of a data type and a variable name. The variable names can be used to access the value of the variables.

W

Working memory

The working storage is a Random Access Memory in the –> CPU containing the user program which is accessed by the processor during program processing.

Workpiece

1. Part or workpiece to be made/machined by the machine tool or
2. A directory where programs and other data is stored. Workpieces are stored in another directory.

Workpiece coordinate system

The starting position of the workpiece coordinate system is the –> workpiece origin. In machining operations programmed in the workpiece coordinate system, the dimensions and directions refer to this system.

Workpiece zero

The workpiece origin is the starting point for the –> workpiece coordinate system. It is defined by the distance from the machine zero.

References

C

Documentation

An overview of publications, which is updated monthly and also provides information about the language versions available, can be found on the Internet at:

http://www.siemens.com/motioncontrol

Follow menu items --> “Support” --> “Technical Documentation” --> “Overview of Documentation” or “DOConWEB”.

Notes

Index

Symbols

Numbers

A

B

C

D

J

K

L

M

S

T

U

V

W

To
SIEMENS AG
A&D MC BMS
P.O. Box 3180

D–91050 Erlangen

(Phone: 49 (0) 180 5050 – 222 [Hotline]
Fax: 49 (0) 9131 98 – 63315 [Documentation]
E-mail: motioncontrol.docu@siemens.com

Suggestions

Corrections

For Publication/Manual:

SINUMERIK 840D/840Di/810D
Description of Functions
Tool Management

Manufacturer Documentation

Description of Functions

Order No.:6FC5297–7AC60–0BP0
09.05 Edition

From

Name

Company/Dept.

Street

Zip code: Town:

Phone: /

Fax: /

If you find any errors when reading this publication, please use this form to notify us.
Suggestions for improvement are also welcome.

Suggestions and/or corrections

Overview of SINUMERIK 840D/840Di/810D Documentation (08/2005)

General Documentation

- *SINUMERIK* — Brochure
- *SINUMERIK* 840D / 840Di / 810D — Catalog NC 61 ***)**
- *Saftey Integrated* — Saftey Integrated Application Manual

User Documentation

- *SINUMERIK* 840D / 840Di / 810D — Operator's Guide
  - **HMI Embedded *)**
  - **ShopMill**
  - **ShopTurn**
  - HT6
- *SINUMERIK* 840D sl / 840D / 840Di / 810D — Operator's Guide
  - **HMI Advanced *)**
  - Programming compact
- *SINUMERIK* 840D sl / 840D / 840Di / 810D — **Programming Guide**
  - Fundamentals ***)**
  - Advanced ***)**
  - Programming
  - Lists System Variables
  - ISO Turning/Milling
- *SINUMERIK* 840D sl / 840D / 840Di / 810D — Programming Guide
  - **Cycles**
  - **Measuring Cycles**
- *SINUMERIK* 840D sl / 840D / 840Di / 810D — **Diagnostics Guide *)**
- *SINUMERIK* 840Di — **System Overview**

Manufacturer/Service Documentation

- *SINUMERIK* — **Configuring** (HW) ***)**
  - 840D
  - 810D
- *SINUMERIK* 840D / 840Di / 810D — Equipment Manual **Operator Components *)**
- *SINUMERIK* — **Installation and Start-up Guide *)**
  - 840D
  - 810D
  - HMI
- *SINUMERIK* 840D / 840Di / 810D — Description of Functions
  - **ShopMill**
  - **ShopTurn**
- *SINUMERIK* 840Di — **Installation and Start-up Guide**
- *SINUMERIK* 840Di — **List Manual *)**
  - Part 1
  - Part 2

Manufacturer/Service Documentation

- *SINUMERIK* 840D sl / 840D / 840Di / 810D — **Description of Functions**
  - Basic Machine ***)**
  - Extended Functions
  - Special Functions
  - **Synchronized Actions**
  - **Iso Dialects**
  - **EMC Guidlines**
- *SINUMERIK* 840D / 840Di / 810D — Description of Functions
  - **Drive Functins**
  - **Tool Management**
  - **Hydraulics Module**
  - **Analog Module**
- *SINUMERIK* 840D / 840Di / 810D — Description of Functions
  - **Remote Diagnosis**
  - **@Event**
- *SINUMERIK* 840D — Description of Functions **Saftey Integrated**
- *SINUMERIK* 840D / 810D — Configuring
  - **Operator Interface OP030**
  - **HMI Embedded**
- *SINUMERIK* 840D / 810D — **MCIS**
  - Computer Link
  - Tool Data Inform. Sys.
  - NC Data Management
  - NC Data Transfer
  - Tool Data Communication

Electronic Documentation

- *SINUMERIK SIMODRIVE Motors* — DOCONCD ***)**, DOCONWEB

***)** These documents are a minimum requirement