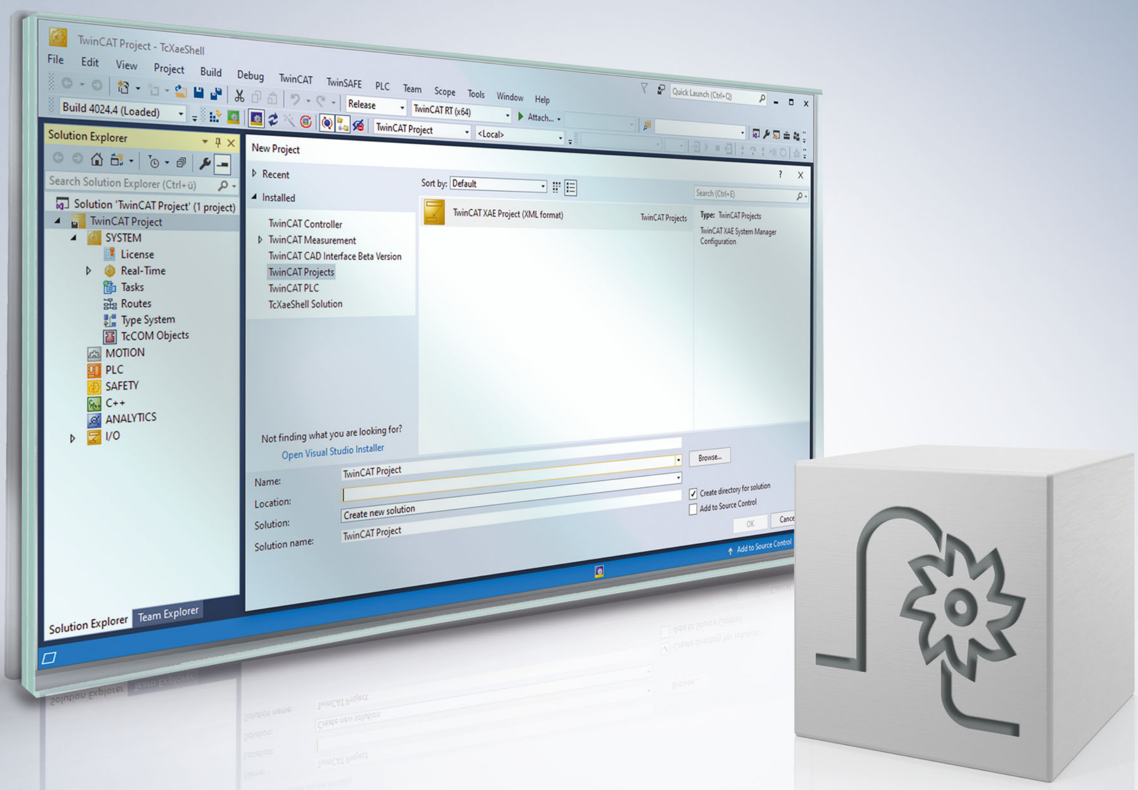
BECKHOFF New Automation Technology

Functional description | EN

TF5200 | TwinCAT 3 CNC

MultiCore



7/16/2024 | Version: 1.05

Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT technology is patent protected, in particular by the following applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

General and safety instructions

Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

Icons in explanatory text

1. Indicates an action.

⇨ Indicates an action statement.

⚠ DANGER

Acute danger to life!

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.

⚠ CAUTION

Personal injury and damage to machines!

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.

NOTICE

Restriction or error

This icon describes restrictions or warns of errors.

Tips and other notes

This icon indicates information to assist in general understanding or to provide additional information.

General example

Example that clarifies the text.

NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.

Specific version information

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

Table of contents

List of figures

1 Overview

Task

The aim here is to split CNC functions requiring intensive calculation times among separate CPU cores with multicore processors.

Possible applications

Individual decoding processes and web interpolators can be split among different CPU cores in a multi-channel machine configuration.

This function is available as of CNC Build V3.1.3077

Parameterisation

Parameterisation takes place in the TwinCAT development environment.

Mandatory note on references to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

For technical reasons, these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

2 Description

Structural description of a multi-channel CNC

A CNC can be designed for several NC channels with additional single-axis interpolators.

The processing of an NC program can be executed in each channel. A group of axes moved together is used for this.

A single-axis interpolator can move a single axis, e.g, by a PLC command.

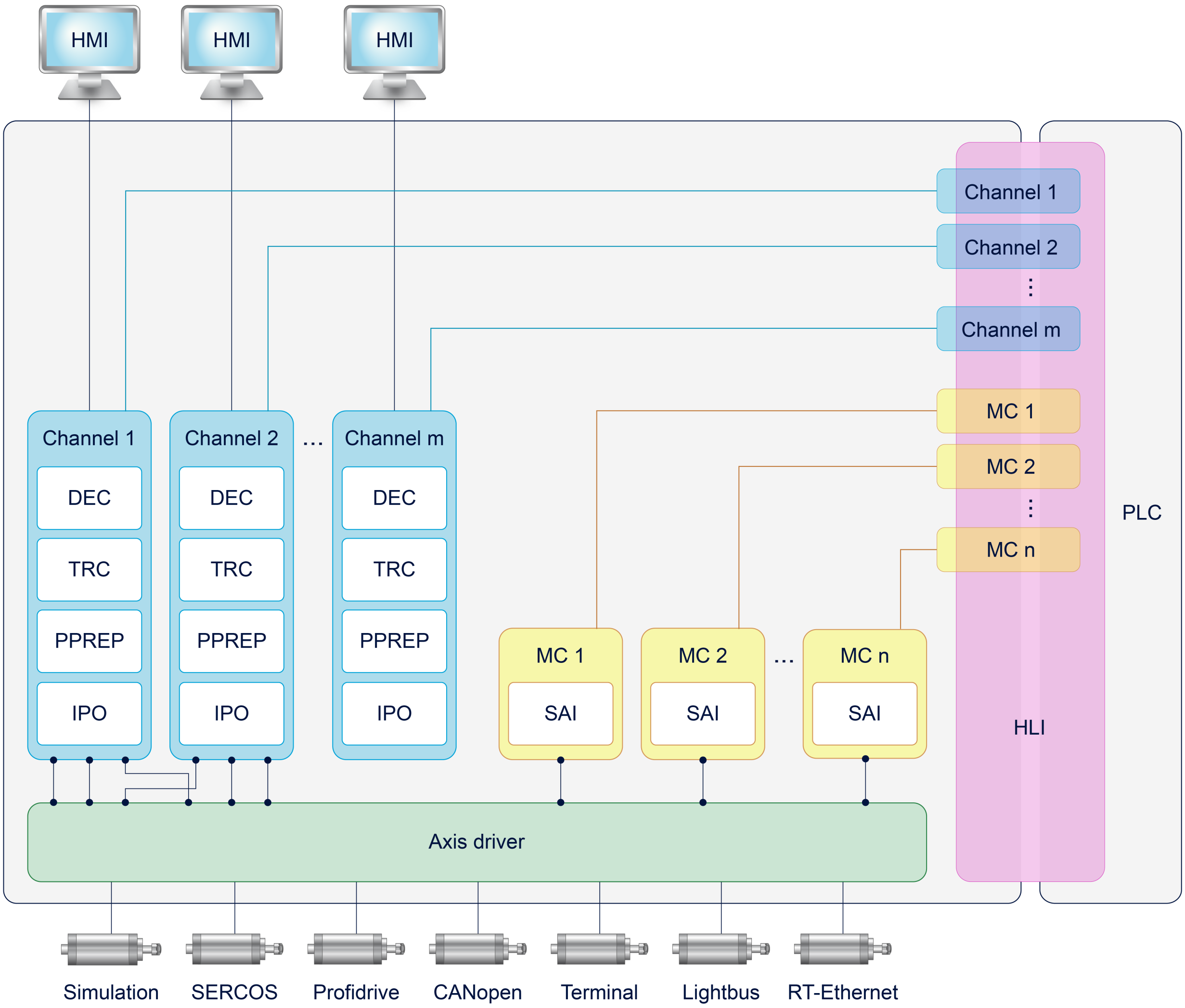


Fig. 1: Structure of a multi-channel CNC

2.1 Standard configuration of CNC tasks

By default, the CNC consists of 3 tasks which can be implemented in a real-time operating system.

- COM task: Driver supplying display values for a user interface.
- SDA task: Consists of the decoder, calculation of tool radius compensation and interpolation preparation (DEC, TRC, PPREP).
- GEO task: Executes the actual cycle-synchronous interpolation, i.e. generates the cyclic axis command values and outputs to the drives.

Depending on the CNC application, the CNC tasks can be prioritised differently and assigned with corresponding cycle times.

Below are examples of several criteria that affect the tasks:

- The COM task affects the transfer rate of objects for the user interface and can be adapted depending on the response time of the display.
- It is recommended to adjust the SDA task for HSC machining which involves a high volume of short motion information. It is advisable to select a short cycle time for the SDA task (decoder) in order to supply the interpolation with a sufficient number of motion blocks and achieve the required programmed velocity (data throughput, block cycle time).
- In general, the GEO task must run synchronously with the bus cycle time so that the drive receives a new command position in each cycle.

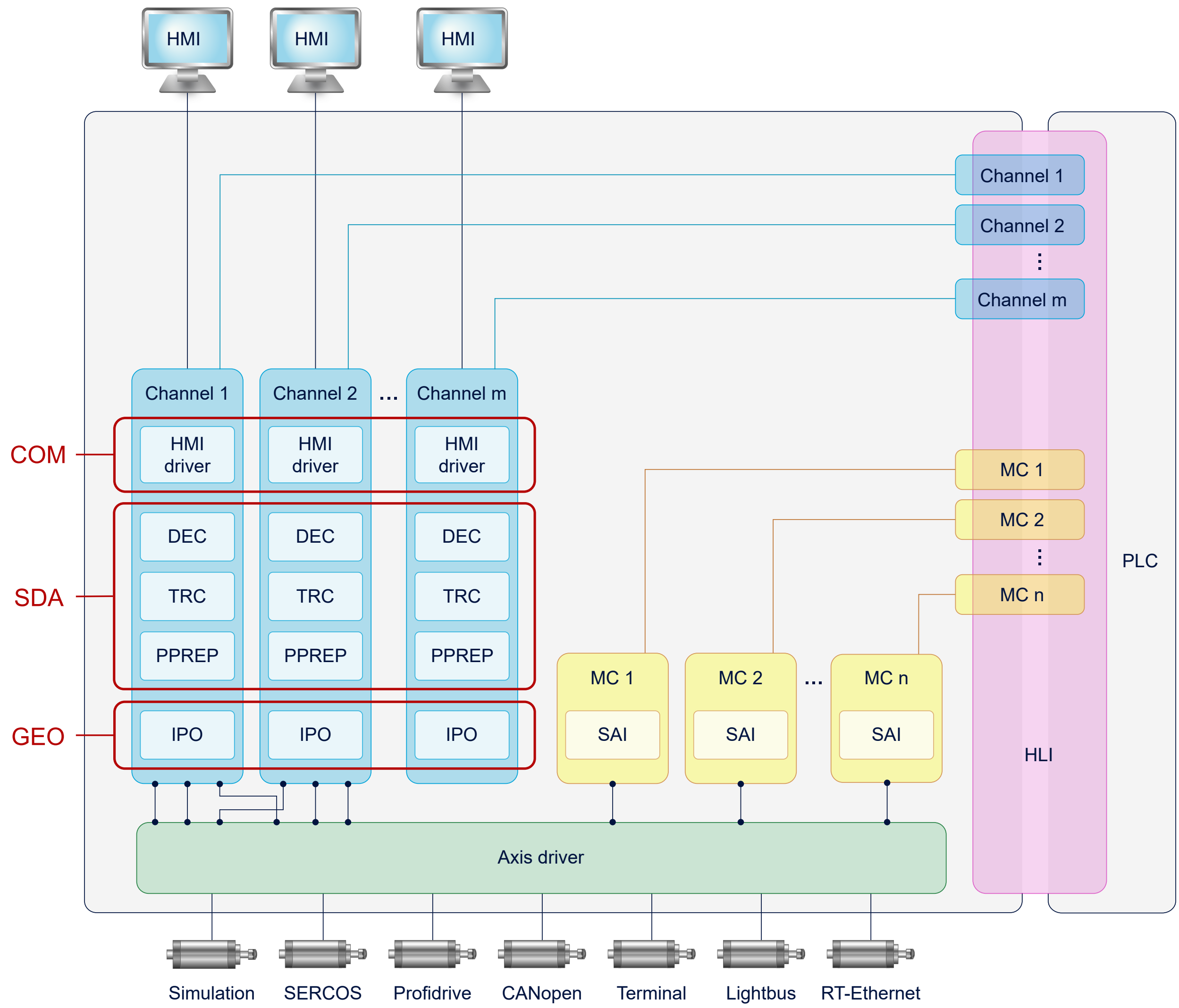


Fig. 2: Standard tasks of a multi-channel CNC

2.2 Configuration of GEO tasks

Standard task distribution of a multi-channel configuration

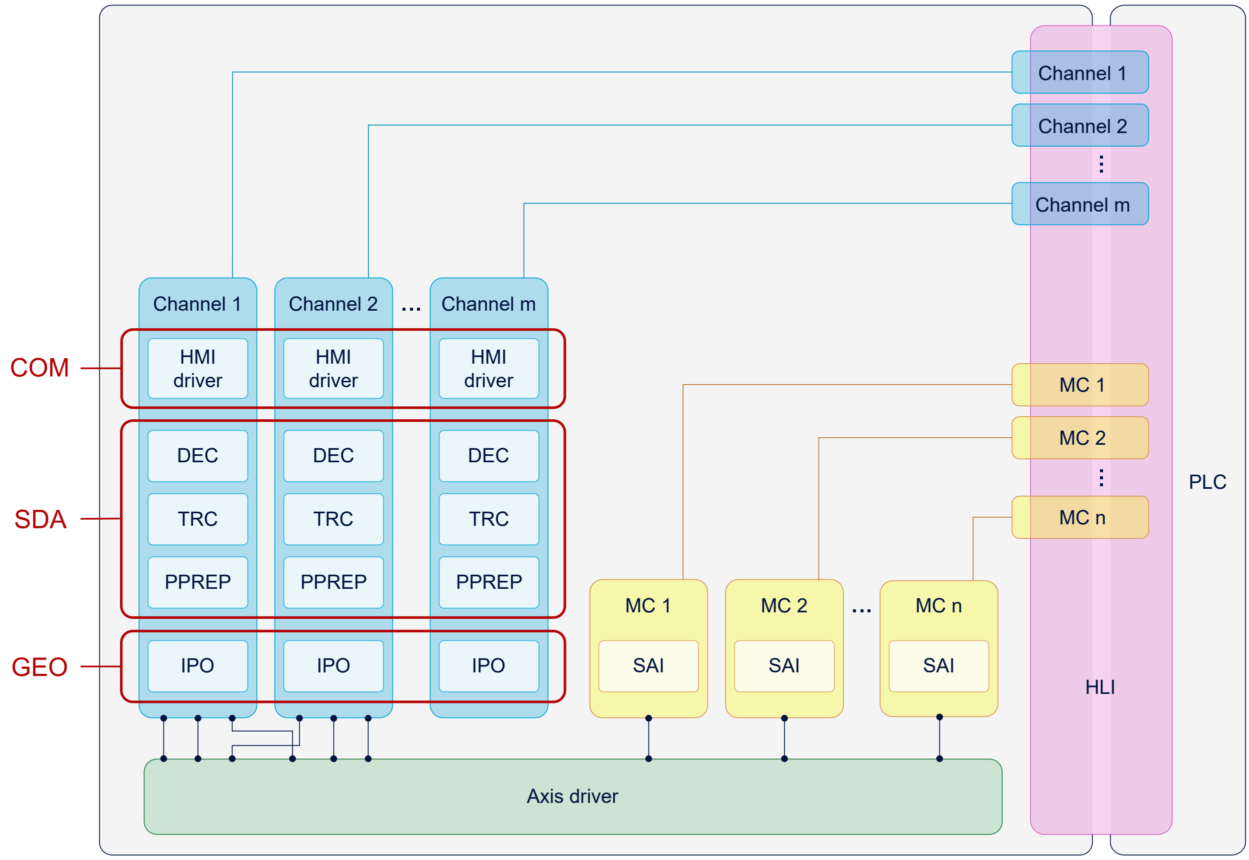


Fig. 3: Initial position (without multicore channel distribution)

The interpolation of each CNC channel can be assigned to a GEO task based on this standard task distribution. Each GEO task can be assigned 1-n channels of the CNC.

In the case below, two additional tasks, GEO 2 and GEO 3, are integrated in a 4-channel configuration:

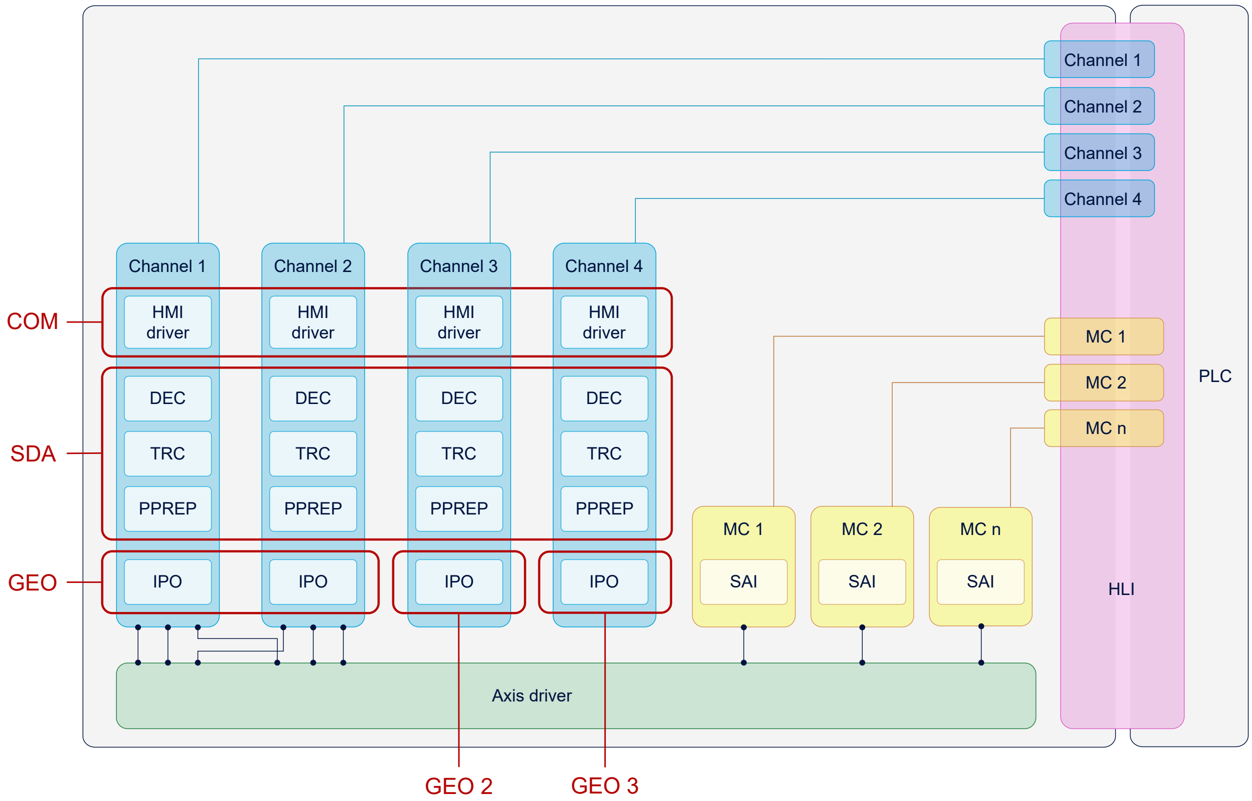


Fig. 4: Distribution of GEO tasks of a 4-channel CNC to multiple cores

2.3 Assignment of CNC task and CNC channel

The individual channel functions (SDA, COM or IPO) are indirectly assigned to a CNC task by defining contexts.

The configuration of contexts is described in the next section.

COM is not split into channels. SAIs are executed in the GEO of the 1st channel.

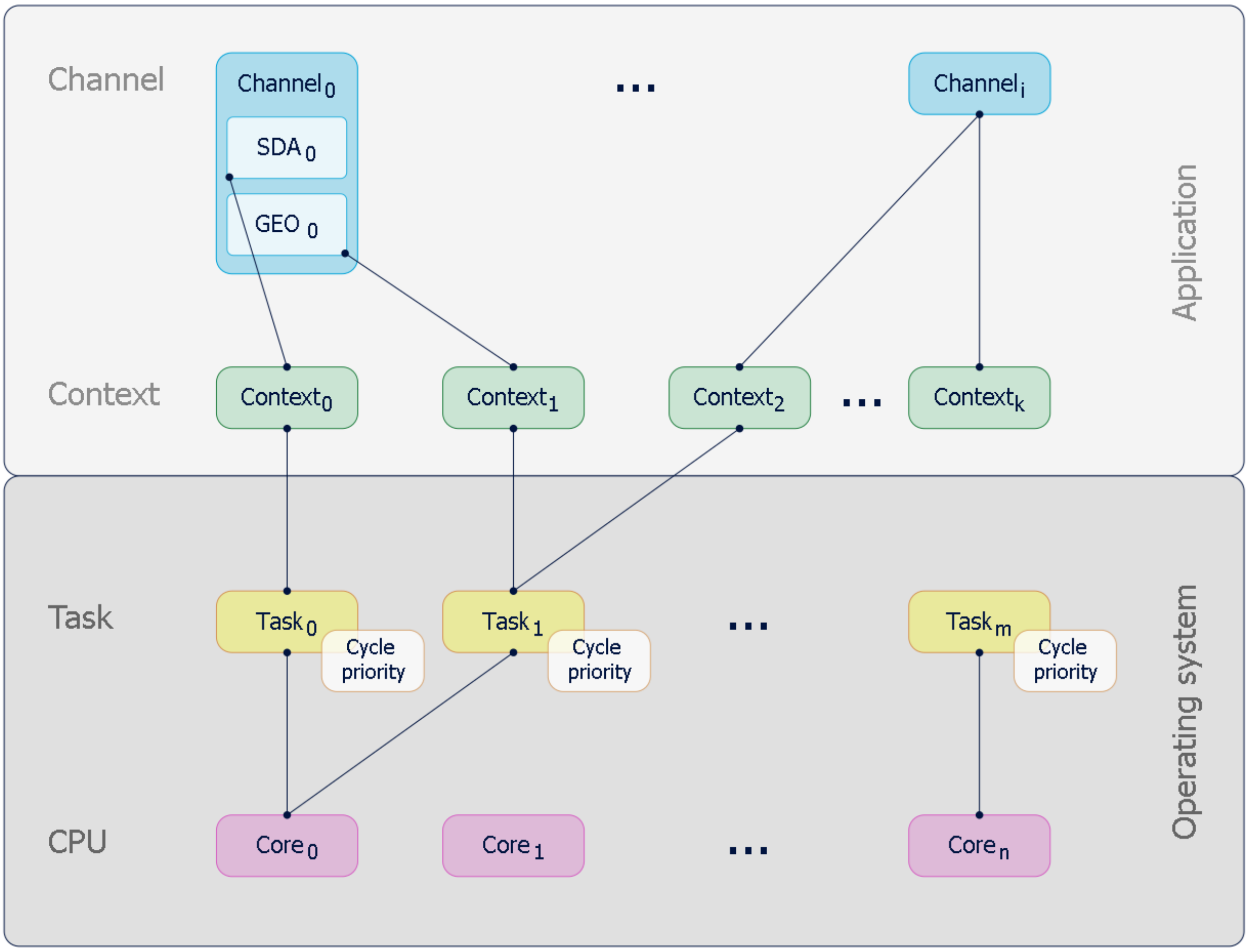


Fig. 5: Assignment by context

3 CNC scheduler

The real-time part of the CNC controller runs in the GEO task. The GEO task performs the following tasks, among others, in each CNC cycle:

Identifier	Task
Input	Read out axis actual values/status/etc. from the fieldbus
Output	Output new axis command values/status/etc. to the fieldbus
IPO	Interpolation, calculation of new axis command values, channel-specific
CHAN	Display, channel-specific

The CNC scheduler defines the order in which these tasks are executed. Defining the suitable order is dependent on the existing hardware (drives, fieldbus,etc.)

The following task orders are available:

- STANDARD
- COMPLETE
- SWITCHED

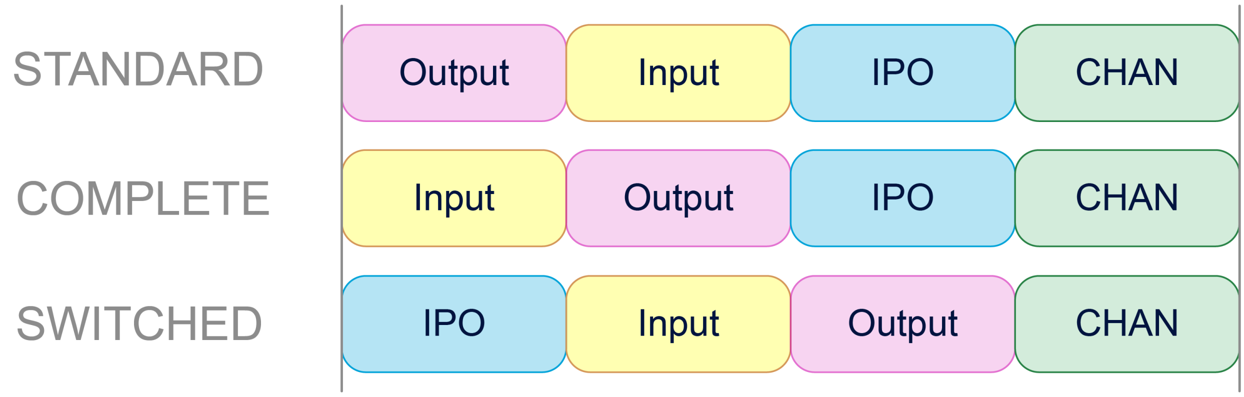


Fig. 6: Overview of task order in the cycle

STANDARD

Especially with conventional +-10V drives, it is important to output command values in cycles that are as constant as possible. To avoid fluctuations, the command values calculated in the previous cycle are output to STANDARD directly at the start of the cycle. Then actual values are read in, compensations are calculated and new command values are calculated by IPO for the next cycle.
This order results in a delay between interpolation and the output of command values.

COMPLETE

If the axis parameter P-AXIS-00276 "field_bus_allows_optimised_schedule" is set for all axes, actual values/ compensations are first processed and only then are the new command values output. This prevents any delay between interpolation and output.

If the parameter P-AXIS-00276 is not set for all axes, the schedule corresponds to the STANDARD case.

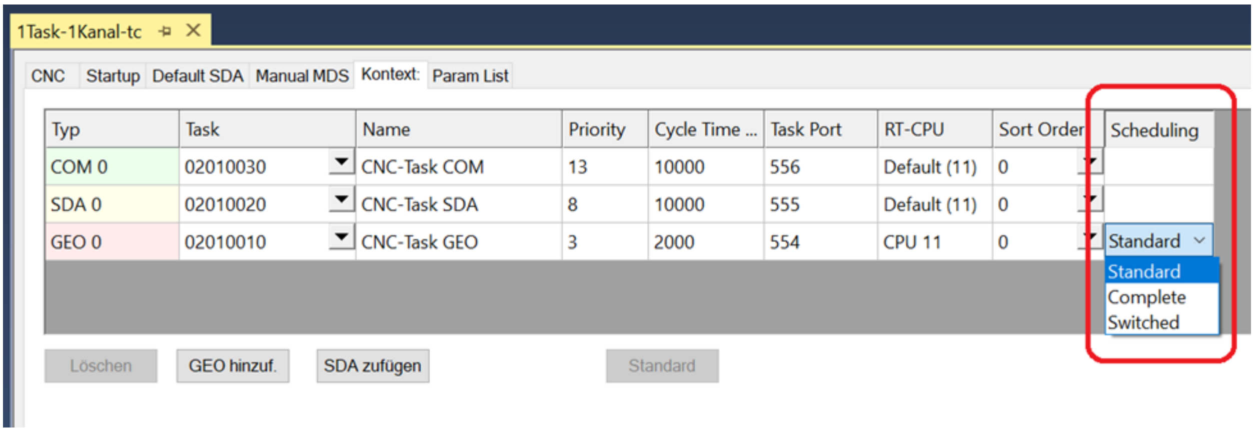
SWITCHED

For digital drives, the order can be further optimised to avoid any delay. In SWITCHED mode, actual values are read in

1. interpolated
2. .
3. Compensations calculated and command values output

Configuration

In TwinCAT, the schedule is set in the "Context" tab of the "CNC" node:



In real-time Linux or Windows Simulator, the P-RTCF-00018 parameter is used for this.

4 Configuration

4.1 Configuration in TwinCAT

The following steps are required for a new CNC task:

1. Determine the available CPU cores
2. Create a new CNC task
3. Set the properties of the new CNC task
4. Generate the context for the new CNC task
5. Link the new CNC task to the context

Determine the available CPU cores

Before splitting tasks, the available cores on the current CPU must be determined. This is achieved using the "Real-time" and "Read from target system" tabs. The cores can be set to isolated / non-isolated. These cores can then be assigned to the tasks.

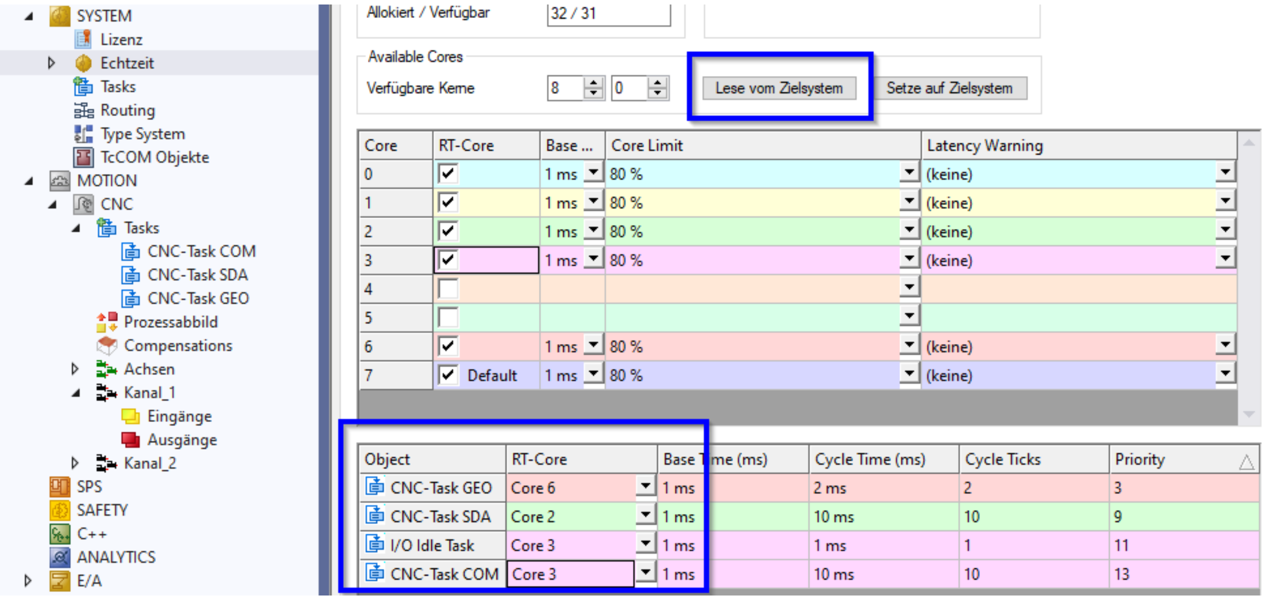


Fig. 7: Determining the available CPU cores

If you specify the available cores incorrectly, TwinCAT may not start properly.

Generate another GEO task for interpolation

By default, a CNC GEO task is created for the CNC. Synchronous tasks must be created to be able to create the interpolation of individual CNC channels on different GEO tasks.

The following sequence describes the procedure:

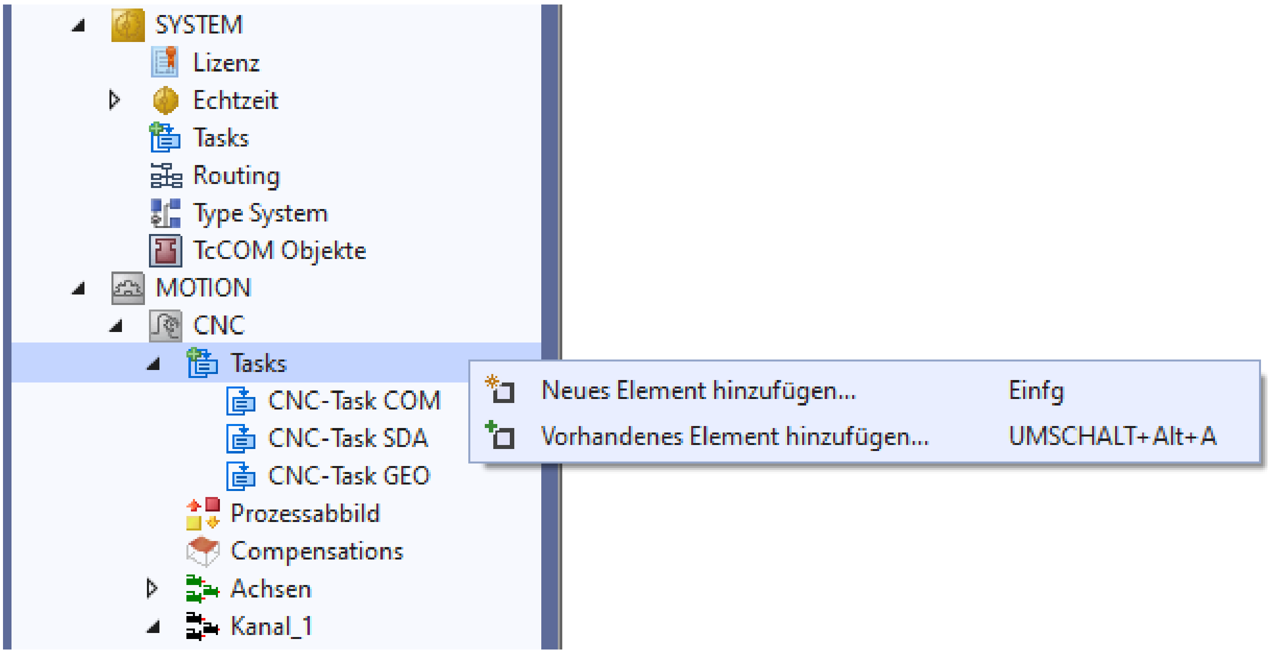


Fig. 8: Generating a new GEO task

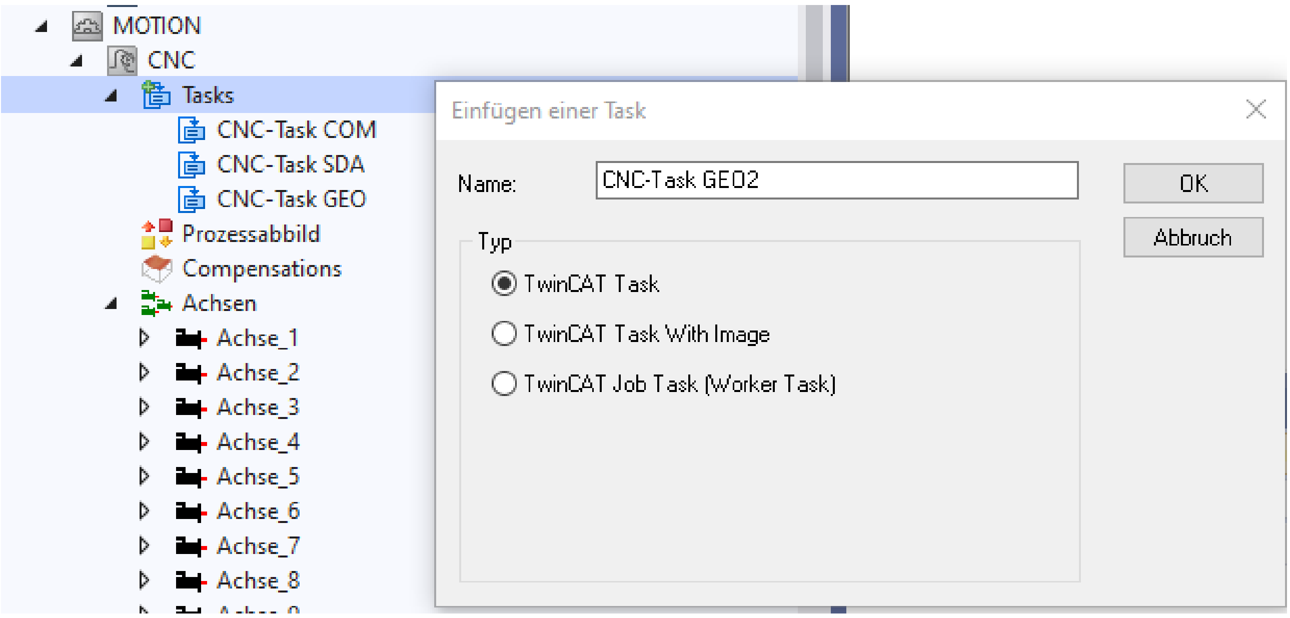


Fig. 9: Create a GEO task with name

The priority, cycle time and port must be modified or checked for each new GEO TASK created.

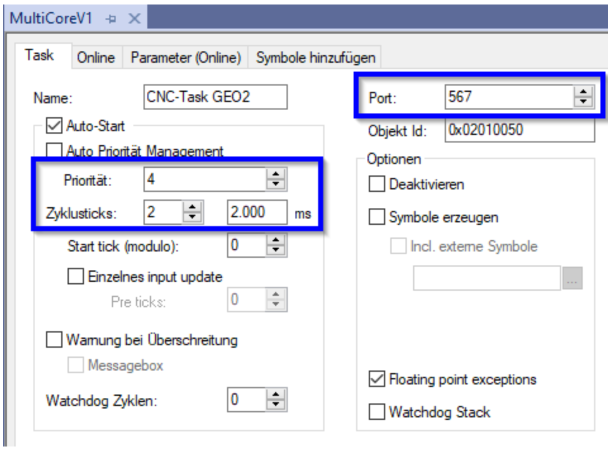


Fig. 10: Settings of task priorities

For the port number it is recommended to use the next number after the port numbers of the existing CNC tasks.

Each GEO task requires a unique priority, whereby the priority of a new task created can be based on the priority of the existing GEO task.

All GEO tasks should be ranked higher in priority than SDA or COM tasks.

The cycle times of all GEO tasks must have the same setting.

Creating the context between CPU core and the new CNC task

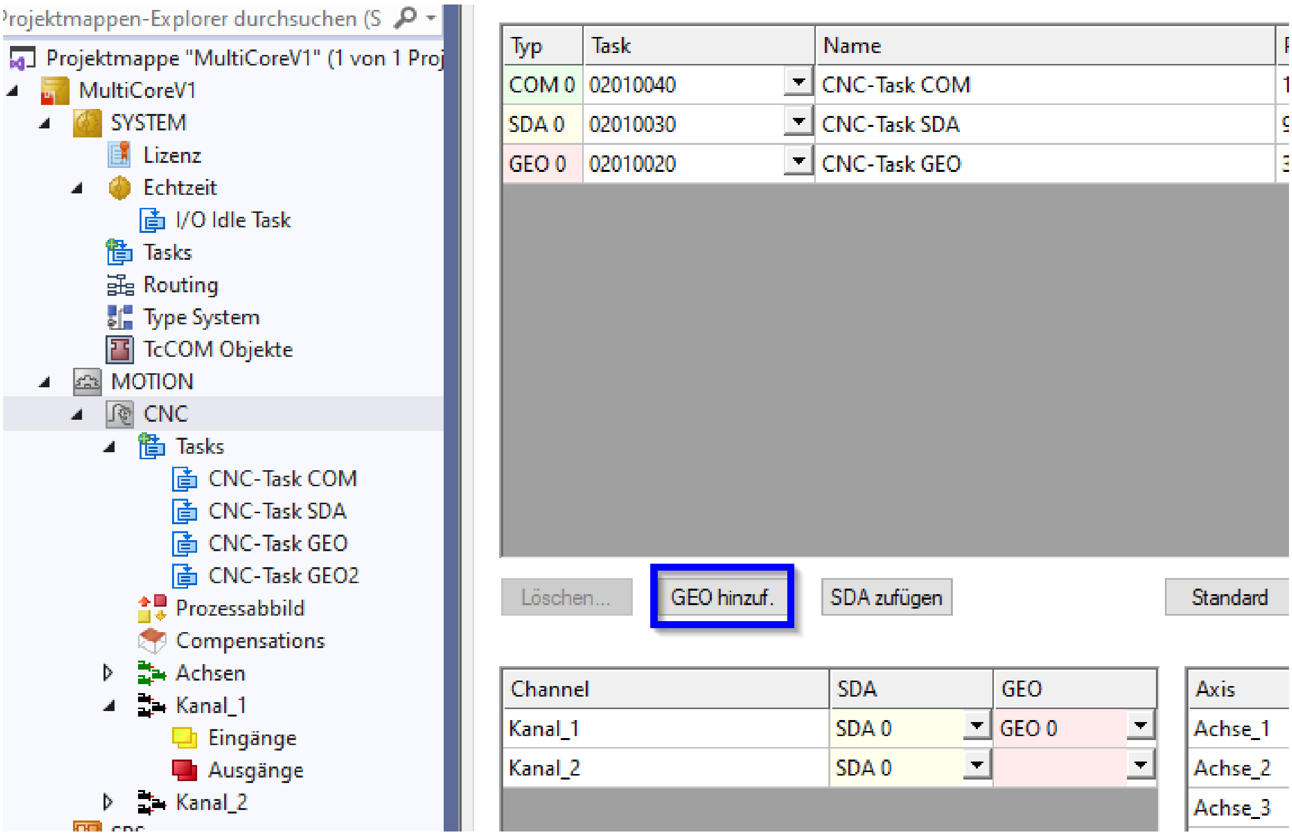


Fig. 11: Create context for new GEO task

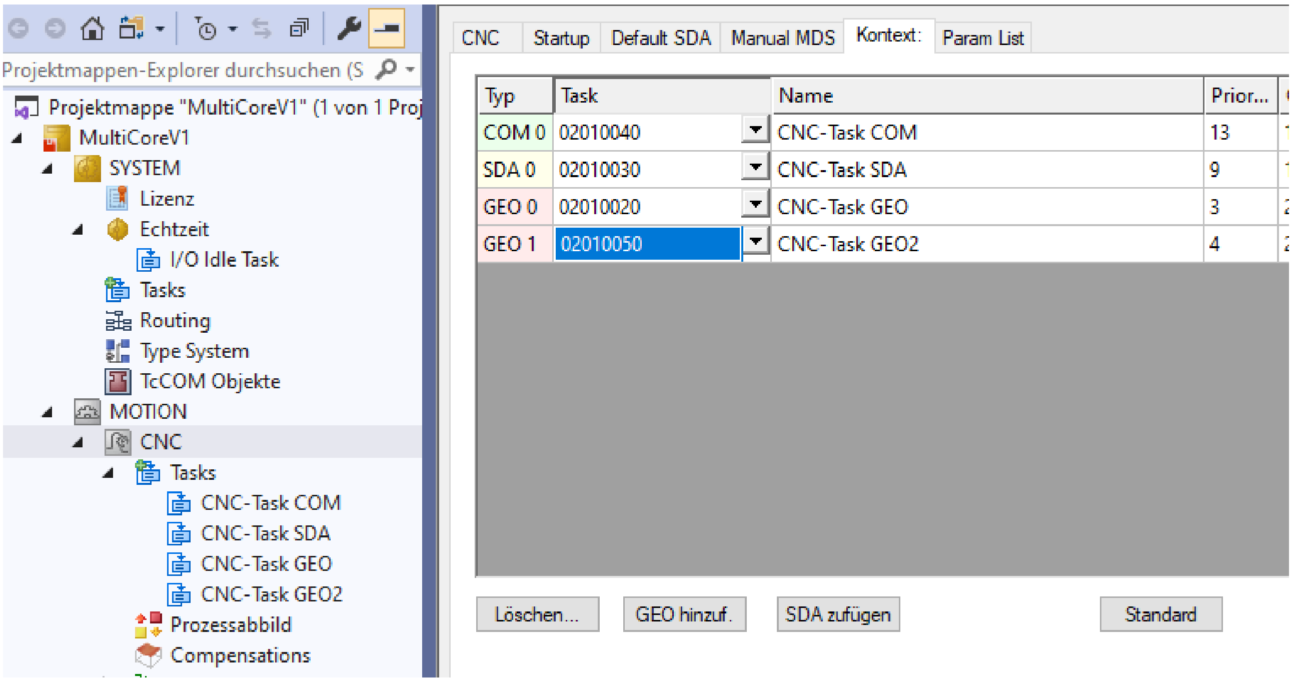


Fig. 12: Assign the new GEO task to the context created

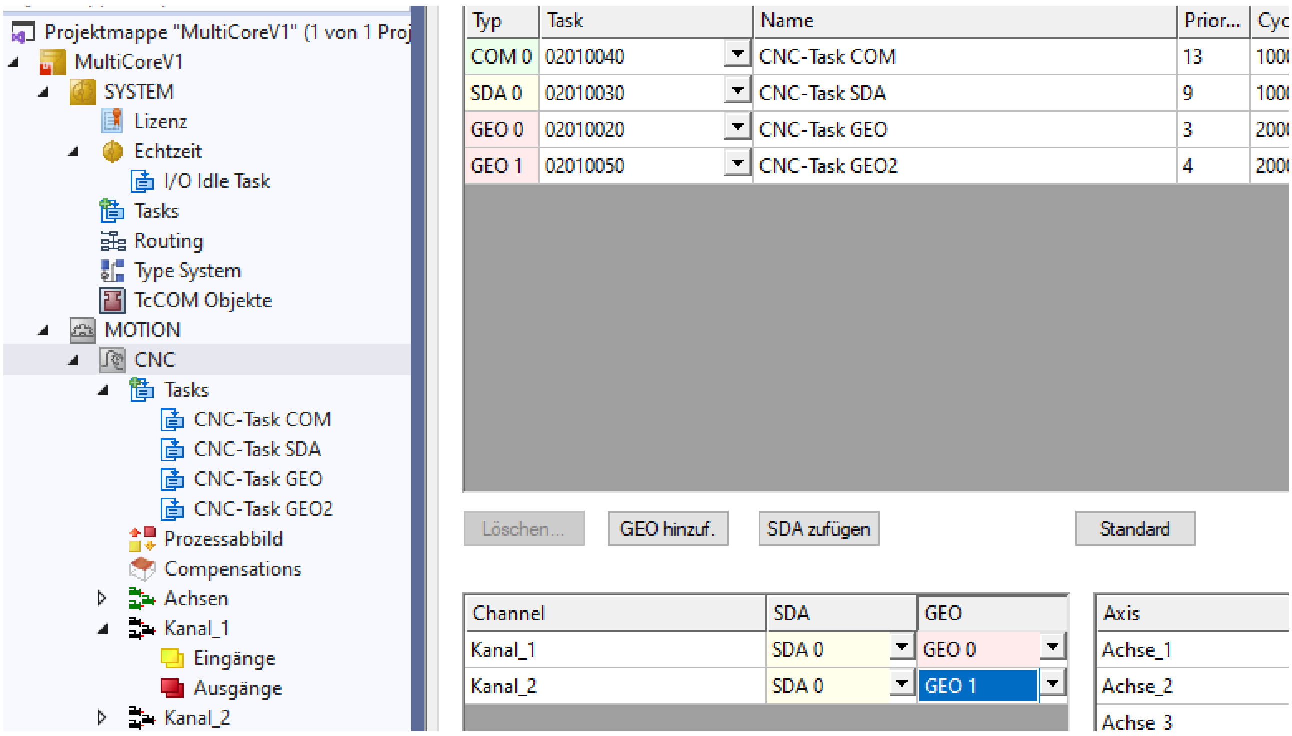
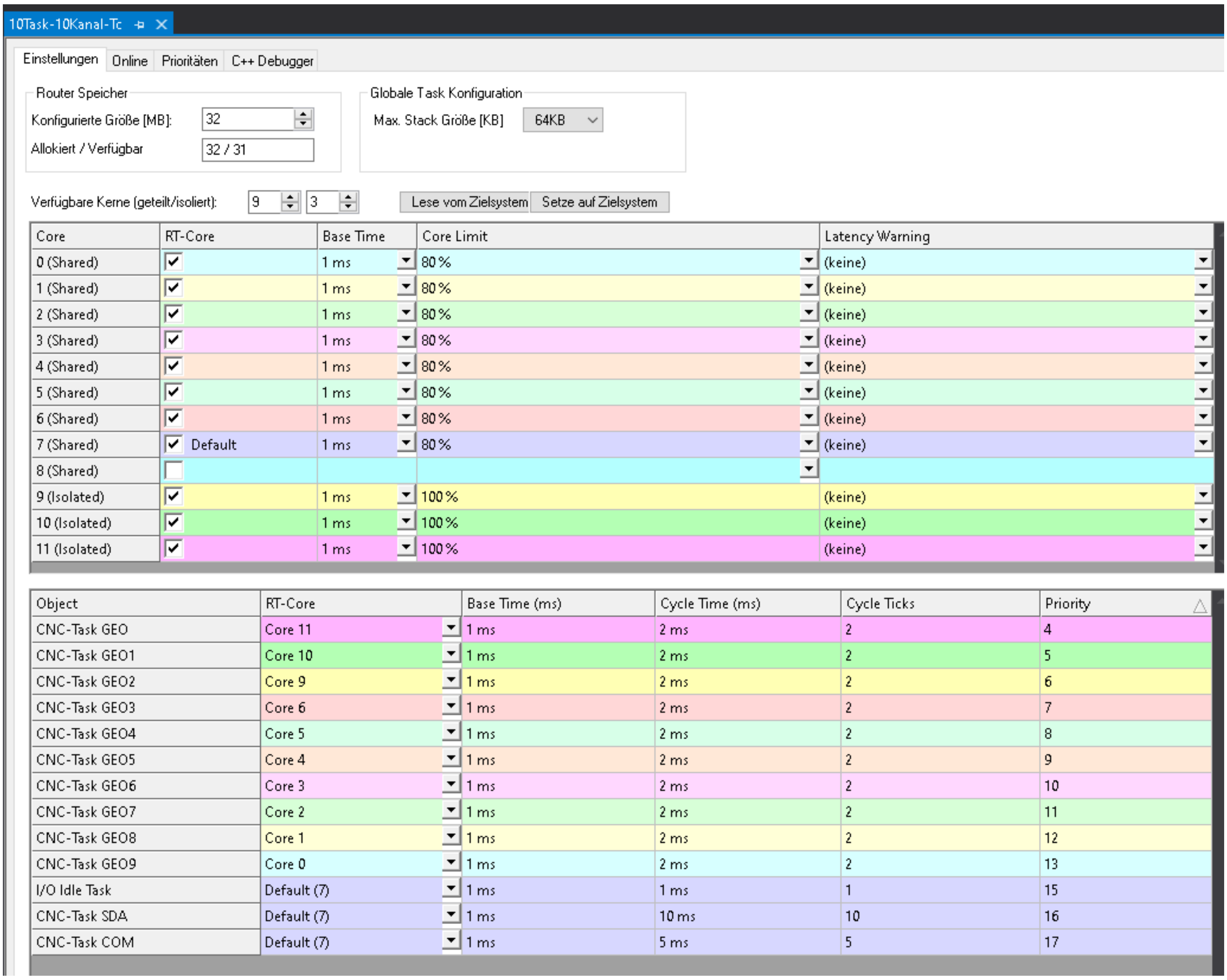


Fig. 13: Assign the interpolator of channel 2 to the new context

Configuration of 10 channels (CNC GEO task) on 10 CPU cores



Core	RT-Core	Base Time	Core Limit	Latency Warning
0 (Shared)	☑	1 ms	80 %	(keine)
1 (Shared)	☑	1 ms	80 %	(keine)
2 (Shared)	☑	1 ms	80 %	(keine)
3 (Shared)	☑	1 ms	80 %	(keine)
4 (Shared)	☑	1 ms	80 %	(keine)
5 (Shared)	☑	1 ms	80 %	(keine)
6 (Shared)	☑	1 ms	80 %	(keine)
7 (Shared)	☑ Default	1 ms	80 %	(keine)
8 (Shared)	☐			
9 (Isolated)	☑	1 ms	100 %	(keine)
10 (Isolated)	☑	1 ms	100 %	(keine)
11 (Isolated)	☑	1 ms	100 %	(keine)

Object	RT-Core	Base Time (ms)	Cycle Time (ms)	Cycle Ticks	Priority
CNC-Task GEO	Core 11	1 ms	2 ms	2	4
CNC-Task GEO1	Core 10	1 ms	2 ms	2	5
CNC-Task GEO2	Core 9	1 ms	2 ms	2	6
CNC-Task GEO3	Core 6	1 ms	2 ms	2	7
CNC-Task GEO4	Core 5	1 ms	2 ms	2	8
CNC-Task GEO5	Core 4	1 ms	2 ms	2	9
CNC-Task GEO6	Core 3	1 ms	2 ms	2	10
CNC-Task GEO7	Core 2	1 ms	2 ms	2	11
CNC-Task GEO8	Core 1	1 ms	2 ms	2	12
CNC-Task GEO9	Core 0	1 ms	2 ms	2	13
I/O Idle Task	Default (7)	1 ms	1 ms	1	15
CNC-Task SDA	Default (7)	1 ms	10 ms	10	16
CNC-Task COM	Default (7)	1 ms	5 ms	5	17

10Task-10Kanal-Tc
Online
1000 µs I/O Idle Task, CPU 7 1.8 µs
Total 2.0 µs
5000 µs CNC-Task COM, CPU 7 0.1 µs
Total 40.7 µs
10000 µs CNC-Task SDA, CPU 7 54.9 µs
Total 57.7 µs
2000 µs CNC-Task GEO, CPU 111.2 µs
Total 106.1 µs
2000 µs CNC-Task GEO1, CPU 105.2 µs
Total 104.7 µs
2000 µs CNC-Task GEO2, CPU 98.0 µs
Total 105.6 µs
2000 µs CNC-Task GEO3, CPU 62.2 µs
Total 107.0 µs
2000 µs CNC-Task GEO4, CPU 54.6 µs
Total 109.6 µs
2000 µs CNC-Task GEO5, CPU 45.8 µs
Total 111.1 µs
2000 µs CNC-Task GEO6, CPU 34.2 µs
Total 108.0 µs
2000 µs CNC-Task GEO7, CPU 25.1 µs
Total 108.2 µs
2000 µs CNC-Task GEO8, CPU 17.1 µs
Total 109.1 µs
2000 µs CNC-Task GEO9, CPU 06.7 µs
Total 110.0 µs
0 µs

5 Diagnostic options

The internal CNC schedule can be logged to diagnose MultiCore functions. This can take place at different times or interactions:

- automatically at CNC start-up
- implicitly on storing/requesting CNC diagnostic data
- by writing a CNC object

Logging first takes place in an internal logging format. Logging is then prepared in a subsequent step into a suitable representation (text format, view).

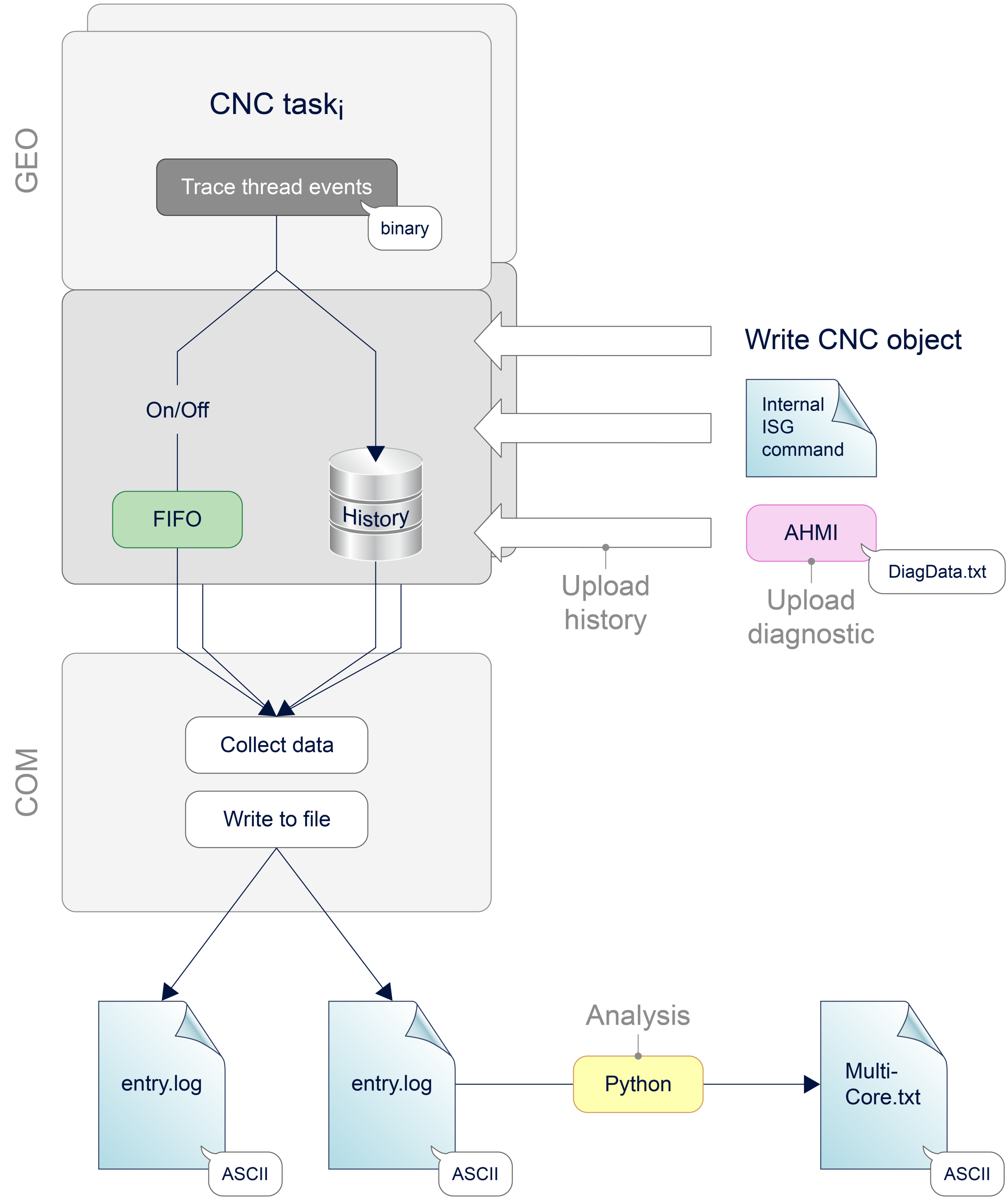


Fig. 14: Logging the MultiCore functions

Controller start-up

For controller start-up, the logging time can be set using P-STUP-00213 [▸ 29] of the schedule events. Logging is deactivated with a value=0 (default).

Events are logged to the text file specified in P-STUP-00214 [▸ 29].

Parameterisation example for diagnosing MultiCore events in the start-up list

```
#
```

```
# ********************************************
# TC_STARTUP_DESC: TwinCAT CNC configuration
# ********************************************
task_trace.geo.max_records        2000
task_trace.geo.filename           multicore-startup.log
task_trace.geo.history_filename   multicore-history.log
#
```

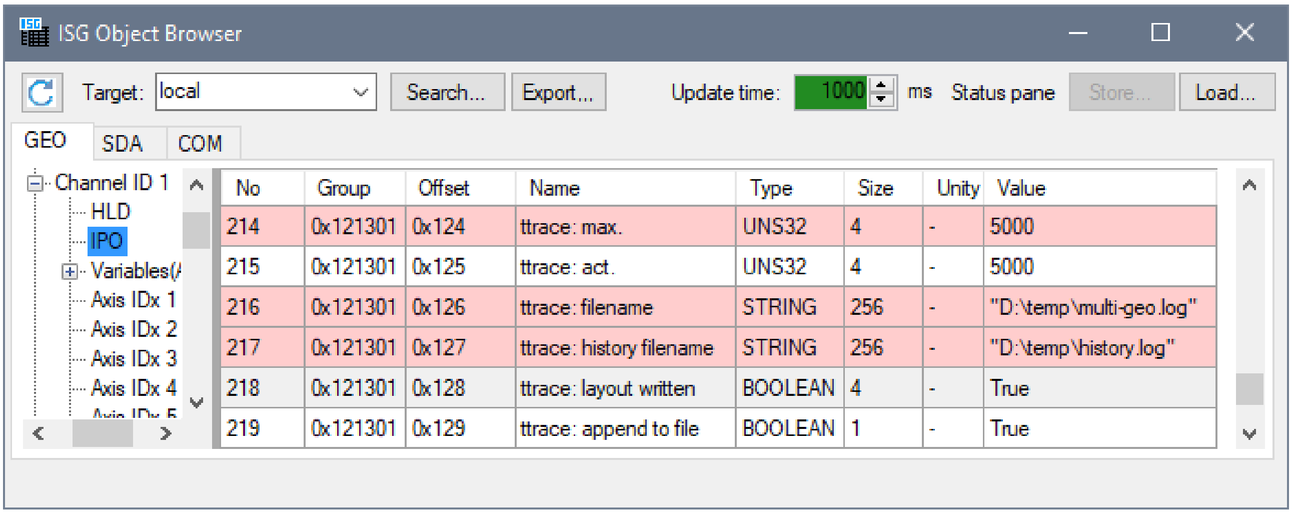
Logging in diagnostic data

When CNC diagnostic data is requested, the past log entries of the schedule (history) are output automatically. The length of the logging time is specified as a fixed value. The name of the output file can be modified using the start-up parameter P-STUP-00215 [▶ 29]

Using CNC objects

The following CNC objects are available for diagnostic purposes:

- ttrace: max. [▶ 31], this parameter can be used to set the maximum number of logs, analogous to P-STUP-00213 [▶ 29]
- ttrace: act. [▶ 31]
- ttrace: filename [▶ 31], analogous to P-STUP-00214 [▶ 29]
- ttrace: history filename [▶ 31], analogous to P-STUP-00215 [▶ 29]
- ttrace: layout written
- ttrace: append to file



No	Group	Offset	Name	Type	Size	Unity	Value
214	0x121301	0x124	ttrace: max.	UNS32	4	-	5000
215	0x121301	0x125	ttrace: act.	UNS32	4	-	5000
216	0x121301	0x126	ttrace: filename	STRING	256	-	"D:\temp\multi-geo.log"
217	0x121301	0x127	ttrace: history filename	STRING	256	-	"D:\temp\history.log"
218	0x121301	0x128	ttrace: layout written	BOOLEAN	4	-	True
219	0x121301	0x129	ttrace: append to file	BOOLEAN	1	-	True

Fig. 15: Available CNC objects in the ISG Object Browser

Example of outputs

The output format, e.g. the MultiCore-Startup.log, looks like this:

```
373664104840000,0,0,4,0,0,0,0
373664104840200,0,5,5,0,2,0,0
373664104840300,0,4,5,0,2,0,0
373664104840400,0,1,5,0,3,0,0
373664104840500,0,2,5,0,3,0,0
373664104840600,0,3,5,0,3,0,0
373664104840800,0,10,5,0,3,0,0
373664104840900,0,8,5,0,3,0,1
373664104841000,0,7,5,0,3,0,0
373664104841100,0,8,5,0,3,0,2
373664104841200,0,9,5,0,3,0,0
373664104841500,0,10,5,0,4,0,0
373664104841600,0,8,5,0,7,0,4
373664104841700,0,0,6,0,0,0,0
373664104841800,0,0,9,0,0,0,3
373664104841900,0,0,10,0,0,0,1
373664104842000,0,0,11,0,0,0,3074
373664104842000,0,0,12,0,0,0,0
```

Fig. 16: Internal logging format

The representations below are produced by an internal tool to prepare the above output format.

```
C:\TwinCAT\3.1\Components\Mc\CNC\Diagnostics\MultiCore-History.txt - Notepad++
File  Edit  Search  View  Encoding  Language  Settings  Tools  Macro  Run  Plugins  Window  ?
MultiCore-History.txt

   CYC_37      6900          <   CHAN 1                   ----SYNC 4                CH_OUT                 OK
   CYC_37       300          >----SYNC 4                  ----SYNC 4    PRE_ADS_BARRIER                    OK
   CYC_37      1100          <----SYNC 4                  ----SYNC 4    PRE_ADS_BARRIER                    OK
   CYC_37      1200                                       <----SYNC 4   PRE_ADS_BARRIER                    OK

-----------------------------------[neuer Zyklus gestartet, Thread 1 Tick Count erhoeht, TC = 117720, erhoeht durch Thread 1]--

   CYC_38   1971100                    CYCLE_TICK_INCR(117720)                     ALL   NEW_CYCLE | EQUI  OK
   CYC_38       300                                        >   CHAN 2              IPO                     OK
   CYC_38       300                                        <   CHAN 2              IPO                     OK
   CYC_38       200                                       >----SYNC 1              IPO                     OK
   CYC_38      1200   CYCLE_TICK_OK(117720)               ----SYNC 1               ALL   NEW_CYCLE | EQUI  OK
   CYC_38       300          >        LR                  ----SYNC 1          POS_CTRL                     OK
   CYC_38     12600          <        LR                  ----SYNC 1          POS_CTRL                     OK
   CYC_38       100          >    FILTER                  ----SYNC 1          POS_CTRL                     OK
   CYC_38       900          <    FILTER                  ----SYNC 1          POS_CTRL                     OK
   CYC_38       900          >    KONFIG                  ----SYNC 1               IPO                     OK
   CYC_38       300          <    KONFIG                  ----SYNC 1               IPO                     OK
   CYC_38         0          >      DIAG                  ----SYNC 1               IPO                     OK
   CYC_38       400          <      DIAG                  ----SYNC 1               IPO                     OK
   CYC_38         0          >     KOMMU                  ----SYNC 1               IPO                     OK
   CYC_38       600          <     KOMMU                  ----SYNC 1               IPO                     OK
   CYC_38       100          >    CHAN 1                  ----SYNC 1               IPO                     OK
   CYC_38      1300          <    CHAN 1                  ----SYNC 1               IPO                     OK
   CYC_38       100          >----SYNC 1                  ----SYNC 1               IPO                     OK
   CYC_38      2000          <----SYNC 1                  ----SYNC 1               IPO                     OK
   CYC_38       100          >      HAND                  ----SYNC 1               IPO                     OK
   CYC_38       900                 HAND                  <----SYNC 1              IPO                     OK
   CYC_38         0                 HAND                  >----SYNC 2              IPO                     OK
   CYC_38       100          <      HAND                  ----SYNC 2               IPO                     OK
   CYC_38       100          >----SYNC 2                  ----SYNC 2               IPO                     OK
   CYC_38       900          <----SYNC 2                  ----SYNC 2               IPO                     OK
   CYC_38         0          >    BAHN 1                  ----SYNC 2               IPO                     OK
   CYC_38      1200               BAHN 1                  <----SYNC 2              IPO                     OK
   CYC_38       100               BAHN 1                  >    BAHN 2              IPO                     OK
   CYC_38      5800               BAHN 1                  <    BAHN 2              IPO                     OK
   CYC_38       200               BAHN 1                  >    CHAN 2           CH_OUT                     OK
   CYC_38      1000          <    BAHN 1                       CHAN 2              IPO                     OK
   CYC_38       300          >    CHAN 1                       CHAN 2           CH_OUT                     OK
   CYC_38      2100               CHAN 1                  <    CHAN 2           CH_OUT                     OK
   CYC_38       100               CHAN 1                  >----SYNC 4   PRE_ADS_BARRIER                    OK
   CYC_38      9000          <    CHAN 1                  ----SYNC 4            CH_OUT                     OK
   CYC_38       200          >----SYNC 4                  ----SYNC 4    PRE_ADS_BARRIER                    OK
   CYC_38      1100          <----SYNC 4                  ----SYNC 4    PRE_ADS_BARRIER                    OK
   CYC_38      1200                                       <----SYNC 4   PRE_ADS_BARRIER                    OK

-----------------------------------[neuer Zyklus gestartet, Thread 1 Tick Count erhoeht, TC = 117721, erhoeht durch Thread 1]--

   CYC_39   1953800                    CYCLE_TICK_INCR(117721)                     ALL   NEW_CYCLE | EQUI  OK
   CYC_39       300                                        >   CHAN 2              IPO                     OK
   CYC_39       300                                        <   CHAN 2              IPO                     OK
```

Fig. 17: Example output of a MultiCore schedule diagnostic file

Each MultiCore log file contains internal context information at the start, such as version number, schedule type, cycle times, etc., which make it easier to diagnose error response.

```
*******************************************************************************
*                                                                             *
*     Diagnose-Skript                                                         *
*     ---------------                                                         *
*                                                                             *
*     Version                            = 1.0.0.0                            *
*                                                                             *
*     CNC                                                                     *
*     ---                                                                     *
*                                                                             *
*     Version                            = 3.1.3074.0                         *
*     Scheduling                         =   Complete                         *
*     Zykluszeit                         =       2000 us                      *
*     THREAD_TRACE_ENTRIES_MAX           =        600                         *
*     THREAD_TRACE_HISTORY_LAYOUT_MAX    =         50                         *
*     THREAD_TRACE_HISTORY_ENTRIES_MAX   =       1000                         *
*     TASK_GEO_NUMBER_OF_THREADS_MAX     =         16                         *
*     SYNC_BARRIER_NUMBER                =          6                         *
*     SYNC_BARRIER_MAX_TASKS             =         16                         *
*                                                                             *
*     EINGABE                                                                 *
*     -------                                                                 *
*                                                                             *
*     Log-Datei                          =             MultiCore-Test.log *
*                                                                             *
*     AUSWERTUNG UNTER FOLGENDEN REGELN                                       *
*     ---------------------------------                                       *
*                                                                             *
*     R1: in jedem Zyklus muss der erste Eintrag CYCLE_TICK_INCR sein,        *
*         bei allen anderen Threads muss der erste Eintrag CYCLE_TICK_OK sein *
*     R2: in jedem Zyklus ueberlappen sich entsprechende Thread-Barrieren     *
*     R3: fuer jeden Thread bleibt die Sequenz der BFs pro Zyklus gleich      *
*     R4: in jedem Zyklus muss es in jedem Thread genau ein TICK-Event geben  *
*     R5: in jedem Zyklus muessen die Tick-Counts aller Threads gleich sein   *
*                                                                             *
*     ERGEBNIS                                                                *
*     --------                                                                *
*                                                                             *
*     Anzahl Fehler                      =          8                         *
*                                                                             *
*     STATISTISCHE ANGABEN                                                    *
*     --------------------                                                    *
*                                                                             *
*     Anzahl Logeintraege                =      11044                         *
*     Anzahl Zyklen                      =        290                         *
*     Zyklendauer - Minimum              =      17500 ns    =   17 us         *
*     Zyklendauer - Mittelwert           =      24351 ns    =   24 us         *
*     Zyklendauer - Maximum              =     121100 ns    =  121 us         *
*     Zyklendauer - Standardabweichung   =       9368 ns    =    9 us         *
*                                                                             *
*******************************************************************************
```

Fig. 18: Script example

5.1 Internal ISG command

Syntax:

#INFO [TO SCHEDULER_CMD S0 = TRACE [**S1 =** <filename>] [**I0 =** <max_number>] **]**

TRACE	Start logging until the specified number of CNC cycles
<filename>	Optionally, you can specify the filename used for the output of scheduling events. Default: <TwinCATInstallation>\Components\Mc\Cnc\Diagnostics\MultiCore-Startup.log
<max_number>	Defines the number of CNC cycles logged. Logging is deactivated with a value=0.

Syntax:

#INFO [TO SCHEDULER_CMD S0 = HISTORY [**S1 =** <filename>] **]**

HISTORY	Writes the past schedule events (history) to the specified file. A fixed logging time is specified.
<filename>	Optionally, you can specify the filename used for the output of past scheduling events (history). Default: <TwinCATInstallation>\Components\Mc\Cnc\Diagnostics\MultiCore-History.log

Using the #INFO command

```
%MultiCore

P1 = 1
N090 V.E.string = "C:\TwinCAT\3.1\Components\Mc\CNC\Diagnostics\MultiCore-Loop" + ".txt"

N100 #FILENAME [ MSG = V.E.string]

$WHILE P1

N010 G01 X0 Y0 Z0 F10000
N040 #INFO[TO SCHEDULER_CMD S0=TRACE S1 = C:\TwinCAT\3.1\Components\Mc\CNC\Diagnostics\MultiCore-
Test.log I0=300 ]
N100 X300
N110 Y400
N240 #INFO[TO SCHEDULER_CMD S0=HISTORY S1 = C:\TwinCAT\3.1\Components\Mc\CNC\Diagnostics\MultiCore-
History.log ]
N110 Z500

N110 #MSG SAVE["%s MultiCore-Test.nc Loop %d", V.G.TIME_STAMP, P1]
N600 #FLUSH WAIT
N400 P1 = P1 + 1

$ENDWHILE
M30
```

6 Parameter

6.1 Overview

restart

ID	Parameter	Description
P-STUP-00213	max_records	Number of logging entries for logging
P-STUP-00214	filename	Name of the output file
P-STUP-00215	history_filename	Name of the history file

Channel parameters

ID	Parameter	Description
P-CHAN-00409	com	Context information of the COM task
P-CHAN-00410	geo	Context information of the GEO task
P-CHAN-00411	sda	Context information of the SDA task

6.2 Description

6.2.1 restart

P-STUP-00213	**Number of logging entries for logging**
Description	This parameter sets the maximum number of log entries for the corresponding task. Real-time events are logged in these entries for diagnostic purposes. After the number is reached, logging stops automatically. With a value=0, no log file is generated at CNC start-up.
Parameter	trace.geo.max_records
Data type	SGN32
Data range	0 <= max_records < MAX_UNS32
Dimension	----
Default value	0
Remarks	Parameter available as of CNC Build V3.1.3077 and higher

P-STUP-00214	**Name of the output file**
Description	This parameter is used to specify the name of the output file for logging the corresponding task. If no path is specified for the output file, the default path or the main directory of the NC controller is used.
Parameter	task_trace.geo.filename
Data type	STRING
Data range	<Filename with relative / absolute path>
Dimension	----
Default value	MultiCore-Startup.log
Remarks	Parameter available as of CNC Build V3.1.3077 and higher

P-STUP-00215	**Name of the history file**
Description	This parameter is used to specify the name of the history file for logging the corresponding task. The file is used to output the history logs. If no path is specified for the file, the default path or the main directory of the NC controller is used.
Parameter	task_trace.geo.history_filename
Data type	STRING
Data range	<Filename with relative / absolute path>
Dimension	----
Default value	MultiCore-History.log
Remarks	Parameter available as of CNC Build V3.1.3077 and higher

6.2.2 Channel parameters

P-CHAN-00409	**Context information of the COM task**
Description	This parameter defines the context information of the COM task. The context information can contain a reference to the context of a CPU thread. See also P-RTCF-00017.
Parameter	schedule.context.com or twincat.context.com
Data type	UNS32
Data range	

Dimension	----
Default value	0
Remarks	This parameter is used automatically in TwinCAT systems.

P-CHAN-00410	**Context information of the GEO task**
Description	This parameter defines the context information of the GEO task. The context information can contain a reference to the context of a CPU thread. See also P-RTCF-00017.
Parameter	schedule.context.geo or twincat.context.geo
Data type	UNS32
Data range	
Dimension	----
Default value	0
Remarks	This parameter is used automatically in TwinCAT systems.

P-CHAN-00411	**Context information of the SDA task**
Description	This parameter defines the context information of the SDA task. The context information can contain a reference to the context of a CPU thread. See also P-RTCF-00017.
Parameter	schedule.context.sda or twincat.context.sda
Data type	UNS32
Data range	
Dimension	----
Default value	0
Remarks	This parameter is used automatically in TwinCAT systems.

6.2.3 CNC objects

Name	ttrace: max.		
Description	This object defines the maximum number of CNC cycles to be logged. This is analogous to P-STUP-00213 [▸ 29].		
Task	GEO (Port 551)		
Index group	0x12130<C_{ID}>	**Index offset**	0x124
Data type	UNS32	**Length**	4
Attributes	read/ write	**Unit**	-
Remarks			

Name	ttrace: act.		
Description	This object reads the current fill level of the log file. The log file can be specified by the CNC object “ttrace: filename” [▸ 31] or by P-STUP-00214 [▸ 29].		
Task	GEO (Port 551)		
Index group	0x12130<C_{ID}>	**Index offset**	0x125
Data type	UNS32	**Length**	4
Attributes	read	**Unit**	-
Remarks			

Name	ttrace: filename		
Description	This object specifies the name of the output file analogous to P-STUP-00214 [▸ 29] .		
Task	GEO (Port 551)		
Index group	0x12130<C_{ID}>	**Index offset**	0x126
Data type	STRING	**Length**	256
Attributes	read/ write	**Unit**	-
Remarks			

Name	ttrace: history filename		
Description	This object specifies the name of the history file analogous to P-STUP-00215 [▸ 29] .		
Task	GEO (Port 551)		
Index group	0x12130<C_{ID}>	**Index offset**	0x127
Data type	STRING	**Length**	256
Attributes	read/ write	**Unit**	-
Remarks			

7 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963-157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963-460
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

Index

P

More Information:
www.beckhoff.com/TF5200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com