DOCUMENTATION ISG-kernel

Functional description MultiCore

Short description:
FCT-C39

© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
All rights reserved
www.isg-stuttgart.de
support@isg-stuttgart.de

Documentation version: 1.051
04/06/2025

Preface

Legal information

This documentation was produced with utmost care. The products and scope of functions described are under continuous development. We reserve the right to revise and amend the documentation at any time and without prior notice.

No claims may be made for products which have already been delivered if such claims are based on the specifications, figures and descriptions contained in this documentation.

Personnel qualifications

This description is solely intended for skilled technicians who were trained in control, automation and drive systems and who are familiar with the applicable standards, the relevant documentation and the machining application.

It is absolutely vital to refer to this documentation, the instructions below and the explanations to carry out installation and commissioning work. Skilled technicians are under the obligation to use the documentation duly published for every installation and commissioning operation.

Skilled technicians must ensure that the application or use of the products described fulfil all safety requirements including all applicable laws, regulations, provisions and standards.

Further information

Links below (DE)

https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

or (EN)

https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

contains further information on messages generated in the NC kernel, online help, PLC libraries, tools, etc. in addition to the current documentation.

Disclaimer

It is forbidden to make any changes to the software configuration which are not contained in the options described in this documentation.

Trade marks and patents

ISG®, ISG kernel®, ISG virtuos®, ISG dirigent®, TwinStore® and the associated logos are registered and licensed trade marks of ISG Industrielle Steuerungstechnik GmbH.

The use of other trade marks or logos contained in this documentation by third parties may result in a violation of the rights of the respective trade mark owners.

Copyright

© ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Germany.

No parts of this document may be reproduced, transmitted or exploited in any form without prior consent. Non-compliance may result in liability for damages. All rights reserved with regard to the registration of patents, utility models or industrial designs.

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

➢ 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.

Attention

Restriction or error

This icon describes restrictions or warns of errors.

Notice

Tips and other notes

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

Example

General example

Example that clarifies the text.

Programing Example

NC programming example

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

Release Note

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.

Release Note

This function is available as of CNC Build V3.1.3077

Parameterisation

Parameterisation depends on the real-time system used.

- For TwinCAT systems in the TwinCAT development environment
- For real-time Linux using real-time and channel parameters

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.

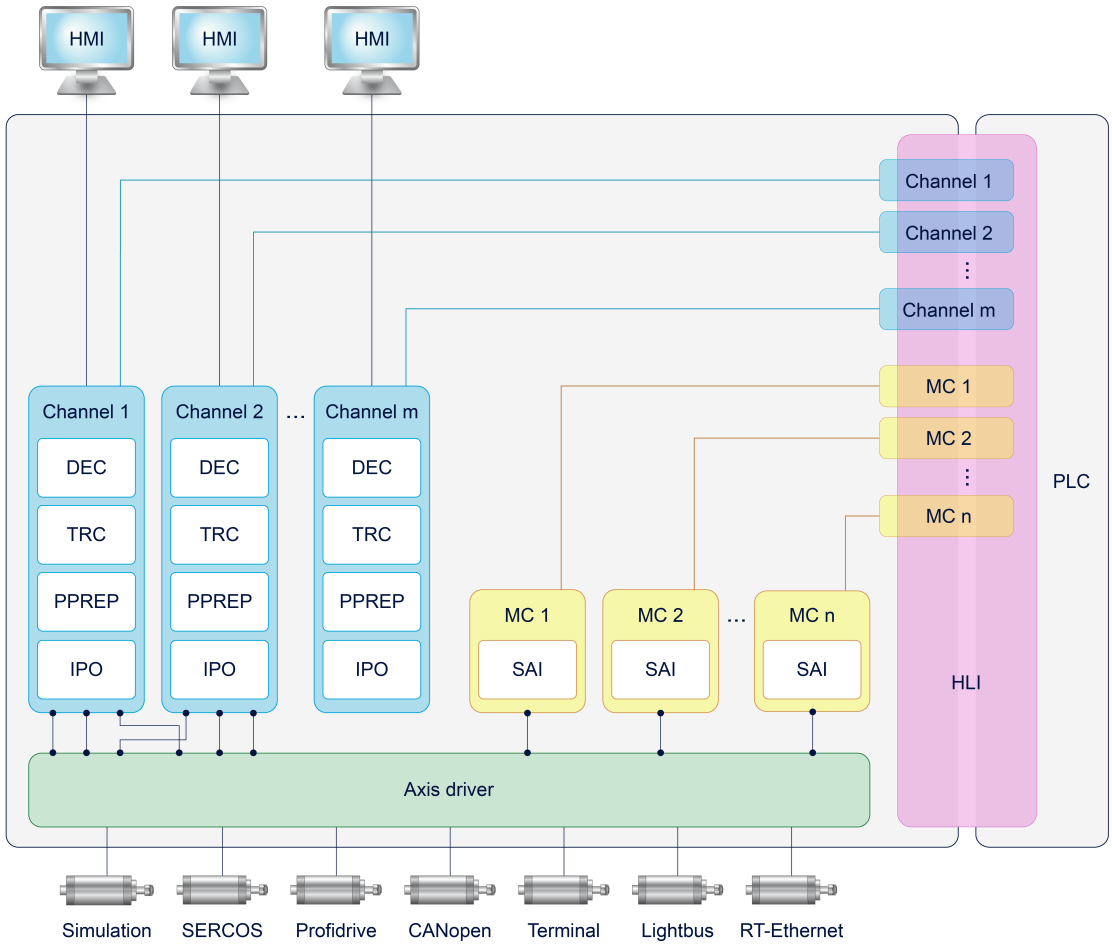


Fig. 1: Structure of a multi-channel CNC

HMI:	User interface	IPO:	Path interpolator
DEC:	Decoder	SAI:	Single-axis interpolator
TRC:	Tool radius compensation	MC:	Motion Control
PPREP:	Path preparation		

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.

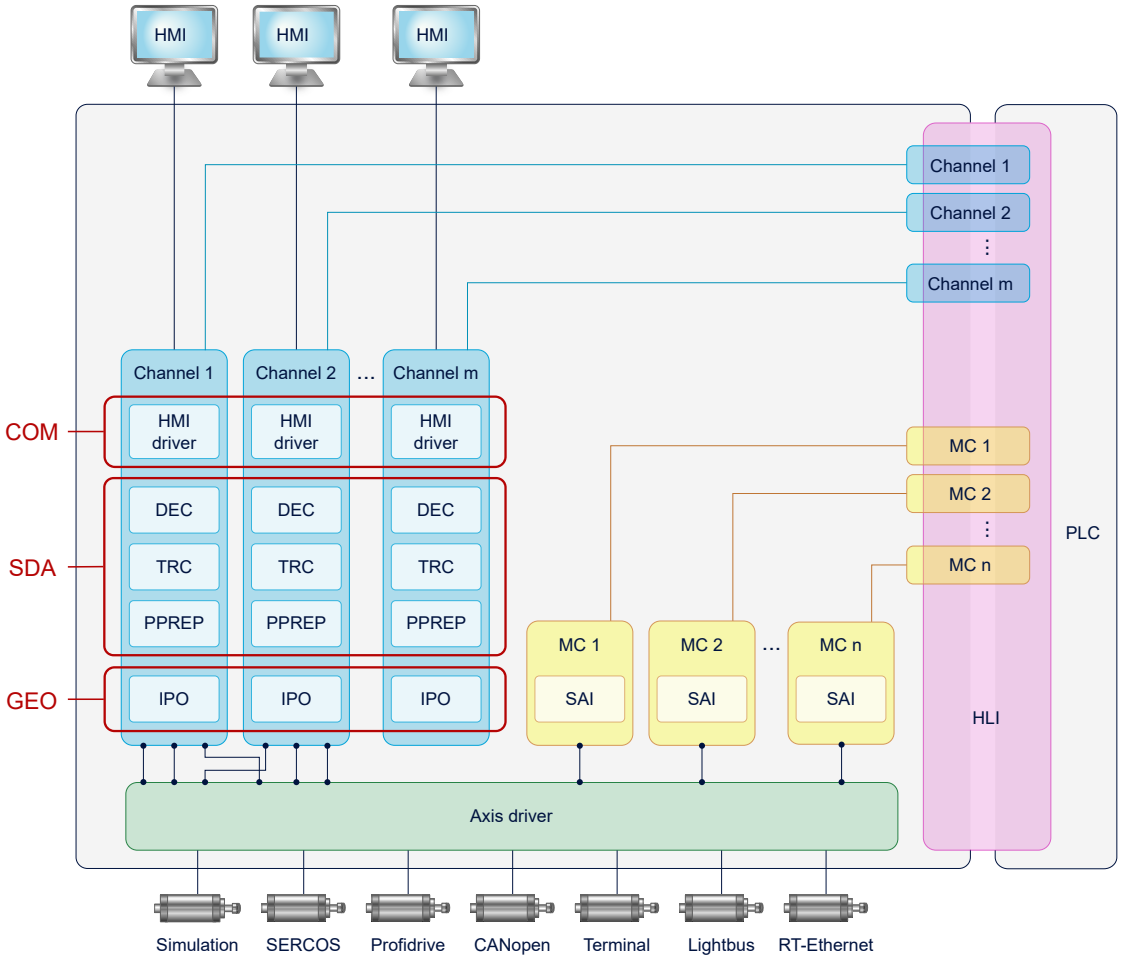


Fig. 2: Standard tasks of a multi-channel CNC

2.2 Configuration of GEO tasks

Standard task distribution of a multi-channel configuration

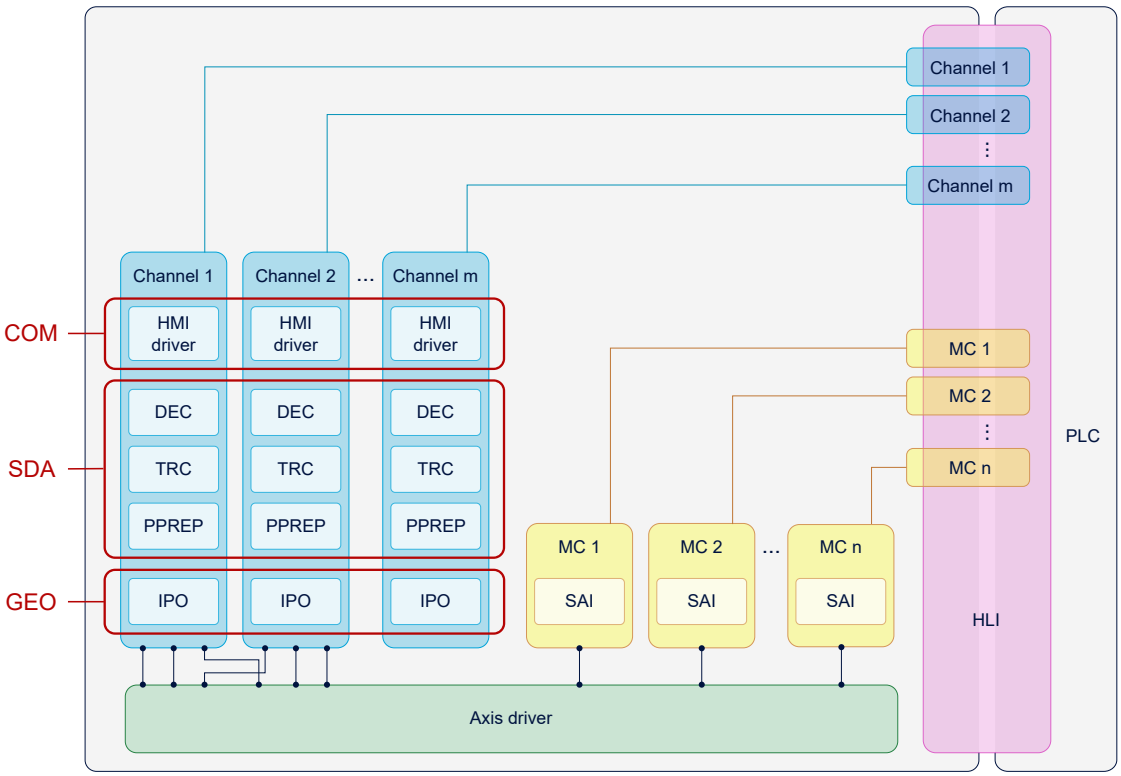


Fig. 3: Initial position (without multicore channel distribution)

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:

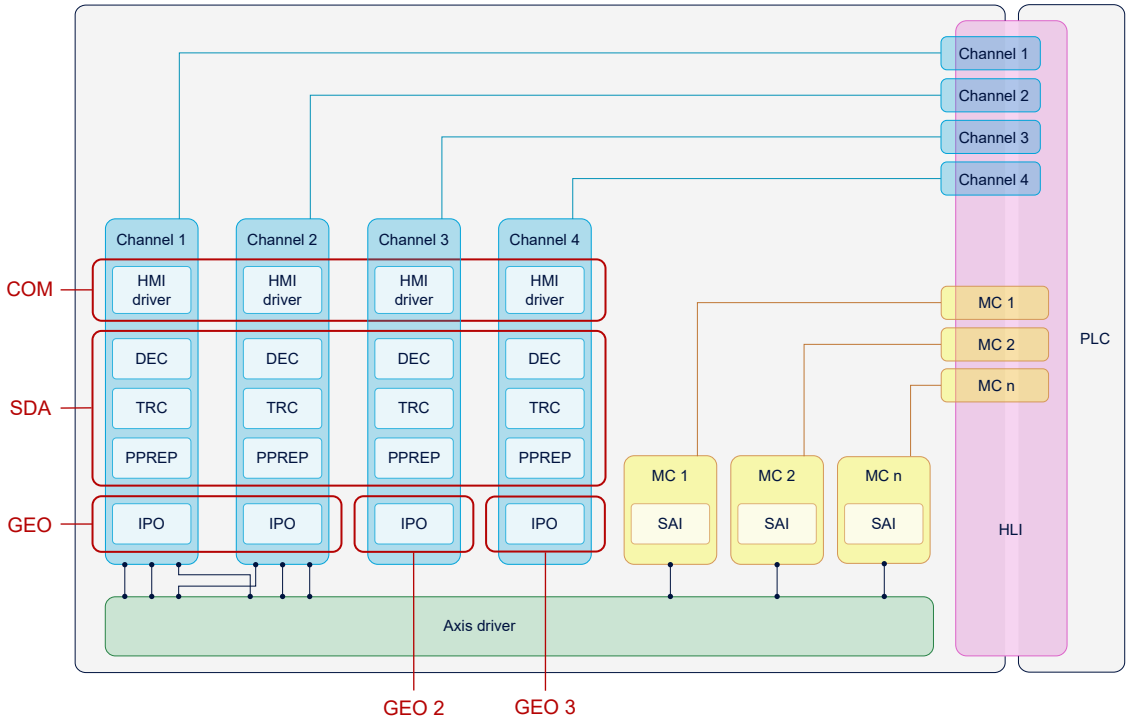


Fig. 4: Distribution of GEO tasks of a 4-channel CNC to multiple cores

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.

Notice

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

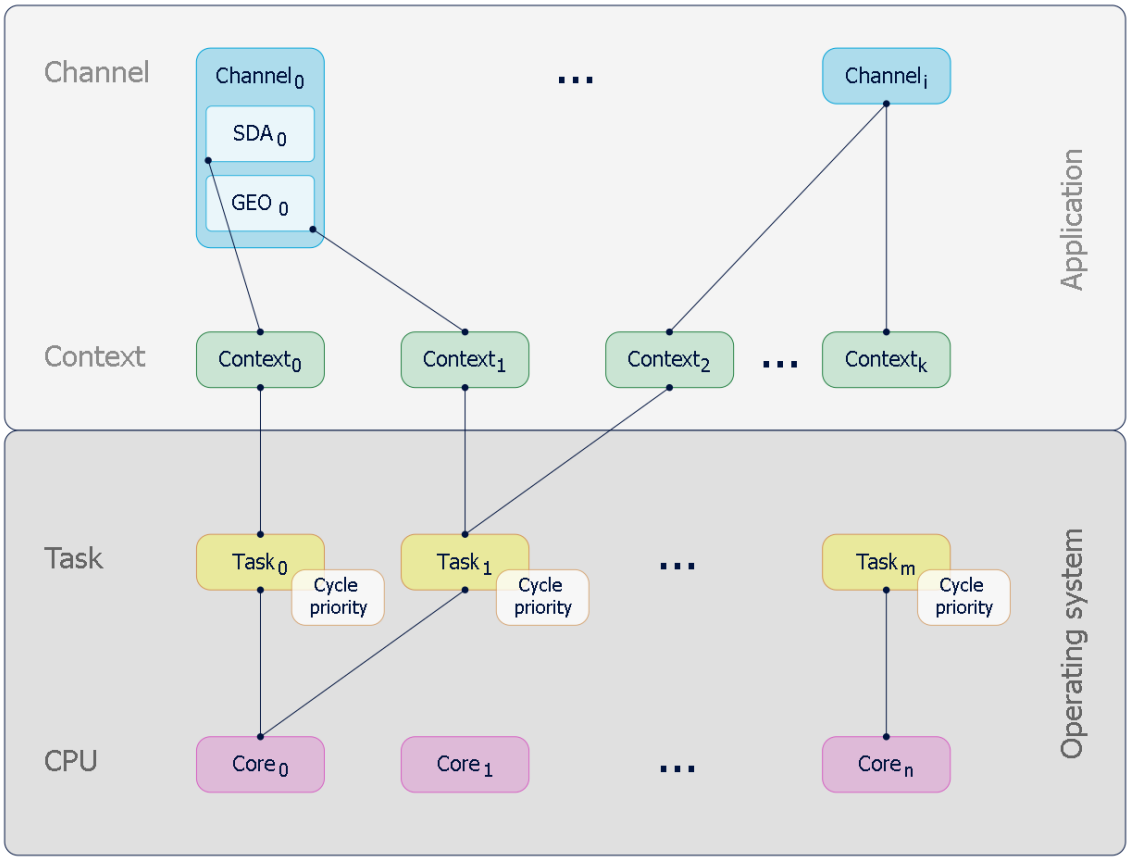


Fig. 5: Assignment by context

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

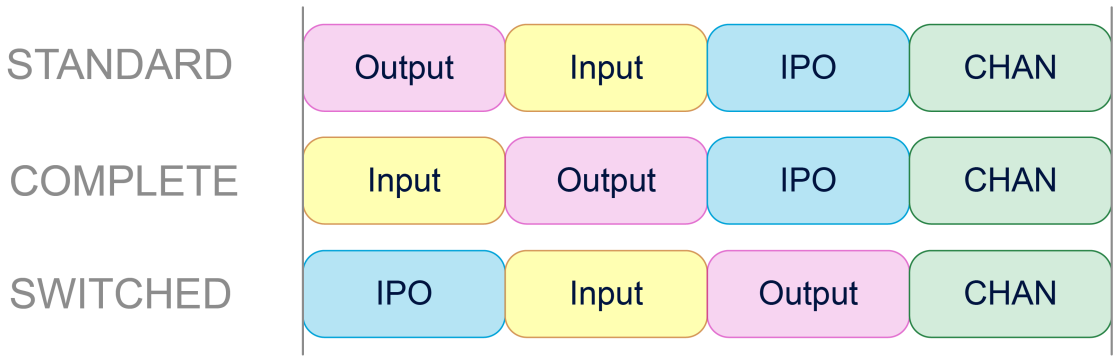


Fig. 6: Overview of task order in the cycle

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.

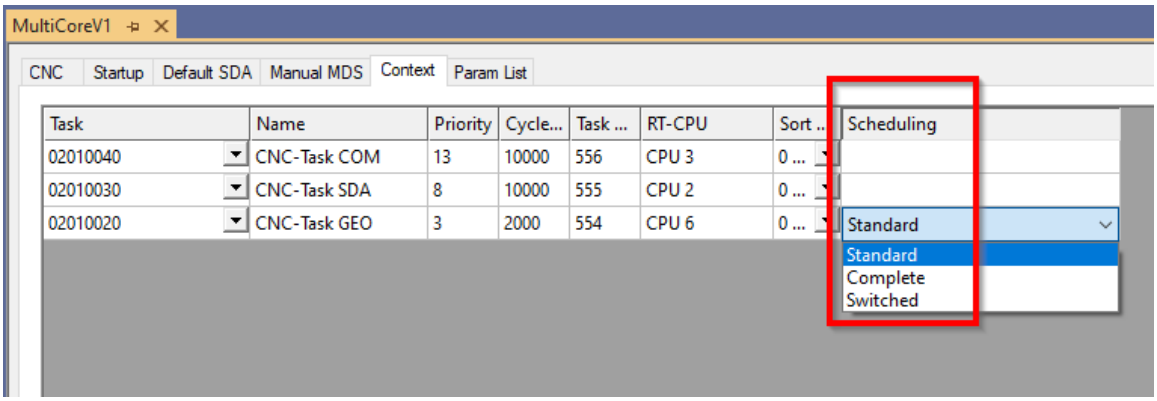
SWITCHED

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 [▶ 32] 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.

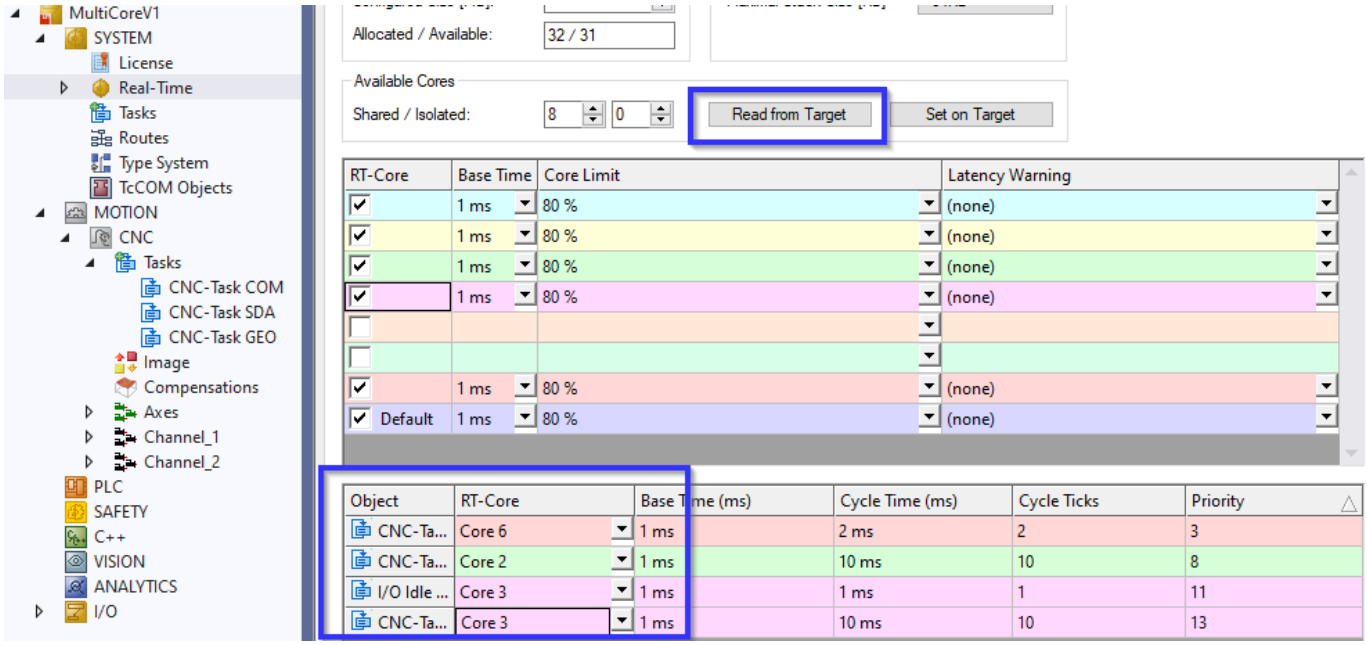


Fig. 7: Determine the available CPU cores

Notice

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:

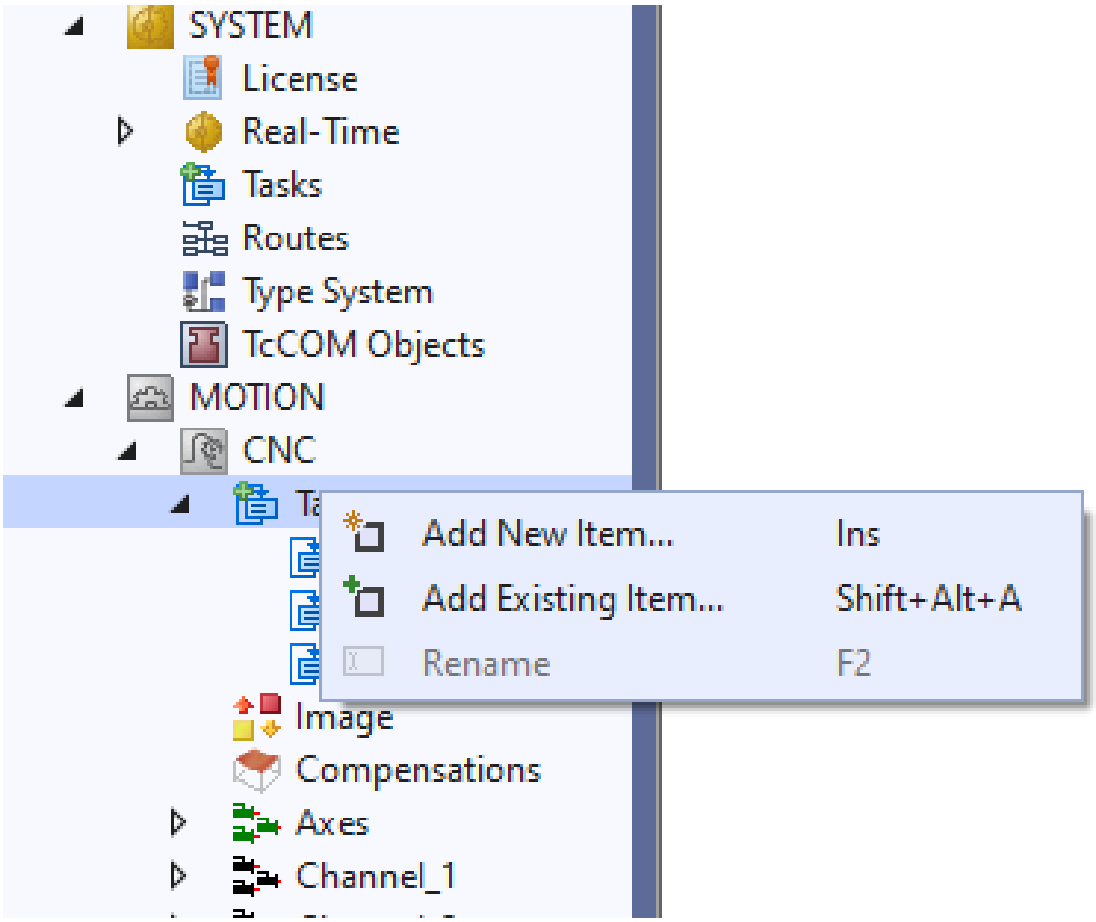


Fig. 8: Generating a new GEO task

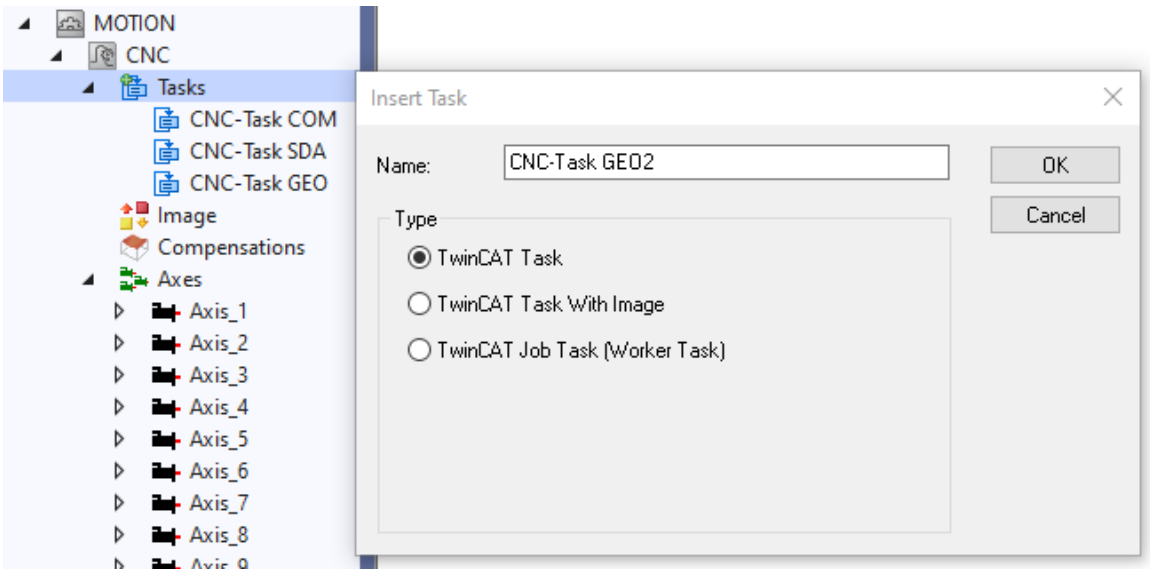


Fig. 9: Create a GEO task with name

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

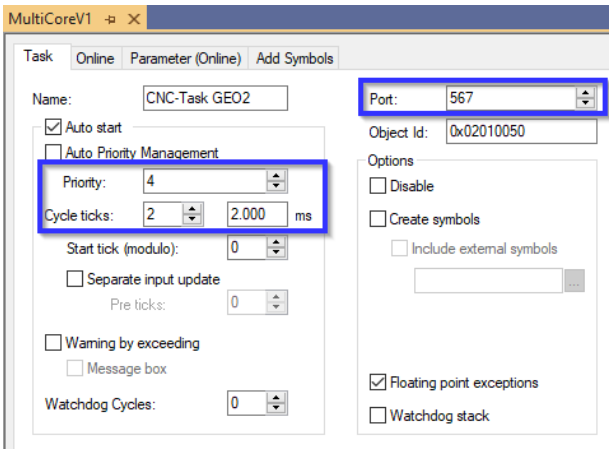


Fig. 10: Settings of task priorities

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.

Notice

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

Creating the context between CPU core and the new CNC task

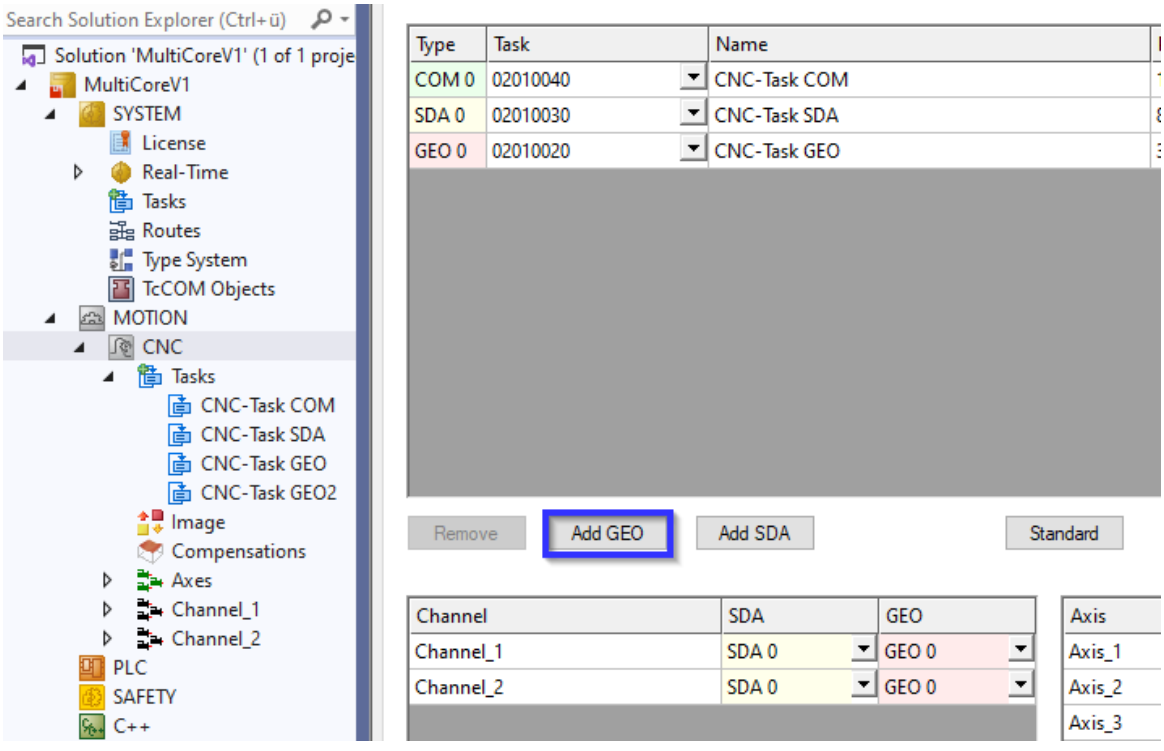


Fig. 11: Create context for new GEO task

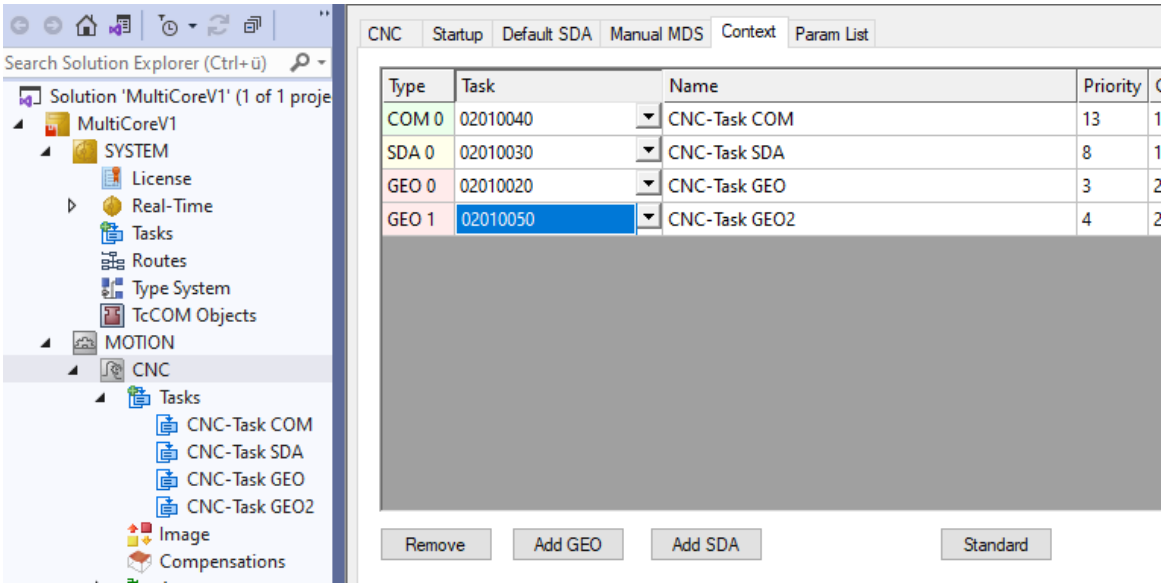


Fig. 12: Assign the new GEO task to the context created

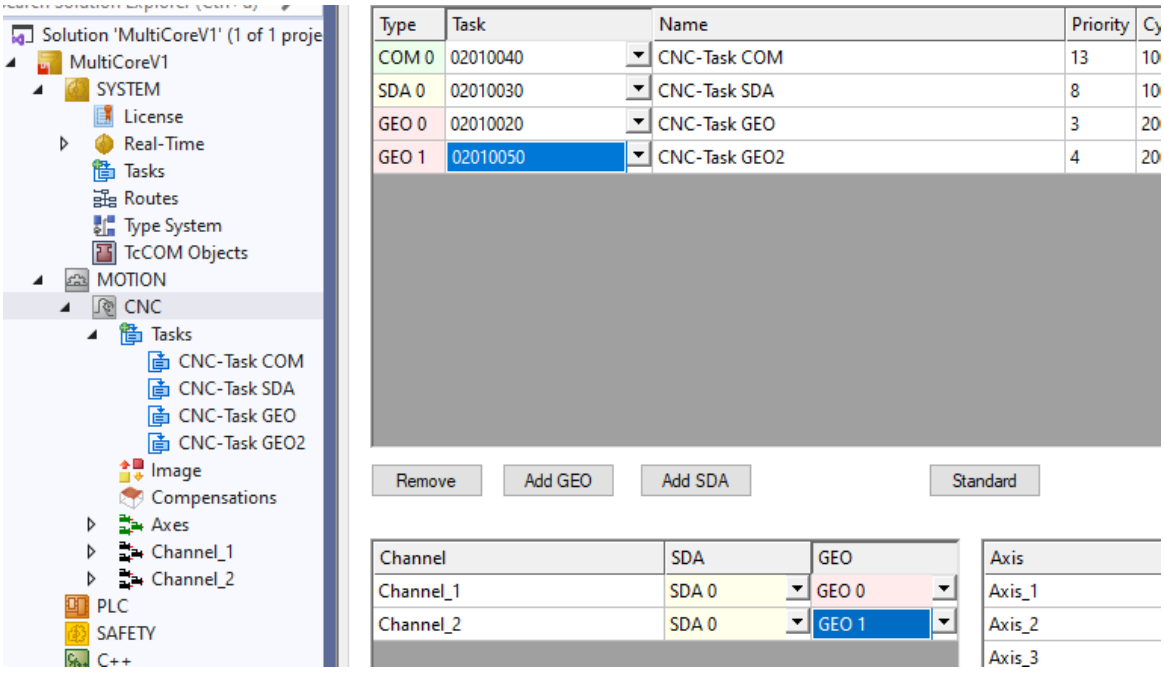


Fig. 13: Assign the interpolator of channel 2 to the new context

Notice

The context of all axes must be identical with the context of the GEOs of the first channel.

If a different context was selected for an axis, error ID 1001851 is output.

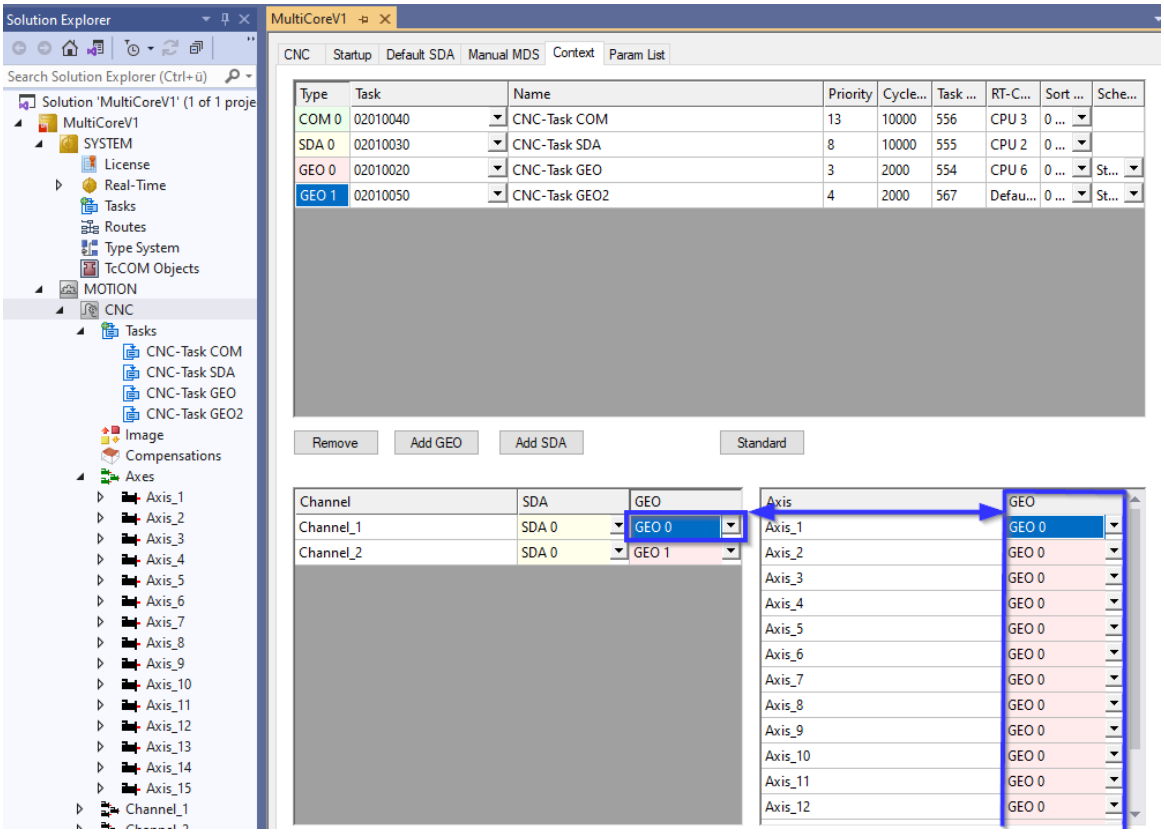
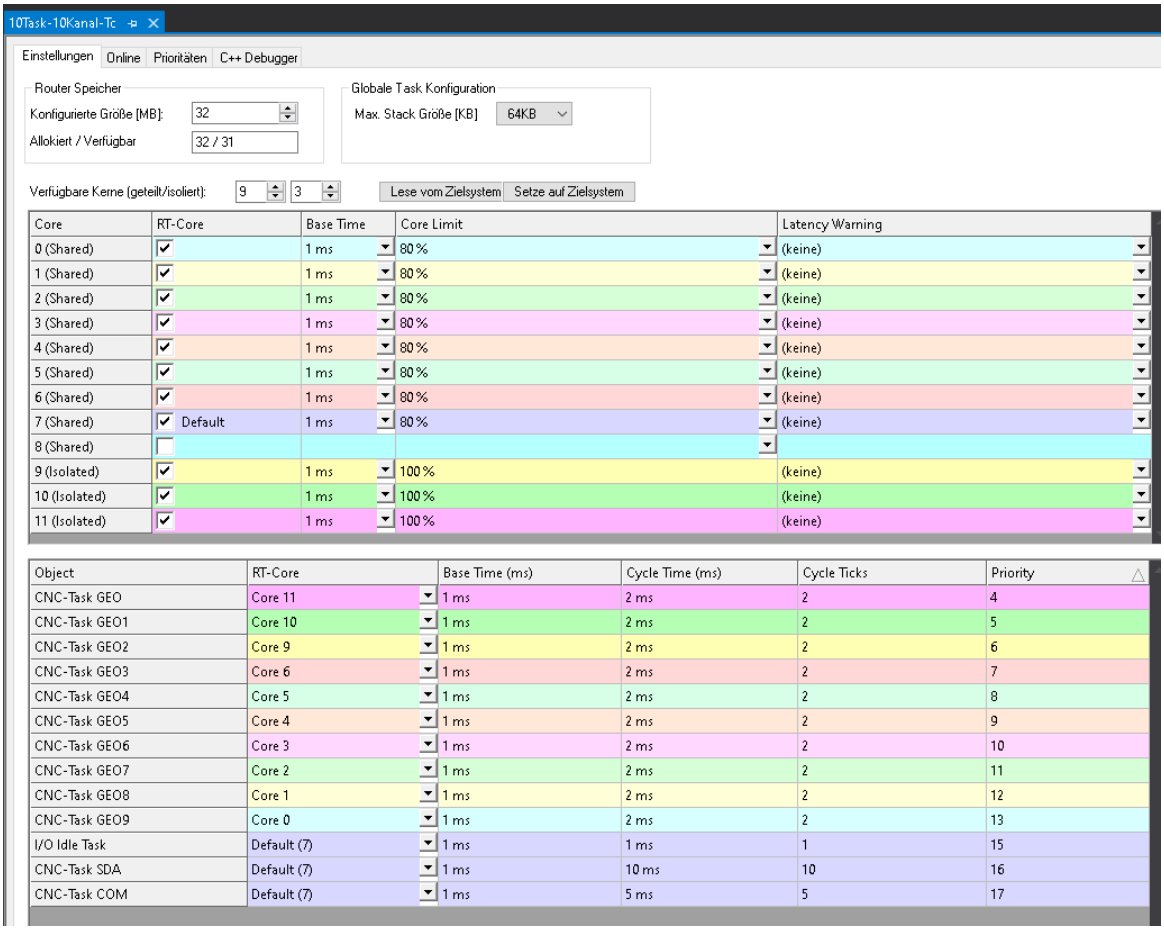


Fig. 14: Context assignment of axes

Example

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 7J 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 754.9 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

4.2 Real-time Linux / Windows Simulator

The threads are assigned to the CPU cores in the realtime configuration of the CNC. Parameterisation takes place in rt_conf.lis.

A CPU core can be assigned to the thread by P-RTCF-00015 [▶ 31]. If this value is invalid, the warning ID 1000181 is output and the operating system assumes the assignment.

The individual threads are assigned to CNC tasks by means of context entries. The value of the context entry (P-RTCF-00017 [▶ 31]) of the thread and the value of the context entry in the channel list must match the particular task.

Example

Parameterisation example: rt_conf.lis

```
# ***************************************
#
#  RT configuration
#
# ***************************************
#  LINUX      [1; 99]   1   highest
#  RTX        [0; 127]  127 highest
#  VXWORKS    [0; 255]  0   highest
#  WindowsCE  [0; 255]  0   highest, 9 CANopen, 10-15 CNC, 16-64 PLC
#
# trace bit : 0 - off, 1 - DTR on COM1, 2 - RTS on COM1, 3 - DTR on
COM2, 4 - RTS on COM2
#
#
interrupt_source                          1    # 1 == internal timer, 2 ==
external IRQ, 3 == external semaphore
cycle_time                             2000    # cycle time in micro s
time_slice                             1000
windows_time                              0    # 0 turned off
schedule                                  1    # Standard
#
thread[0].name                                 GEO1
thread[0].cpu                                  1   # CPU core 1
thread[0].context_info                         0
thread[0].cycle                                2000
thread[0].priority                             31 # HIGHEST
thread[0].error_on_overflow                    1
thread[0].function[0].name                     task_int
thread[0].function[0].calls_per_cycle          1
thread[0].function[0].trace_bit                1
#
thread[1].name                                 BACKGROUND
thread[1].cpu                                  2   # CPU core 2
thread[1].context_info                         4
thread[1].cycle                                1000
thread[1].priority                             34 # NORMAL
thread[1].error_on_overflow                    0
thread[1].function[0].name                     task_rnd
thread[1].function[0].calls_per_cycle          1
thread[1].function[0].trace_bit                3
#
thread[2].name                                 MMI_DRIVER
thread[2].cpu                                  2   # CPU core 2
thread[2].context_info                         6
```

```
thread[2].cycle                              12000
thread[2].priority                           35
thread[2].error_on_overflow                  0
thread[2].function[0].name                   task_mmi_driver
thread[2].function[0].calls_per_cycle        1
thread[2].function[0].trace_bit              0
#
thread[3].name                               COM
thread[3].cpu                                2   # CPU core 2
thread[3].context_info                       5
thread[3].cycle                              4000
thread[3].priority                           35
thread[3].error_on_overflow                  0
thread[3].function[0].name                   task_com
thread[3].function[0].calls_per_cycle        1
thread[3].function[0].trace_bit              4
#
thread[4].name                               GEO2
thread[4].cpu                                3   # CPU core 3
thread[4].context_info                       1
thread[4].cycle                              2000
thread[4].priority                           31 # HIGHEST
thread[4].error_on_overflow                  1
thread[4].function[0].name                   task_int
thread[4].function[0].calls_per_cycle        1
thread[4].function[0].trace_bit              1
#
# external_thread[0].semaphore_name     external_sps
# external_thread[0].cycle              10000
#
End
```

The related parameters for the particular assignments in the channels are as follows:

Channel parameter list 1 - GEO1, SDA and the COM

```
schedulde.context.geo                0
schedulde.context.com                5
schedulde.context.sda                4
```

Channel parameter list 2 – GEO2

```
schedulde.context.geo                1
```

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).

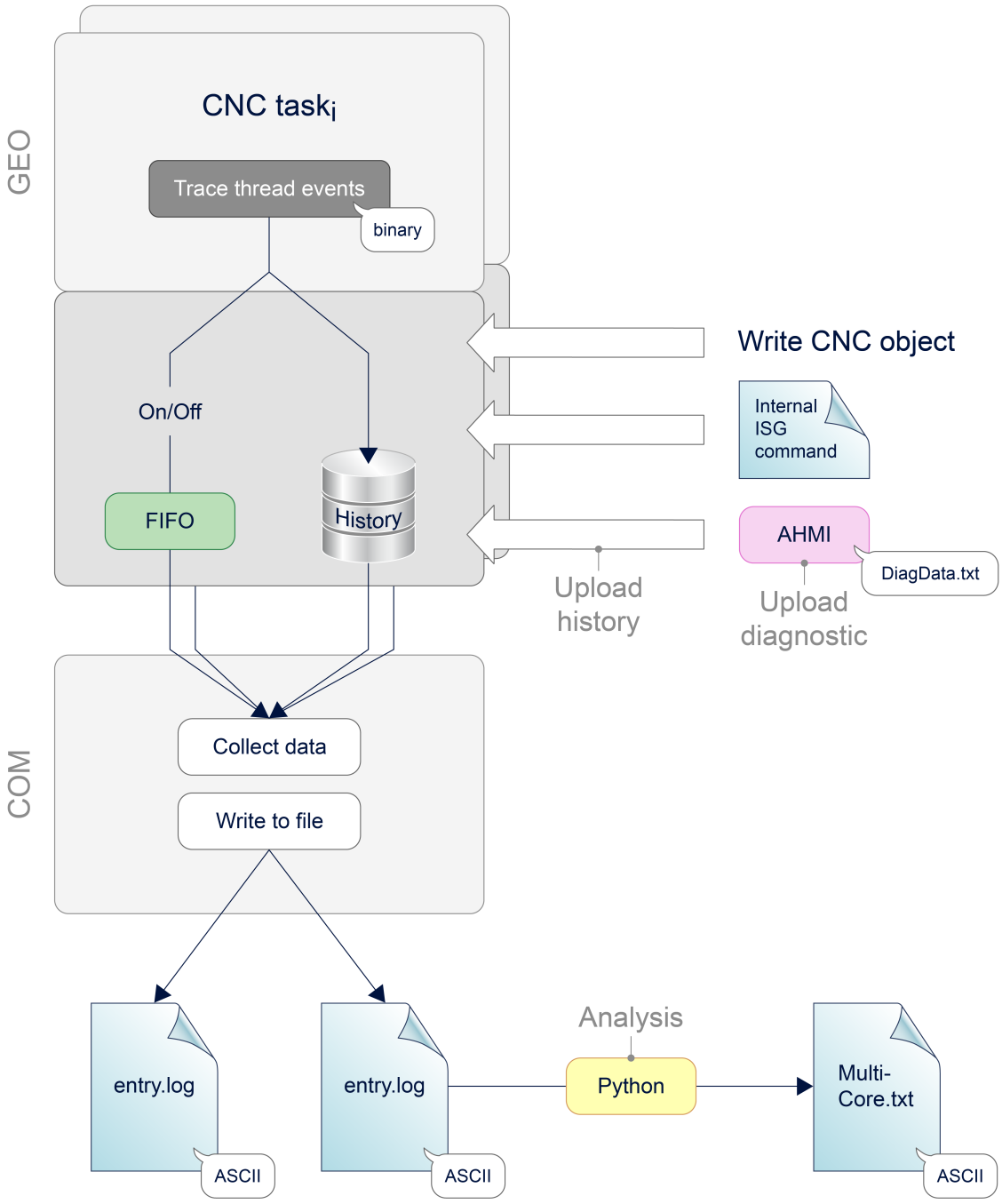


Fig. 15: Logging the MultiCore functions

Controller start-up

For controller start-up, the logging time can be set using P-STUP-00213 [▶ 28] of the schedule events. Logging is deactivated with a value=0 (default).

Events are logged to the text file specified in P-STUP-00214 [▶ 28].

Example

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. [▶ 32], this parameter can be used to set the maximum number of logs, analogous to P-STUP-00213 [▶ 28]
- ttrace: act. [▶ 32]
- ttrace: filename [▶ 33], analogous to P-STUP-00214 [▶ 28]
- ttrace: history filename [▶ 33], analogous to P-STUP-00215 [▶ 29]
- ttrace: layout written
- ttrace: append to file

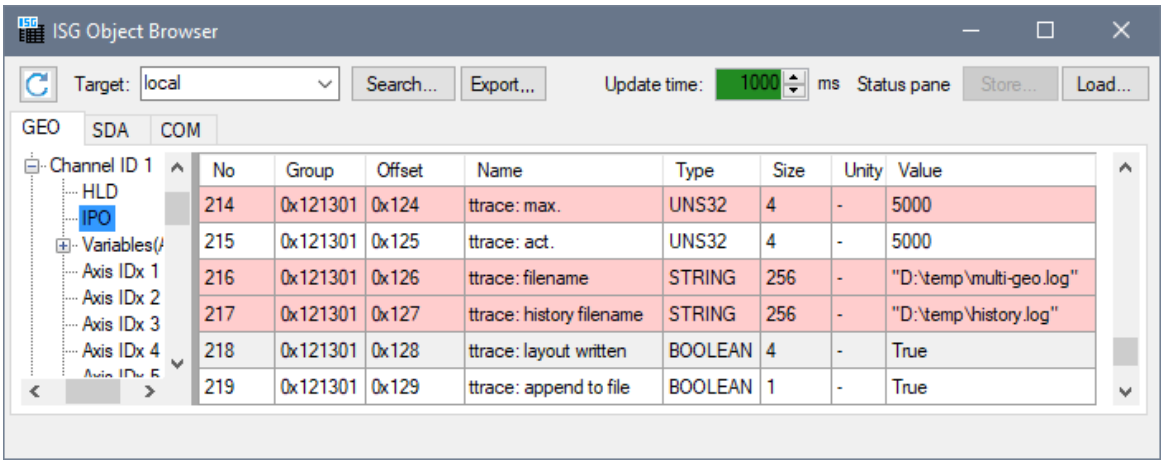


Fig. 16: Available CNC objects in the ISG Object Browser

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. 17: Internal logging format

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

Real-time variables

ID	Parameter	Description
P-RTCF-00017	context	Context information of a thread
P-RTCF-00018	sched- ule 1	Schedule

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	<TwinCATInstallation>\Components\Mc\CNC\Diagnostics\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	<TwinCATInstallation>\Components\Mc\CNC\Diagnostics\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 Real-time parameter [▶ 31].
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 Real-time parameter [▶ 31].
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 Real-time parameter [▶ 31].
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 Real-time parameter

P-RTCF-00015	Assigning the thread to a CPU kernel
Description	The parameter can assign the thread to a specific CPU kernel. The thread is then executed only on this kernel. If the thread is assigned to a kernel that does not exist, error ID 1000180 is output. The operating system then assumes the task of assigning the thread to one or several processor kernels. If the assignment cannot be executed for one reason or another, an error message with ID 1000181 is output.
Parameter	thread[i].cpu
Data type	UNS16
Data range	1 ... 7
Dimension	----
Default value	1
Remarks	Parameterisation example: *thread[0].cpu 1*

P-RTCF-00017	Context information of a thread
Description	When this parameter is specified, the channel can be assigned to a thread (CPU). In order to assign the GEO, SDA or COM task of a channel to the same context, the values of the channel parameter entered in P-CHAN-00410 [▶ 29], P-CHAN-00411 [▶ 30] and P-CHAN-00409 [▶ 29] must be set accordingly.
Parameter	thread[i].context_info
Data type	UNS32
Data range	0 <= thread[i].context_info
Dimension	----
Default value	0
Remarks	Parameterisation example *thread[0].name GEO1* **thread[0].context_info 1** *thread[0].cycle 2000* *thread[0].priority 31 # HIGHEST* *thread[0].error_on_overflow 1* *thread[0].function[0].name task_int* *thread[0].function[0].calls_per_cycle 1* To assign the GEO TASK (P-CHAN-00410) based on the above parameters: **schedule.context.geo 1**

P-RTCF-00018	Schedule
Description	Defines the order in which the actual axis values are read in. Shows the output in which axis command values and interpolation are processed, see CNC scheduler [▶ 12].
Parameter	schedule
Data type	SGN32
Data range	STANDARD COMLETE SWITCHED
Dimension	----
Default value	STANDARD
Remarks	Parameterisation example: *schedule STANDARD*

6.2.4 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 [▶ 28].		
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” [▶ 33] or by P-STUP-00214 [▶ 28].		
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 [▶ 28] .		
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 Appendix

7.1 Suggestions, corrections and the latest documentation

Did you find any errors? Do you have any suggestions or constructive criticism? Then please contact us at documentation@isg-stuttgart.de.

The latest documentation is posted in our Online Help (DE/EN):

QR code link: https://www.isg-stuttgart.de/documentation-kernel/

The link above forwards you to:

https://www.isg-stuttgart.de/fileadmin/kernel/kernel-html/index.html

Notice

Change options for favourite links in your browser;

Technical changes to the website layout concerning folder paths or a change in the HTML framework and therefore the link structure cannot be excluded.

We recommend you to save the above "QR code link" as your primary favourite link.

PDFs for download:

DE:
https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

EN:
https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

Email: documentation@isg-stuttgart.de

Keyword index

P

© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
All rights reserved
www.isg-stuttgart.de
support@isg-stuttgart.de