DOCUMENTATION ISG-kernel

Functional description
RT-Loops

Short Description:
FCT-C42

Documentation version: 1.14
07/11/2024

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

The name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® 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

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

1 Overview

Task

The "real-time loops" function allows the user to influence programmed loops within an NC program during machining.

Effectiveness / possible applications

Real-time loops can be used if the number of loops is not yet known when the NC program is created since the number is determined by process parameters such as the surface quality achieved. Compared to a programmed NC loop, the real-time loop offers the option of moving the loop endlessly in the backward direction immediately after entry.

	Release Note
V200 V260 V300	**This function is available as of CNC Build V3.1.3105.01**

Parameterisation

In order to use the function, P-STUP-00033 [▶ 22], channel parameters P-CHAN-00406 [▶ 22] and P-CHAN-00407 [▶ 23] must be parameterised for the use of real-time cycles.

Programming

The frame of the real-time loops is programmed by the commands #RT WHILE [▶ 15] and #RT ENDWHILE. The condition is programmed by a real-time cycle [▶ 15].

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

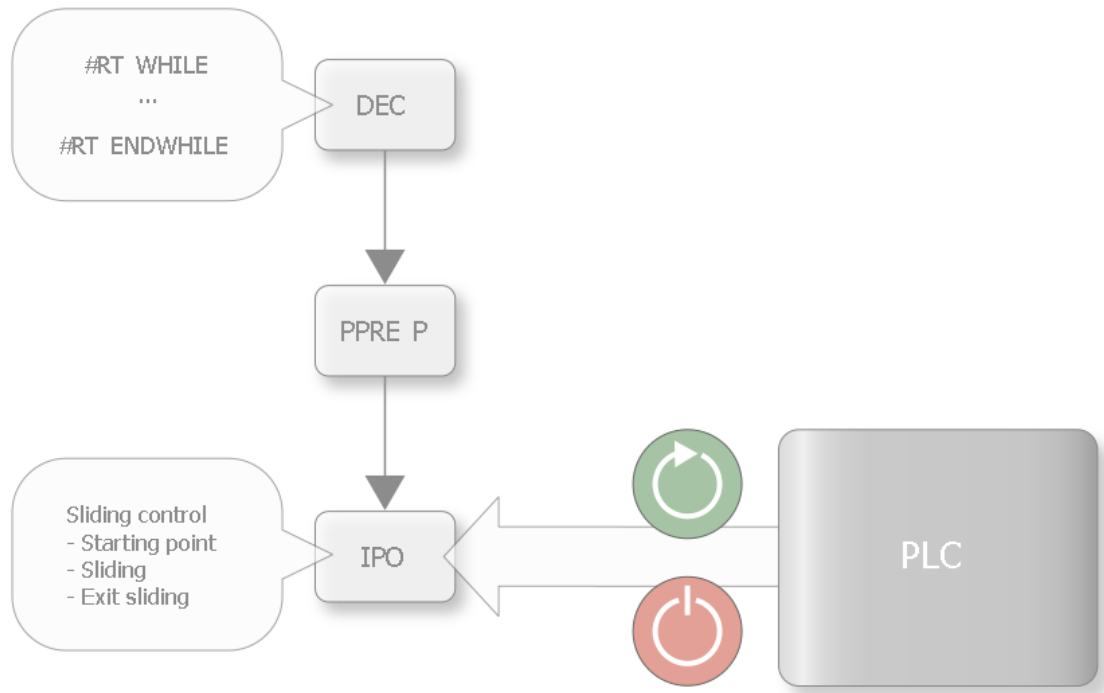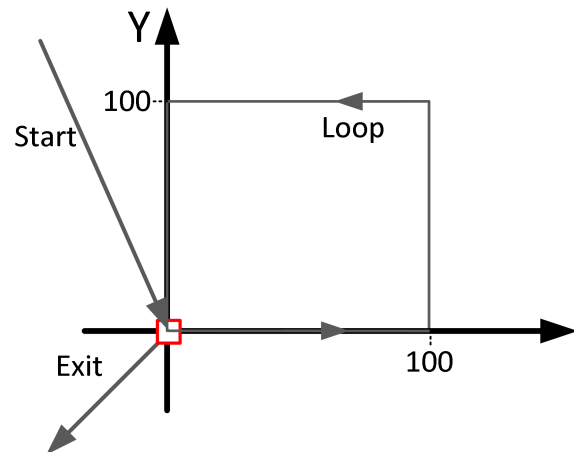Real-time loops allow you to influence the number of loop passes in the NC program during machining.

Fig. 1: Overview

A real-time signal can be used to determine whether to remain in a loop or to exit it at the end of a loop pass.

%RtLoop
N010 X0 Y0 Z0
;-loop : as long as
; V.RTG.LOOP.ENABLED = TRUE
N020 #RT WHILE
N040 X100
N050 Y100
N060 X0
N070 Y0
N090 #RT ENDWHILE
;-----------
N100 X-20 Y-20 Z30
N110 M30

> **ⓘ** **Notice**
>
> **If the real-time variable is not set, the loop is skipped.**

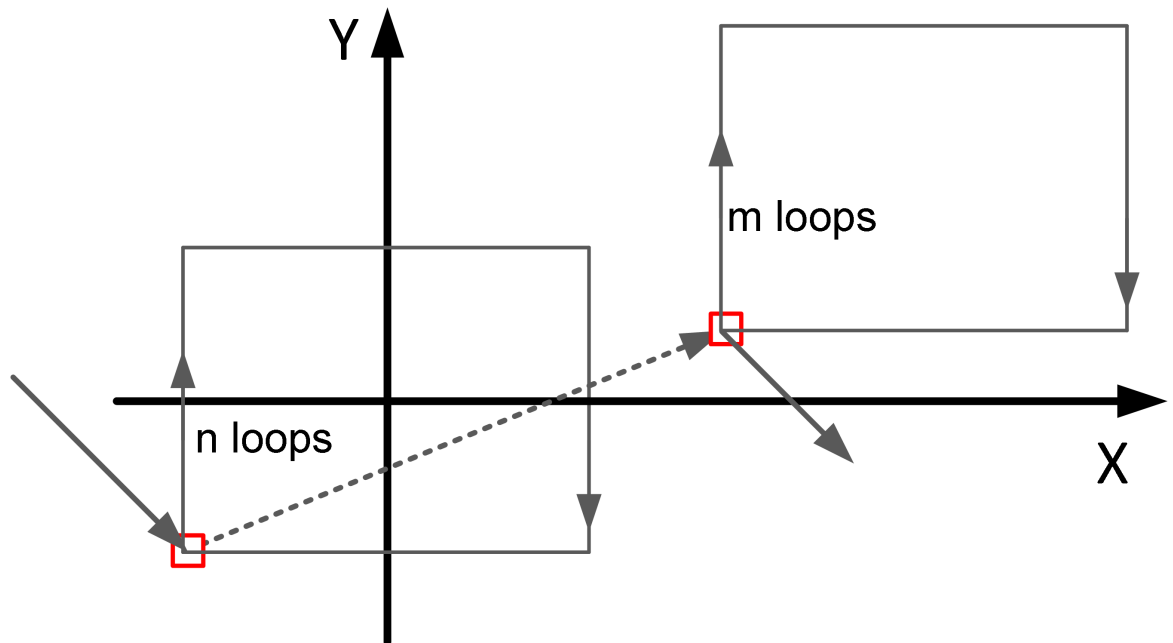It is also possible to program several real-time loops in succession.

Fig. 2: Succession of real-time loops

Definition and behaviour of real-time loops

When programming a real-time loop in the NC program, make absolutely sure that the position in the plane when entering the loop is identical to the position at the end of the loop.

> **ⓘ** **Notice**
>
> **The start point and end point of the contour within a real-time loop must be identical.**
>
> In case of deviations, error ID 50991 is output.

Within the loop you can use the "Backward motion on the path" or "Delete distance to go" functions.

> **ⓘ** **Notice**
>
> **Real-time loops are not permitted with tool radius compensation is active.**
>
> Error ID 22158 is output.

Behaviours with backward motion on the path

With motion backward on the path combined with real-time loops, it is possible to move backward immediately after entering the loop.
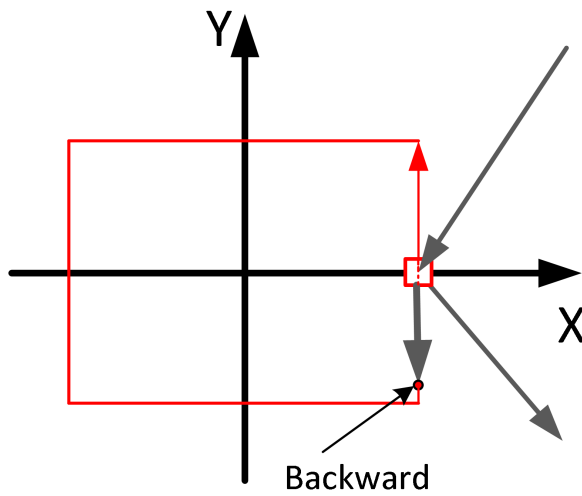
Fig. 3: Direct backward motion immediately after loop entry

It is also possible to enter a loop backwards in an NC program and to complete the loop in the backward direction. The loop can also be exited backwards by setting the real-time variable.
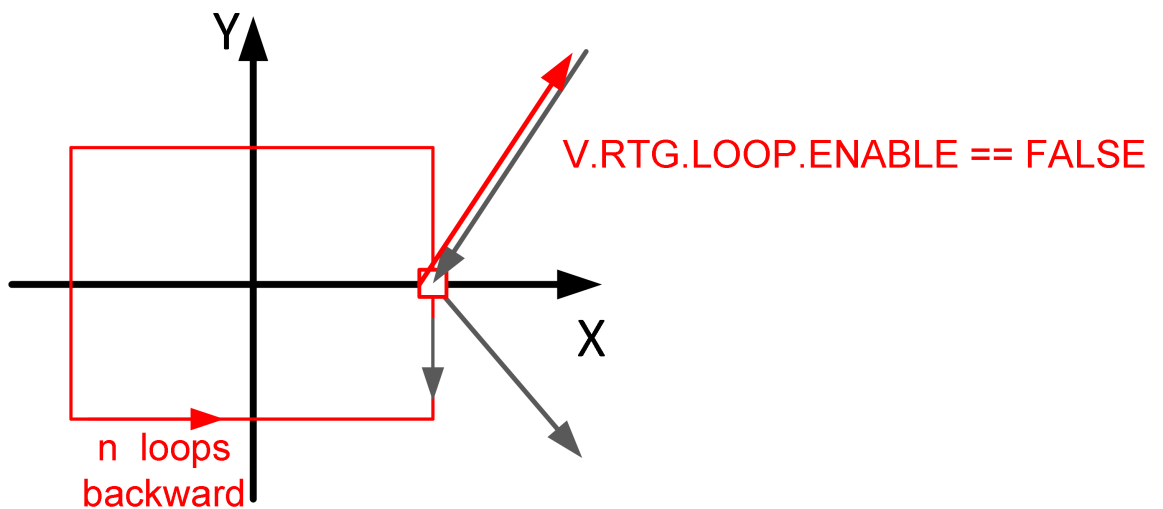
Fig. 4: Existing the loop backwards

Behaviour for deleting distance to go

The "Delete distance to go" control unit can delete the distance to go for a single block in the real-time loop-.
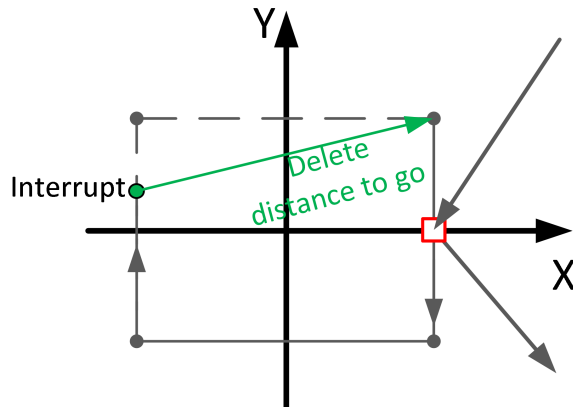
Fig. 5: Delete distance to go for a single block

The function can also be used in the backward direction. The restrictions regarding backward motion in combination with delete distance to go apply.
[FCT-C28// Combination with backward motion]

2.1 Display data

Relevant display data for the "real-time loops" function can be read out via the HLI. The following data is available:

- dist_prog_start [▷ 25]
- block_count_r [▷ 25]
- inside_rt_loop_r [▷ 26]
- rt_loop_count_r [▷ 25]

Programing Example

Differences in display data for #RT WHILE with and without MODULO option

---

The associated display data are displayed on the basis of the following NC program.

```
%RtLoop
N0001 X10
N0002 X30
N0003 X50
N0004 X90
N0005 X130
N0006 #RT WHILE [MODULO]
N0007 X150
N0008 X170
N0009 X200
N0010 X210
N0011 X230
N0012 #RT ENDWHILE
```
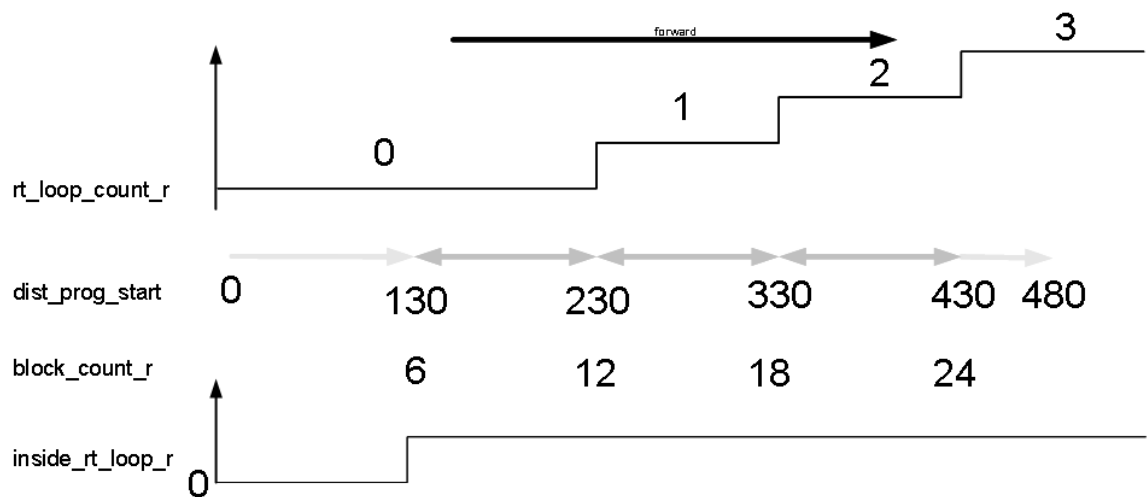

Fig. 6: **Default display data for real-time loops**
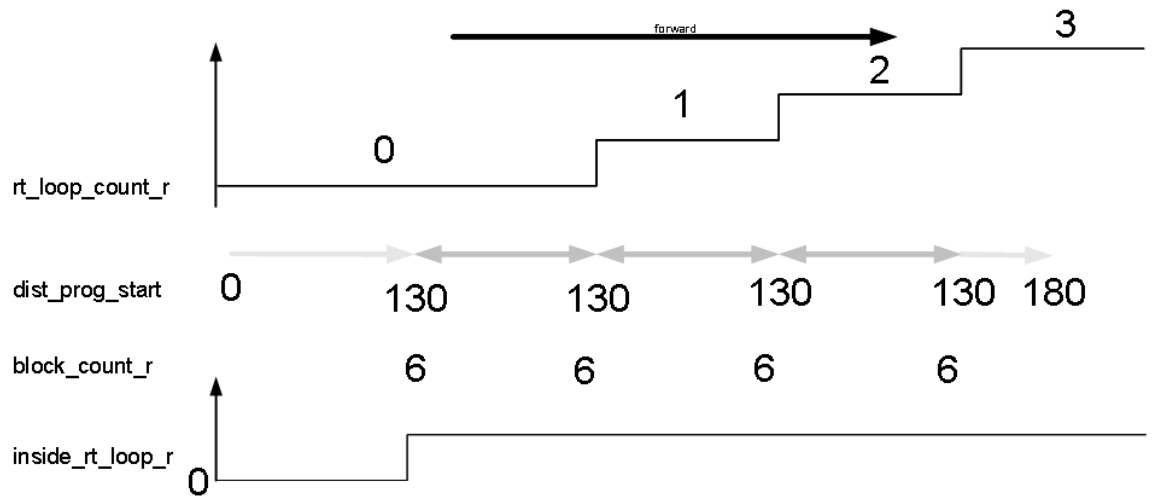
Fig. 7: Display data for real-time loops with MODULO option

2.2 Differences from default loop

Number of loop passes

Default loops usually have a fixed number of loop passes. Real-time loops are endless running loops that are terminated by the variable V.RTG.LOOP.ENABLED. They have no fixed number of loop passes.

Motion stop

With default loops with synchronous variables (e.g. V.E. variable) as loop condition, there is always a motion stop at the loop transition since the current value of the variable must first be determined in the channel. Real-time loops do not have a motion stop at the loop transition, although the condition is evaluated in real time.

Number of loop passes backward direction

A default loop can only be move backward as many times as it previously moved forward. A real-time loop can move backward immediately any number of times until the loop is terminated by the variable V.RTG.LOOP.ENABLED.

2.3   Combination of functions "Insert motion stops during processing" and real-time loops

When combining the "Insert motion stops during machining" function with "Distance" mode and the real-time loops function when using the "MODULO" option, the following special features must be considered:

- An absolute commanded stop is only possible within the current loop or after exiting the loop.
- A relative one-time stop in a future loop that is not reached by a premature exit is rejected.

Absolute commanded stop
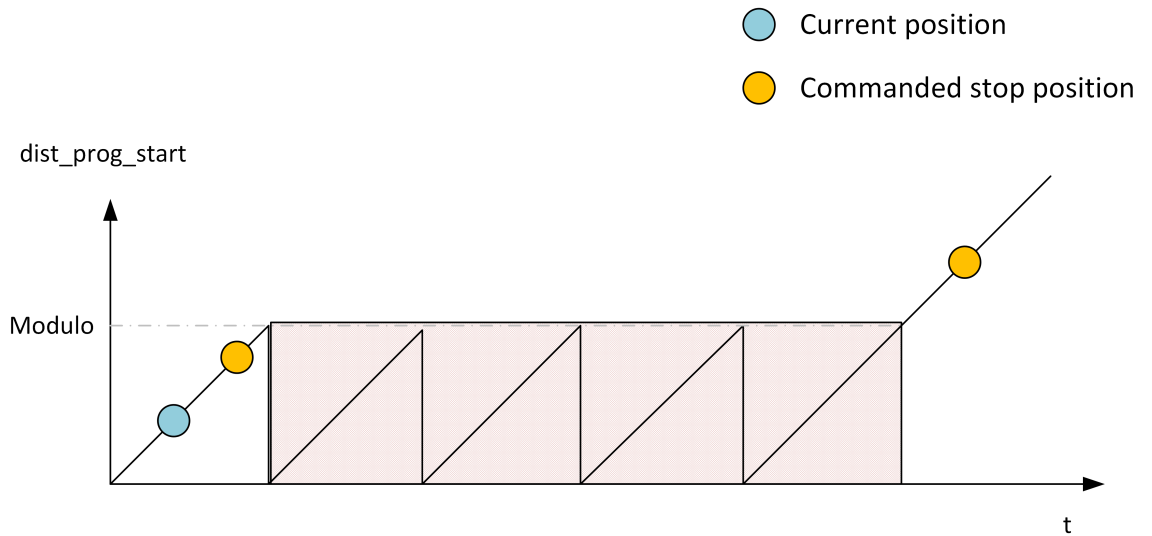
Fig. 8: Absolute commanded stop

Relative one-time/iterative commanded stop
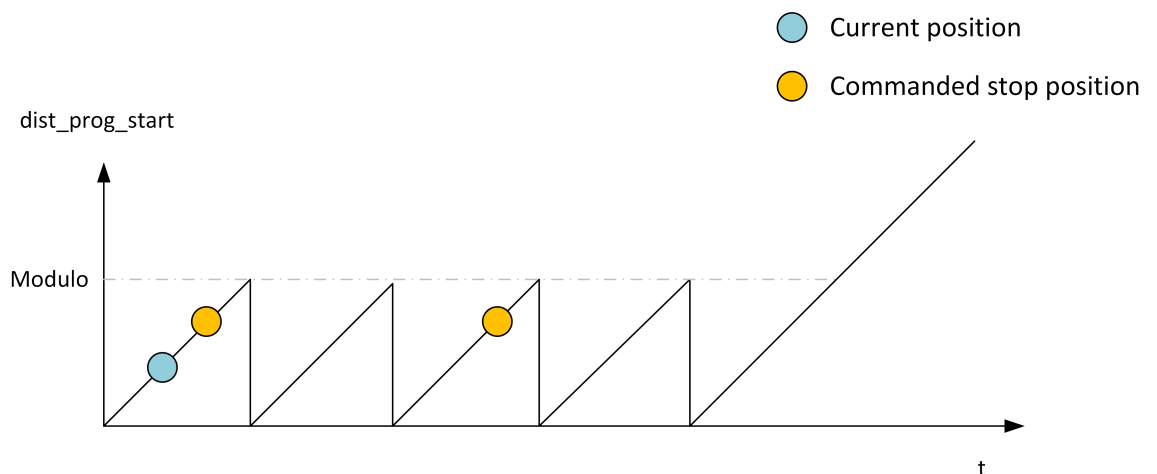
Fig. 9: Relative commanded stop

Difference in case of premature exit from real-time loop
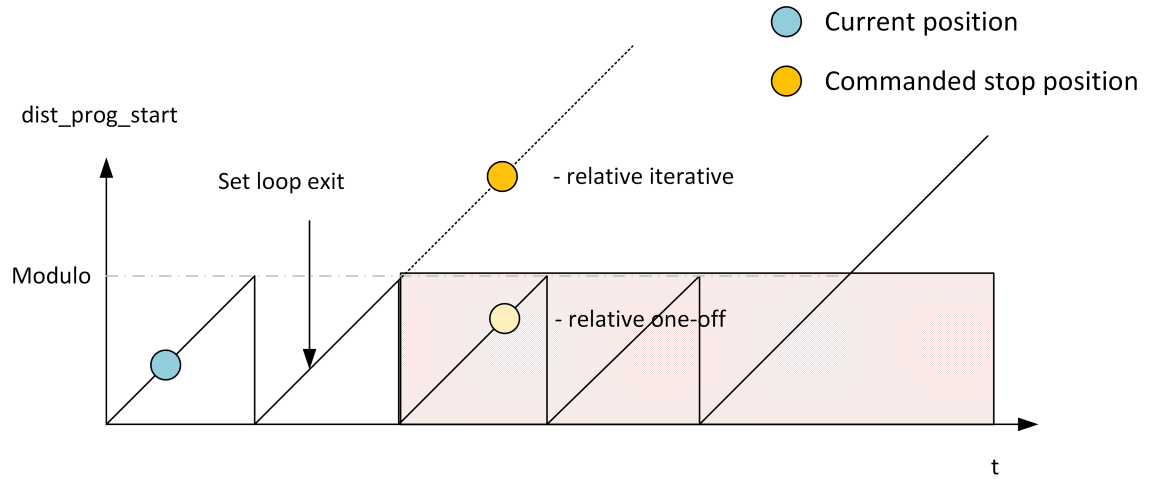
Fig. 10: Premature exit from real-time loop

3 Programming

The loop condition is coupled to the real-time variable V.RTG.LOOP.ENABLED [▶ 19]. The real-time loop is programmed by the following command.

Syntax:

#RT WHILE [**[**MODULO**]**]

#RT ENDWHILE

MODULO   (optional) at the end of the loop, the "dist_prog_start" [▶ 25] function is reset to the start value of the real-time loop. If this parameter is not programmed, this variable is continuously incremented. See Comparison chart in display data [▶ 10].

The loop condition is programmed using the syntax of the real-time cycles.

Syntax:

#RT CYCLE [DEF] [[ID =..**] SCOPE =.. END_ACTION =..]**

;…Instructions

#RT CYCLE END

DEF   Realtime cycle is only defined but not started yet.
ID=..   Unambiguous identifier of the realtime cycle, see Note.
SCOPE   For validity, see Validity.

Permissible identifiers are **BLOCK**, **PROG** or **GLOBAL**.
END_ACTION   Behaviour at end of the real-time cycle, see Action at end.

Permissible identifiers are **MOVE_ABORT** or **MOVE_CONT**.

In the following programming example, the loop condition is coupled to the value of the external variable V.E.RtLoopEnable in a real-time cycle.

In the parameter P-EXTV-00005 [▶ 24] **the external variable must be parameterised as synchronous.**

Programing Example

Real-time loop with external variable as loop condition

```
%RtLoop
; RT WHILE condition
N0010 #RT CYCLE DELETE [ID = 4711 ]
N0020 #RT CYCLE [ID=4711 SCOPE = GLOBAL]
N0030   $IF V.E.RtLoopEnable != 0
N0040     V.RTG.LOOP.ENABLED = TRUE
N0050   $ELSE
N0060     V.RTG.LOOP.ENABLED = FALSE
N0070   $ENDIF
N0080 #RT CYCLE END

; start movement
N0090 X0 Y0 Z0
N0100 #BACKWARD STORAGE CLEAR
N0110 #DISTANCE PROG START CLEAR
N0120 Z33 G01 F100
N0130 Z0
;------------
; while until V.E.RtLoopEnable == 0
N0140 #RT WHILE
N0150 X100
N0160 Y100
N0170 X0
N0180 Y0
N0190 #RT ENDWHILE
;------------
N0200 X10
N0210 Y20
N0220 Z30
N0230 M30
```

Programing Example

Real-time loop with variable geometry

This example shows a real-time loop which has a variable geometry due to the command #OPTIONAL EXECUTION ON/OFF [SIMULATE]. The geometry in #OPTIONAL EXECUTION ON/OFF [SIMULATE] can be enabled or disabled by the simulate_motion control unit during machining. It is also possible to enable other NC commands such as M functions with #OPTIONAL EXECUTION ON/OFF [SIMULATE].

```
%RtLoop
; RT WHILE condition
N0010 #RT CYCLE DELETE [ID = 4711 ]
N0020 #RT CYCLE [ID=4711 SCOPE = GLOBAL]
N0030 $IF V.E.RtLoopEnable != 0
N0040    V.RTG.LOOP.ENABLED = TRUE
N0050   $ELSE
N0060    V.RTG.LOOP.ENABLED = FALSE
N0070   $ENDIF
N0080 #RT CYCLE END

; start movement
N0090 X0 Y0 Z0
N0100 #BACKWARD STORAGE CLEAR
N0110 #DISTANCE PROG START CLEAR
N0120 Z33 G01 F100
N0130 Z0
;-----------
; while until V.E.RtLoopEnable == 0
N0140 #RT WHILE
N0150 X100
N0160 Y100
N0170 #OPTIONAL EXECUTION ON [SIMULATE]
N0180 Z100
N0190 Z0
N0200 #OPTIONAL EXECUTION OFF
N0210 X0
N0220 Y0
N0230 #RT ENDWHILE
;-----------
N0240 X10
N0250 Y20
N0260 Z30
N0270 M30
```

3.1 Permissible NC commands within a real-time loop

Only the following NC commands are permitted within a real-time loop.

- Motion blocks G00, G01, G02, G03
- #SIGNAL SYN / #WAIT SYN
- Write an external variable V.E. =
- #OPTIONAL EXECUTION [SIMULATE]
- #CHANNEL SET [FAST_FORWARD_IN_CENTER=]
- #SPLINE ON/OFF
- #STOP REVERSIBLE

---

i | **Notice**

No spline mechanism may be active when #RT WHILE/ #RT ENDWHILE is programmed.

If the spline mechanism is active, #RT WHILE [▶ 15] will result in error with ID 22128; #RT END-WHILE [▶ 15] will result in error with ID 22129.

---

i | **Notice**

Only the NC commands listed are permissible.

Error ID 22073 is output if other NC commands are programmed.

3.2   Real-time variables for real-time loops

The following real-time variables are available for the real-time loop:

- V.RTG.LOOP.ENABLED - loop condition for a real-time loop
- V.RTG.LOOP.COUNT – number of executed real-time loops

3.3   Contour look-ahead and real-time loops

There are two programming options to obtain contour elements in a real-time loop (#RT WHILE/ ENDWHILE [▶ 15]) in Mode 1:

1. Only one geometry is provided when programming #CONTOUR LOOKAHEAD LOG ON/OFF within a real-time loop [▶ 5]. It has the properties of the entry loop, which means the parameter "tangent_variation" has the angle to the motion block before the loop.

2. If #CONTOUR LOOKAHEAD LOG ON/OFF is programmed outside the real-time loop [▶ 5], the elements are supplied three times to obtain all the properties of the possible transitions. The transitions are:

    – start of loop

    – loop-loop

    – end of loop

!   **Attention**

Do not program the NC command #CONTOUR LOOKAHEAD CLEAR within a real-time loop.

If the command is programmed within the real-time loop, error ID 22073 is output.

Programing Example

Behaviour of contour look-ahead within a real-time loop

In this programming example, the contour look-ahead function is programmed within a real-time loop. The contour of the motion blocks 80 - 120 is logged once.

```
N060 #RT WHILE
N070 #CONTOUR LOOKAHEAD LOG ON [PARAM=1]
N080 G1 X4 Y4
…
N120 G1 X0 Y0
N130 #CONTOUR LOOKAHEAD LOG ON [PARAM=8]
N140 #RT ENDWHILE
```

Programing Example

Behaviour of contour look-ahead outside a real-time loop

In this programming example, the contour look-ahead is programmed outside a real-time loop. The contour of the motion blocks 80 to 120 is logged three times.

```
N060 #CONTOUR LOOKAHEAD LOG ON [PARAM=1]
N070 #RT WHILE
N080 G1 X4 Y4
…
N120 G1 X0 Y0
N130 #RT ENDWHILE
N140 #CONTOUR LOOKAHEAD LOG ON [PARAM=8]
```

4   Parameter

4.1   Overview

restart

ID	Parameter	Description
P-STUP-00033	fb_storage_size	Memory size for backward motion

Channel parameters

ID	Parameter	Description
P-CHAN-00406	rt_cycles.enable	Enable real-time cycle function
P-CHAN-00407	rt_cycles.memory	Memory size for real-time cycles
P-CHAN-00425	rt_cycles.max_duration	Max. execution time of real-time cycles per CNC cycle
P-CHAN-00426	rt_cycles.cont_steps	Number of elementary instructions for time check

External variables

ID	Parameter	Description
P-EXTV-00005	synchronisation	Synchronisation type of an external variable

4.2   Description

Start-up parameters

P-STUP-00033	Memory size for backward motion
Description	This parameter defines the memory size in bytes used for backward motion on the path. During start-up, the NC checks whether the required minimum size is available. If this is not the case, a warning is output and the memory size is set to the required minimum value. If the size is set to 0, the "forward/ backward motion on the path" function is not available. The maximum size is only limited by the resources available on the PC.
Parameter	fb_storage_size[i] where i = 0 to 11 (maximum number of channels: 12, application-specific)
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

Channel parameters

P-CHAN-00406	Activating real-time cycles
Description	This parameter enables the real-time cycle function in the NC channel. The controller must be restarted to adopt the change. Example: `configuration.rt_cycles.enable 1`
Parameter	configuration.rt_cycles.enable
Data type	BOOLEAN
Data range	0/1
Dimension	----
Default value	0
Remarks	This parameter is available as of CNC Build V3.1.3107.10 or higher. Use of the parameter "rt_cycles.enable" `rt_cycles.enable 1` (as of V3.1.3105) continues to be supported.

P-CHAN-00407	Memory size for real-time cycles
Description	This parameter defines the size of the memory for real-time cycles. The size of the memory is specified in bytes. The controller must be restarted to adopt the change. Then the specified memory is also available for real-time cycles. Example: `configuration.rt_cycles.memory 60000`
Parameter	configuration.rt_cycles.memory
Data type	UNS32
Data range	0 ... MAX(UNS32) - 1
Dimension	----
Default value	48000
Remarks	**Note:** The assignment of P-CHAN-00407 is only necessary if the memory set by default is no longer sufficient due to activation of the real-time cycles (P-CHAN-00406 [▶ 22]). This parameter is available as of CNC Build V3.1.3107.10 or higher. Use of the parameter "rt_cycles.memory" `rt_cycles.memory 60000` (as of V3.1.3105) continues to be supported.

P-CHAN-00425	Max. execution time of realtime cycles per CNC cycle
Description	This parameter defines the maximum execution time of realtime cycles in the NC channel. The parameter is specified in percent (%) and refers to the length of the CNC cycle. Example: If the realtime task of the CNC is clocked at 2ms and parameter P-CHAN-00425 is 75, the realtime cycles may require a total execution time of maximum 1.5ms. If this time is exceeded, error ID 50939 is output.
Parameter	rt_cycles.max_duration
Data type	UNS16
Data range	0 < P-CHAN-00425 < MAX_UNS16
Dimension	%
Default value	75
Remarks	The user has no restrictions regarding the number of instructions within a realtime cycle. Realtime timeouts will occur if realtime cycles contain too many instructions and cannot be executed in one CNC cycle, The parameters P-CHAN-00426 and P-CHAN-00427 represent a safety mechanism to prevent such realtime timeouts as early as possible.

P-CHAN-00426	Number of elementary instructions for time check
Description	This parameter defines the number of elementary instructions after which another time check is executed. A check for the execution time of realtime cycles must be made within a CNC cycle to see whether the permitted execution time is already exceeded. This takes place by checking the expired time after a specific number of elementary instructions in a cycle. Parameter P-CHAN-00426 indicates the number of these elementary instructions.
Parameter	rt_cycles.cont_steps
Data type	UNS32
Data range	0 < P-CHAN-00426 < MAX_UNS32
Dimension	----
Default value	100
Remarks	The user has no restrictions regarding the number of instructions within a realtime cycle. Realtime timeouts will occur if realtime cycles contain too many instructions and cannot be executed in one CNC cycle, The parameters P-CHAN-00425 and P-CHAN-00427 represent a safety mechanism to prevent such realtime timeouts as early as possible.

External variables

P-EXTV-00005	Synchronisation type of external variable
Description	Write/read access is normally performed synchronously with processing. In individual cases, this implicit synchronisation can be suppressed (see Access synchronisation by NC channel). If the variable is a variable structure, all subordinated structure elements inherit the synchronisation type. In addition, the synchronisation type can be individually defined for each structure element in the type definition (P-EXTV-00018).
Parameter	var[i].synchronisation
Data type	BOOLEAN
Data range	TRUE, FALSE
Dimension	----
Default value	TRUE
Remarks	Synchronous accesses always cause the decoder to stop during a read operation until the synchronous read value is available to the decoder. In addition, it is not permitted to read synchronous variables, e.g. with functions such as an enabled Tool radius compensation; the message is output with ID 20651.

4.3 PLC interface for display data

Currently covered path in the NC program (PCS)	
Description	Reads the current distance covered in the NC program since program start or since the last # DISTANCE PROG START CLEAR NC command. The calculation is based on the current position in the current NC block.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**dist_prog_start**
Data type	UDINT (* LREAL)
Unit	0.1 µm
Access	PLC is reading
Special features	* As of CNC Build V3.1.3104.01 the data element is provided in LREAL forma.

Line counter, NC program	
Description	The datum indicates the NC program line which is the source of the command just processed by the interpolator.
	The value is derived from the number of NC program lines which the decoder has read since the NC program started. All the lines read the decoder are counted, i.e. repeatedly read lines, empty and comment lines. All commands to the interpolator resulting from decoding a NC program line are assigned to the associated line counter.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.block_count_r
Data type	UDINT
Access	PLC is reading

Numerator of the real-time loop	
Description	The datum shows the number of passes of the real-time loop in which the current machining is located.
	If a loop transition is in a forward direction, the datum is incremented by 1 and it is decremented in a backward direction.
	If more loop passes were processed in the backward direction, a negative numeral is displayed.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**rt_loop_count_r**
Data type	DINT
Access	PLC reads
Special feature	Date available as of CNC Build V3.1.3105.01 and higher

Real-time loop active	
Description	The datum indicates whether current machining is located in a #RT WHILE - #RT ENDWHILE sequence.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**inside_rt_loop_r**
Data type	BOOL
Value range	[TRUE = Machining within #RT WHILE #RT ENDWHILE, FALSE]
Access	PLC is reading
Special feature	Date available as of CNC Build V3.1.3105.01 and higher

5 Appendix

5.1 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

---

i	**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

E-Mail: documentation@isg-stuttgart.de

Keyword index