



# DOKUMENTATION ISG-kernel

## Handbuch Konfigurationsliste der Echtzeitparameter

Kurzbezeichnung:  
RTCF

© Copyright  
ISG Industrielle Steuerungstechnik GmbH  
STEP, Gropiusplatz 10  
D-70563 Stuttgart  
Alle Rechte vorbehalten  
[www.isg-stuttgart.de](http://www.isg-stuttgart.de)  
[support@isg-stuttgart.de](mailto:support@isg-stuttgart.de)

Dokumentation Version: 1.07  
12.12.2023

# Vorwort

## Rechtliche Hinweise

---

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte und der Funktionsumfang werden jedoch ständig weiterentwickelt. Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

## Qualifikation des Personals

---

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen, der zugehörigen Dokumentation und der Aufgabenstellung vertraut ist.

Zur Installation und Inbetriebnahme ist die Beachtung der Dokumentation, der nachfolgenden Hinweise und Erklärungen unbedingt notwendig. Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zum betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbarer Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

## Weiterführende Informationen

---

Unter den Links (DE)

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

bzw. (EN)

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

finden Sie neben der aktuellen Dokumentation weiterführende Informationen zu Meldungen aus dem NC-Kern, Onlinehilfen, SPS-Bibliotheken, Tools usw.

## Haftungsausschluss

---

Änderungen der Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig.

## Marken und Patente

---

Der Name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® und entsprechende Logos sind eingetragene und lizenzierte Marken der ISG Industrielle Steuerungstechnik GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltene Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

## Copyright

---

© ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster oder Geschmacksmustereintragung vorbehalten.

# Allgemeine- und Sicherheitshinweise

## Verwendete Symbole und ihre Bedeutung

In der vorliegenden Dokumentation werden die folgenden Symbole mit nebenstehendem Sicherheitshinweis und Text verwendet. Die (Sicherheits-) Hinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

## Symbole im Erklärtext

- Gibt eine Aktion an.
- ⇒ Gibt eine Handlungsanweisung an.



### **GEFAHR**

#### **Akute Verletzungsgefahr!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!



### **VORSICHT**

#### **Schädigung von Personen und Maschinen!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen und Maschinen geschädigt werden!



### **Achtung**

#### **Einschränkung oder Fehler**

Dieses Symbol beschreibt Einschränkungen oder warnt vor Fehlern.



### **Hinweis**

#### **Tipps und weitere Hinweise**

Dieses Symbol kennzeichnet Informationen, die zum grundsätzlichen Verständnis beitragen oder zusätzliche Hinweise geben.



### **Beispiel**

#### **Allgemeines Beispiel**

Beispiel zu einem erklärten Sachverhalt.



### **Programmierbeispiel**

#### **NC-Programmierbeispiel**

Programmierbeispiel (komplettes NC-Programm oder Programmsequenz) der beschriebenen Funktionalität bzw. des entsprechenden NC-Befehls.



### **Versionshinweis**

#### **Spezifischer Versionshinweis**

Optionale, ggf. auch eingeschränkte Funktionalität. Die Verfügbarkeit dieser Funktionalität ist von der Konfiguration und dem Versionsumfang abhängig.

# Inhaltsverzeichnis

<b>Vorwort</b> .....	<b>2</b>
<b>Allgemeine- und Sicherheitshinweise</b> .....	<b>3</b>
<b>1 Übersicht Echtzeitparameter</b> .....	<b>6</b>
<b>2 Allgemeine Beschreibung</b> .....	<b>8</b>
2.1 Verweise auf andere Dokumente .....	8
2.2 Verwendung der rtconf.lis .....	8
2.3 Struktur und Gliederung der RT-Konfigurationsdaten .....	9
2.4 Syntax und Interpretation der ASCII-Listendatei .....	10
2.5 Kommentare in der ASCII-Listendatei .....	11
<b>3 Beschreibung der grundlegenden Parameter</b> .....	<b>12</b>
3.1 Quelle des Interrupts (P-RTCF-00001) .....	13
3.2 NC-Zykluszeit (P-RTCF-00002) .....	13
3.3 Timer-Interrupt (P-RTCF-00003) .....	14
3.4 Garantierte Windows-Rechenzeit (P-RTCF-00004) .....	15
3.5 Name des externen Betriebssystem-Semaphore (P-RTCF-00005) .....	15
<b>4 Thread-spezifische Parameter (thread[i].*)</b> .....	<b>17</b>
4.1 Name des Threads (P-RTCF-00008) .....	17
4.2 Zykluszeit des Threads (P-RTCF-00009) .....	17
4.3 Aufrufreihenfolge des Threads (P-RTCF-00010) .....	18
4.4 Fehlermeldung bei Überlauf des Threads (P-RTCF-00011) .....	18
4.5 Funktionen in einem Thread (thread[i].function[j].*) .....	19
4.5.1 Name einer Thread-Funktion (P-RTCF-00012) .....	19
4.5.2 Anzahl Aufrufe einer Thread-Funktion (P-RTCF-00013) .....	19
4.5.3 Markierungsbit einer Thread-Funktion (P-RTCF-00014) .....	20
4.5.4 Zuweisung des Threads an einen CPU-Kern (P-RTCF-00015) .....	20
4.6 Kontextinformation eines Threads (P-RTCF-00017) .....	21
4.7 Schedule (P-RTCF-00018) .....	21
<b>5 Externe Threads</b> .....	<b>22</b>
5.1 Allgemeine Beschreibung und Überblick .....	22
5.2 Spezifische Parameter externer Threads (external_thread[i].*) .....	23
5.2.1 Zykluszeit eines externen Threads (P-RTCF-00006) .....	23
5.2.2 Name der Semaphore eines externen Threads (P-RTCF-00007) .....	23
5.2.3 Beispiel einer Rtconf.lis für eine SERCOS Applikation .....	24
<b>6 Anhang</b> .....	<b>26</b>
6.1 Anregungen, Korrekturen und neueste Dokumentation .....	26
<b>Stichwortverzeichnis</b> .....	<b>27</b>

## Abbildungsverzeichnis

Abb. 1:	Scheduling der CNC und eines externen Threads .....	8
Abb. 2:	Grundlegende Parameter des Scheduling .....	12
Abb. 3:	Synchronisation mit einem externen Thread .....	22

# 1 Übersicht Echtzeitparameter



## Achtung

**Dieses Dokument ist für TwinCAT-Anwender nicht relevant.  
In TwinCAT-Systemen werden die erforderlichen Echtzeitparameter über den System Manager parametrierd.**

### Die Übersicht der Echtzeitparameter ist tabellarisch in 4 Spalten sortiert

- In der 1. Spalte steht die eindeutige Kennung des Echtzeitparameter, die sog. "ID". Diese setzt sich aus dem Präfix "P-RTCF" und einer eindeutigen 5-stelligen Nummer zusammen, z.B. P-RTCF-00002.
- In der 2. Spalte ist die Datenstruktur dargestellt, in der der Parameter definiert ist, z.B. thread[i].  
Die Struktur dient der Kategorisierung, welche sich folgend im Kapitelaufbau widerspiegelt. Wenn bei 'Struktur' die Angabe fehlt, ist dies kein Fehler; in dem Fall gilt nur der Parameter in Spalte 3 alleine.
- In der 3. Spalte findet sich der "Parameter" mit seiner genauen Bezeichnung, z.B. priority  
Wichtig zu erwähnen ist, dass "Struktur"+"Parameter" immer zusammen gehören und exakt so in der Liste der Echtzeitparameter konfiguriert werden müssen, z.B. thread[i].priority
- In der 4. Spalte wird die "Funktionalität" in einem zusammenfassenden Begriff/Kurzbeschreibung dargestellt, z.B. Priorität/ Aufrufreihenfolge des Threads.

ID	Struktur	Parameter	Funktionalität/ Kurzbeschreibung
P-RTCF-00001 [▶ 13]		interrupt_source	Quelle des Interrupts
P-RTCF-00002 [▶ 13]		cycle_time	NC-Zykluszeit
P-RTCF-00003 [▶ 14]		time_slice	Timer-Interrupt
P-RTCF-00004 [▶ 15]		windows_time	Garantierte Windows-Rechenzeit
P-RTCF-00005 [▶ 15]		external_object_name	Name des externen Betriebssystem-Semaphore
P-RTCF-00006 [▶ 23]	external_thread[i].	cycle	Externer Threads- Zykluszeit
P-RTCF-00007 [▶ 23]	external_thread[i].	semaphore_name	Externer Threads- Name des Semaphore
P-RTCF-00008 [▶ 17]	thread[i].	name	Name des Threads
P-RTCF-00009 [▶ 17]	thread[i].	cycle	Zykluszeit des Threads
P-RTCF-00010 [▶ 18]	thread[i].	priority	Aufrufreihenfolge des Threads innerhalb des Scheduling
P-RTCF-00011 [▶ 18]	thread[i].	error_on_overflow	Fehlermeldung bei Überlauf des Threads
P-RTCF-00012 [▶ 19]	thread[i].function[j].	name	Name der Funktion innerhalb eines Threads
P-RTCF-00013 [▶ 19]	thread[i].function[j].	calls_per_cycle	Anzahl Funktionsaufrufe innerhalb eines Threads
P-RTCF-00014 [▶ 20]	thread[i].function[j].	trace_bit	Markierungen für Diagnose
P-RTCF-00015 [▶ 20]	thread[i].	cpu	Zuweisung des Threads an einen CPU-Kern
P-RTCF-00017 [▶ 21]	thread[i].	context	Kontextinformation eines Threads
P-RTCF-00018 [▶ 21]	thread[i].	schedule	Schedule

## 2 Allgemeine Beschreibung

### 2.1 Verweise auf andere Dokumente

Es wird zwecks Übersichtlichkeit eine verkürzte Darstellung der Verweise (Links) auf andere Dokumente bzw. Parameter gewählt, z.B. [PROG] für Programmieranleitung oder P-AXIS-00001 für einen Achsparameter.

Technisch bedingt funktionieren diese Verweise nur in der Online-Hilfe (HTML5, CHM), nicht allerdings in PDF-Dateien, da PDF keine dokumentenübergreifende Verlinkungen unterstützt.

### 2.2 Verwendung der `rtconf.lis`

Die RT-Konfigurationsliste findet Verwendung, wenn das Betriebssystem Windows mit einem thread-basierten Echtzeitbetriebssystem (z.B. VxWorks, RTX) auf einer gemeinsamen Plattform betrieben wird.

Durch die Parameter der `rtconf.lis` lässt sich ein Scheduling mit folgenden Leistungsmerkmalen parametrieren:

- Skalierbares Scheduling zur effizienten Nutzung der CPU-Leistung.
- Fest vorzugebende CPU-Aufteilung zwischen Windows und Echtzeitbetriebssystem.
- Aufrufhäufigkeit jeder Thread pro Zyklus.
- Reduzierung der Aufrufhäufigkeit eines Threads (Modulo Zykluszeit).
- Einbindung externer Threads (z. B. eines SPS -Runtimesystems).

Das folgende Bild zeigt ein Scheduling mit einem externen Interrupt und einem internen Timer-Interrupt zur Steuerung der Rechenzeit der einzelnen Threads.

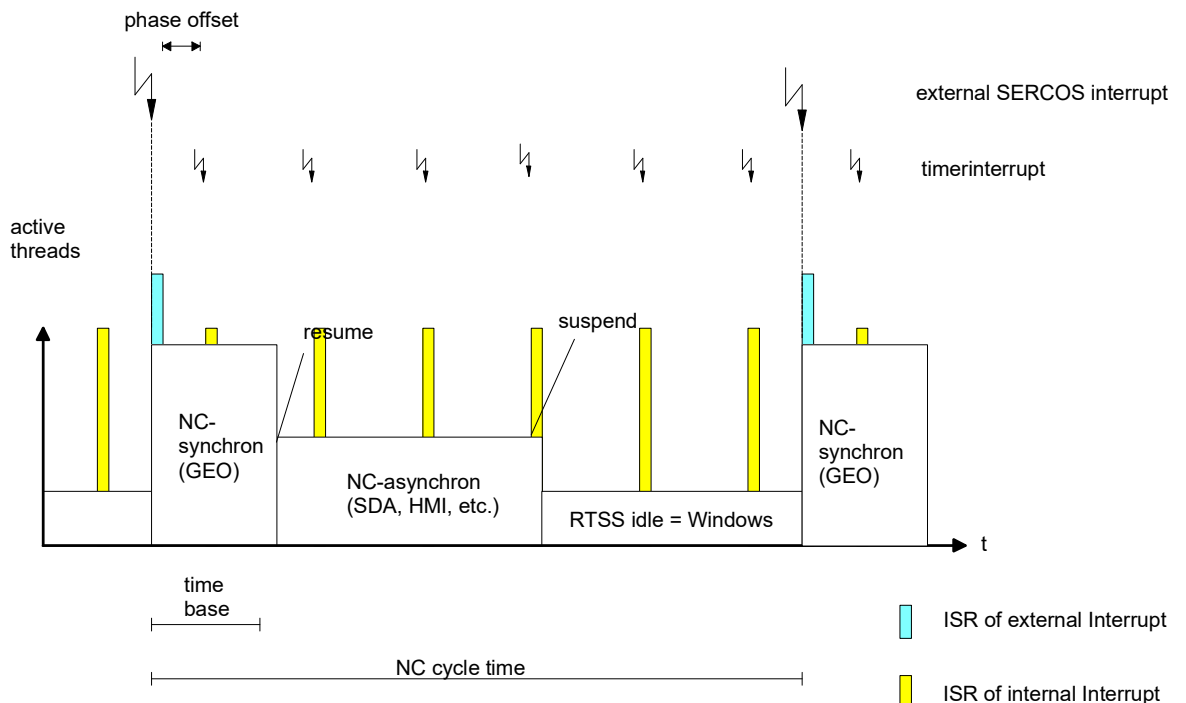


Abb. 1: Scheduling der CNC und eines externen Threads



Nach dem externen Interrupt werden alle am Scheduling beteiligten Threads ausführungsbereit gesetzt. Danach richtet sich die Ausführungsreihenfolge der Threads ausschließlich nach deren Priorität. Diese wird ebenfalls in der RT-Konfigurationsliste parametrieren. In der Regel wird der synchrone NC - Bearbeitungsteil (GEO) am höchsten priorisiert und kommt somit zuerst zur Ausführung. Danach folgt der asynchrone Thread bestehend z.B. aus den Funktionsaufrufen für Steuerdatenaufbereitung (SDA), HMI und Systemablaufsteuerung.

Die Funktionen des asynchronen Threads werden so oft in einer Schleife aufgerufen wie dies in der RT-Konfigurationsliste parametrieren ist. Allerdings gibt es eine Obergrenze für die Ausführungszeit des asynchronen Threads. Diese wird mittels des höher auflösenden Timer-Interrupts überwacht. D.h. der asynchrone Thread wird ggf. suspendiert damit Windows die (ebenfalls in rt-conf.lis) parametrieren Rechenzeit erhält.

## 2.3 Struktur und Gliederung der RT-Konfigurationsdaten

Die Parameter der **Real-Time-Configuration Liste (rtconf.lis)** sind hierarchisch gegliedert. Sie bestehen aus:

Gliederungskriterium	Inhalt
Allgemeine Parameter	Grundlegende Parameter für das gesamte Scheduling
Thread-spezifische Parameter	Angaben zur Parametrierung pro Thread
Parameter pro Funktion innerhalb eines Threads	Angaben zur Parametrierung der Funktionsaufrufe innerhalb eines Threads.
Externe Threads	Parameter für externe Threads

Software-Intern sind die Parameter der **rtconf.lis** in folgenden C-Strukturen definiert:

Struktur	Definition im File:	String im ASCII-File:	Inhalt
RTCONF_DATA	rt_conf.inc	-	Parameter für das Scheduling
CNC_THREAD_PROPERTIES	rt_conf.inc	thread[i]	Thread-spezifische Parameter
CNC_FUNCTION_PROPERTIES	rt_conf.inc	thread[i].function[j]	funktionsspezifische Parameter pro Thread
CNC_EXTERNAL_THREAD_PROPERTIES	rt_conf.inc	external_thread[i]	spezifische Parameter für externen Thread

## 2.4 Syntax und Interpretation der ASCII-Listendatei

Die in der ASCII-Listendatei enthaltenen Einträge werden von einem Interpreter in die entsprechenden internen Strukturen übernommen und danach auf Plausibilität geprüft. Damit ein sicherer Hochlauf der Steuerung immer gewährleistet ist, werden die bei der Plausibilitätsprüfung festgestellten fehlerhaften Einträge durch Standardwerte ersetzt.

Unbekannte Einträge werden nicht übernommen. Diese Unregelmäßigkeiten werden durch Warnmeldungen angezeigt. Es wird empfohlen, diesen Warnmeldungen nachzugehen und fehlerhafte Einträge in der ASCII-Listendatei zu bereinigen!



### Hinweis

Für Daten vom Typ BOOLEAN gilt folgende Vereinbarung:

Wert	Bedeutung
0	Definition von FALSE
1	Definition von TRUE

## 2.5 Kommentare in der ASCII-Listendatei

Kommentare können ganzzeilig oder am Ende einer Zeile eingefügt werden.

Bei ganzzeiligem Kommentar muss am Zeilenanfang das Kommentarzeichen '#' gefolgt von einem Leerzeichen eingefügt werden.

Soll am Ende einer Zeile ein Kommentar eingefügt werden, so muss vor dem Kommentar ein Leerzeichen vorhanden sein. Wurde in der Zeile jedoch ein String definiert, so muss dem Kommentar das Kommentarzeichen '(' vorangestellt werden.

Leerzeilen sind ebenfalls möglich.



### Beispiel

Kommentare in ASCII-Listendatei

```
#
*****
# Daten
#
*****
#
# Auflistung

dummy[1] 1 Kommentar
dummy[2] 1 # Kommentar
dummy[3] 1 ( Kommentar
dummy[4] 1 /* Kommentar
...
...
beispiel[0].bezeichnung STRING_2 (Kommentar, hier Kommentarklammer nötig!)
```

### 3 Beschreibung der grundlegenden Parameter

In diesem Kapitel werden die grundlegenden Parameter für das gesamte Scheduling beschrieben. Die Parameter sind auch in folgendem Bild dargestellt:

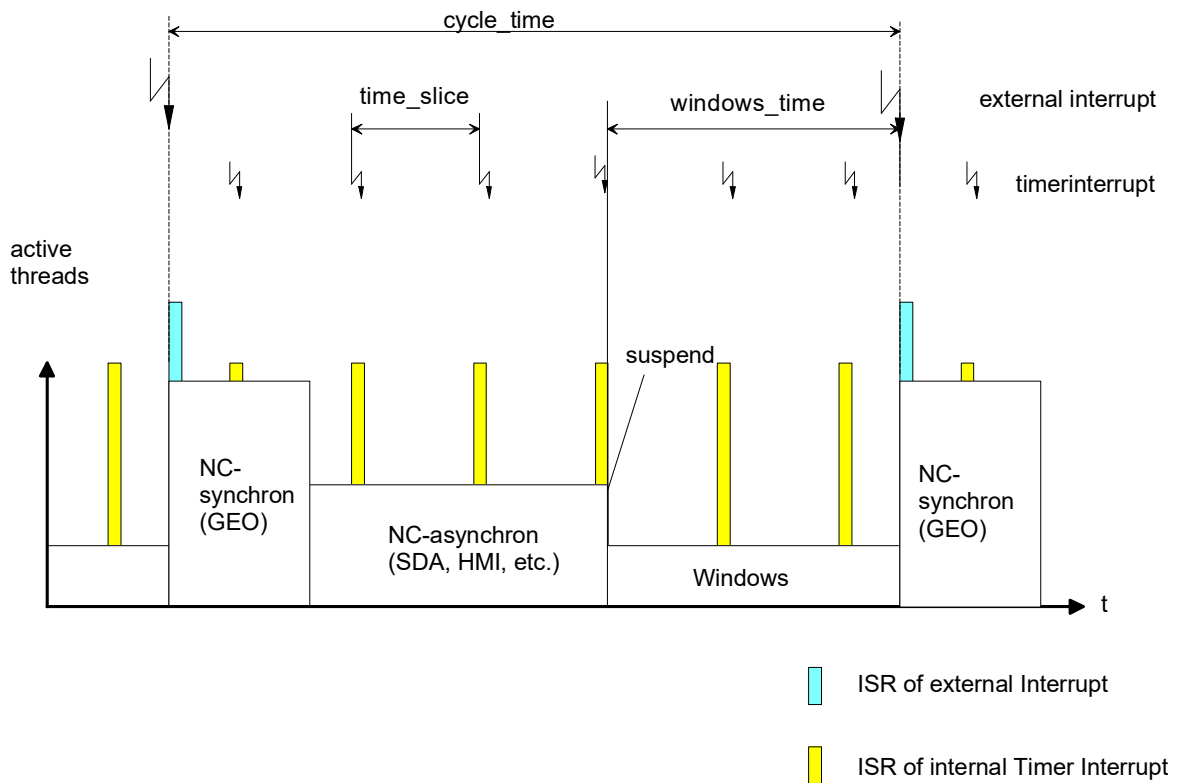


Abb. 2: Grundlegende Parameter des Scheduling

### 3.1 Quelle des Interrupts (P-RTCF-00001)

P-RTCF-00001	Quelle des Interrupts, der den Beginn eines neuen NC-Zyklus auslöst.
Beschreibung	<p>Es wird zwischen externem Interrupt (z.B. SERCOS Hardware) und internem Interrupt (Timer-Interrupt des Echtzeitbetriebssystems) unterschieden. Falls ein externer Interrupt zur Verfügung steht, wird die Ausführung des NC-synchronen Thread direkt an diesen gekoppelt. Der höher auflösende Timer-Interrupt des Betriebssystems wird dann lediglich zur eventuellen Suspendierung des asynchronen Threads benötigt. NC-Zykluszeit und Timer-Interrupt laufen dabei nicht synchron. Das bedeutet auch, dass die Suspendierung des asynchronen Threads um den Betrag von P-RTCF-00003 [▶ 14] (<code>time_slice</code>) variieren kann.</p> <p>Steht kein externer Interrupt zur Verfügung, dann wird nur ein hochauflösender Timer-Interrupt des Betriebssystems zum Starten des NC-synchronen Threads (bei jedem n-ten Timer-Interrupt) herangezogen. NC-Zykluszeit und Timer-Interrupt laufen dann synchron. Bei Verwendung eines externen Betriebssystem-Semaphores wird der Name des Semaphores mit P-RTCF-00005 [▶ 15] (<code>external_object_name</code>) definiert.</p>
Parameter	<code>interrupt_source</code>
Datentyp	UNS16
Datenbereich	1: Hardware ISR 2: Echtzeitbetriebssystem Timer 3: Echtzeitbetriebssystem Semaphore
Dimension	----
Standardwert	0
Anmerkungen	

### 3.2 NC-Zykluszeit (P-RTCF-00002)

P-RTCF-00002	NC-Zykluszeit
Beschreibung	<p>Die NC-Zykluszeit muss stets ein ganzzahliges Vielfaches der Zykluszeit des Timer-Interrupts des Echtzeitbetriebssystems P-RTCF-00003 [▶ 14] (<code>time_slice</code>) sein. Ist dies nicht der Fall, so wird sie auf das nächste größere ganzzahlige Vielfache aufgerundet.</p>
Parameter	<code>cycle_time</code>
Datentyp	UNS32
Datenbereich	$0 < \text{cycle\_time} < \text{MAX}(\text{UNS32})$
Dimension	$\mu\text{s}$
Standardwert	-
Anmerkungen	

### 3.3 Timer-Interrupt (P-RTCF-00003)

<b>P-RTCF-00003</b>	<b>Timer-Interrupt Zeitdistanz des Echtzeitbetriebssystems</b>
Beschreibung	Die Timer-Interrupt Zeitdistanz des Echtzeitbetriebssystems sollte deutlich geringer (z.B. Faktor 3 bis 20) als die NC-Zykluszeit sein. Ein Fehler wird jedoch nur dann erzeugt, wenn das Verhältnis kleiner als 1 ist. Bei den Betriebssystemen VxWorks und RTX kann der interne Timer eingestellt werden.
Parameter	time_slice
Datentyp	UNS32
Datenbereich	$0 < \text{time\_slice} < \text{MAX}(\text{UNS32})$
Dimension	$\mu\text{s}$
Standardwert	-
Anmerkungen	Für RTX gilt: Im Initialisierungsmenü (RTX Properties Register Setting) von RTX muss die HAL-Timer-Period mit dem Wert von <i>time_slice</i> übereinstimmen, sonst wird eine Fehlermeldung ausgegeben.

### 3.4 Garantierte Windows-Rechenzeit (P-RTCF-00004)

P-RTCF-00004	Garantierte Windows-Rechenzeit innerhalb eines NC-Zyklus
Beschreibung	<p>Die Windows-Rechenzeit <code>windows_time</code> muss ein ganzzahliges Vielfaches des <code>time_slice</code> sein. Ist dies nicht der Fall, so wird sie auf das nächste kleinere ganzzahlige Vielfache abgerundet.</p> <p>Ist der Vorgänger-Thread vorzeitig fertig, wird der verbleibende Zeitraum, bis zum Ende der Zykluszeit Windows als Rechenzeit bereitgestellt.</p> <p>Falls die garantierte Rechenzeit von Windows durch den Vorgänger-Thread beschnitten wird, erfolgt dessen Suspendierung durch die Timer ISR.</p> <p>Weiter darf die <code>windows_time</code> nicht zu groß gewählt werden, da sonst die von Windows suspendierbaren Threads (alle Threads außer der GEO-Task ) nicht genügend ( bzw. gar keine ) Rechenzeit erhalten. Wird die <code>windows_time</code> kleiner als P-RTCF-00003 [▶ 14] (<code>time_slice</code>) gewählt, wird eine Fehlermeldung ausgegeben.</p>
Parameter	<code>windows_time</code>
Datentyp	UNS16
Datenbereich	$0 < \text{windows\_time} < \text{MAX}(\text{UNS16})$
Dimension	$\mu\text{s}$
Standardwert	-
Anmerkungen	<p>Bei <code>windows_time = 0</code> entfällt die Suspendierung des Vorgängerprozesses komplett, wodurch Windows keine garantierte Rechenzeit mehr bekommt. Eine solche Parametrierung ist nur sinnvoll, wenn sichergestellt ist, dass regelmäßig alle Threads vor Ablauf der NC-Zykluszeit „fertig“ werden. Das setzt voraus, dass sich alle Threads kooperativ verhalten.</p> <p>Geeignete Einstellung der <code>windows_time</code> in Bezug auf <code>time_slice</code> (P-RTCF-00003 [▶ 14]):  <b><code>windows_time = time_slice = 500\mu\text{s}</code></b></p> <p>Kleinere Zeiträume (z.B. 100 - 200<math>\mu\text{s}</math>) sind ungünstiger, da ein Kontextwechsel zwischen Windows und Echtzeitbetriebssystem relativ zeitintensiv ist.</p> <p>Bei zu klein gewählter <code>windows_time</code> (z.B. 100<math>\mu\text{s}</math>) können Instabilitäten im Echtzeitbetrieb auftreten (Problem von RTX).</p>

### 3.5 Name des externen Betriebssystem-Semaphore (P-RTCF-00005)

P-RTCF-00005	Name des externen Betriebssystem-Semaphore
Beschreibung	Der Parameter legt den Namen des Semaphors für P-RTCF-00001 [▶ 13] (interrupt source) fest.
Parameter	<code>external_object_name</code>
Datentyp	STRING
Datenbereich	Maximale Namenslänge abhängig vom Betriebssystem
Dimension	----
Standardwert	*
Anmerkungen	<p>* Hinweis: Der Standardwert der Variablen ist ein Leerstring.</p> <p>Dieser Parameter wird nur verwendet, wenn <code>interrupt_source = 3</code>.</p>





## 4 Thread-spezifische Parameter (thread[i].\*)

Strukturname	Index
thread[i]	i = 0 ... 6 (Maximale Anzahl Threads 7, applikationsspezifisch)

### 4.1 Name des Threads (P-RTCF-00008)

P-RTCF-00008	Name des Threads im Scheduling
Beschreibung	Der Parameter legt den Namen des Threads im Scheduling fest.
Parameter	thread[i].name
Datentyp	STRING
Datenbereich	Maximal 127 Zeichen
Dimension	----
Standardwert	*
Anmerkungen	* Hinweis: Der Standardwert der Variablen ist ein Leerstring. Der Name ist prinzipiell frei wählbar z.B. GEO, BACKGROUND, MMI_DRIVER, COM

### 4.2 Zykluszeit des Threads (P-RTCF-00009)

P-RTCF-00009	Zykluszeit des Threads
Beschreibung	Mit diesem Parameter wird festgelegt, ob der Thread in jedem oder nur in jedem n-ten Intervall durch den Scheduler aufgerufen wird. Deshalb muss der Wert des Parameters als ganzzahliges Vielfaches der Zykluszeit gewählt werden. Ist das nicht der Fall, wird der Wert auf das nächste ganzzahlige Vielfache der NC-Zykluszeit aufgerundet.
Parameter	thread[i].cycle
Datentyp	UNS32
Datenbereich	n * cycle_time (P-RTCF-00002 [▶ 13])
Dimension	----
Standardwert	0
Anmerkungen	

### 4.3 Aufrufreihenfolge des Threads (P-RTCF-00010)

<b>P-RTCF-00010</b>	<b>Festlegung der Aufrufreihenfolge des Threads innerhalb des Scheduling</b>
Beschreibung	Mit diesem Parameter wird die Aufrufreihenfolge des Threads innerhalb des Scheduling festgelegt.
Parameter	thread[i].priority
Datentyp	SGN32
Datenbereich	Betriebssystemabhängig: Bei RTX: 0 ... 127 (127 = highest) Bei VxWorks: 255 ... 0 (0 = highest)
Dimension	----
Standardwert	0
Anmerkungen	<p>Hier sei nochmals darauf hingewiesen, dass die Threadprioritäten die Reihenfolge der Ausführung der einzelnen Threads festlegen.</p> <p><b>Für RTX gilt:</b> Bei RTX dürfen keine Prioritäten &gt; 124 parametrieren, da in diesem Prioritätsbereich der Scheduler selbst (die ISR-Threads) sowie der Shutdown Handler laufen. Typische Prioritätswahl:</p> <ul style="list-style-type: none"> <li>• GEO 115</li> <li>• BACKGROUND 105</li> <li>• MMI_DRIVER 95</li> <li>• COM 85</li> </ul>

### 4.4 Fehlermeldung bei Überlauf des Threads (P-RTCF-00011)

<b>P-RTCF-00011</b>	<b>Fehlermeldung bei Überlauf des betreffenden Threads</b>
Beschreibung	Parameter legt fest, ob eine Fehlermeldung bei Überlauf des betreffenden Threads ausgegeben wird.
Parameter	thread[i].error_on_overflow
Datentyp	BOOLEAN
Datenbereich	0: Keine Ausgabe einer Fehlermeldung 1: Ausgabe einer Fehlermeldung bei Threadüberlauf
Dimension	----
Standardwert	0
Anmerkungen	

## 4.5 Funktionen in einem Thread (thread[i].function[j].\*)

Jeder Thread kann aus Funktionen bestehen, die mit den hier aufgeführten Variablen parametrisiert werden.

Strukturname	Index
thread[i].function[j]	i = 0 ... 6 (Maximale Anzahl Thread-Funktionen 7, applikationsspezifisch)

### 4.5.1 Name einer Thread-Funktion (P-RTCF-00012)

P-RTCF-00012	Name einer Funktion innerhalb eines Threads
Beschreibung	Der Parameter legt den Namen einer Funktion innerhalb des Threads fest.
Parameter	thread[i].function[j].name
Datentyp	STRING
Datenbereich	Maximal 127 Zeichen Die zu verwendenden Namen sind bereits im Source fest eincompiliert. Folgende Parameterbelegungen sind erlaubt: task_rnd: Rundlauf Thread mit FB SDA task_int: Interrupt Thread mit FB GEO task_com: Kommunikations Thread mit FB COM task_mmi_driver: Systemablaufsteuerung
Dimension	----
Standardwert	*
Anmerkungen	* Hinweis: Der Standardwert der Variablen ist ein Leerstring. Parametrierbeispiel: <i>thread[0].function[0].name task_rnd</i>

### 4.5.2 Anzahl Aufrufe einer Thread-Funktion (P-RTCF-00013)

P-RTCF-00013	Anzahl der Aufrufe für eine bestimmte Funktion innerhalb eines Threads
Beschreibung	Dieser Parameter legt die Anzahl der Aufrufe für eine bestimmte Funktion innerhalb eines Threads fest.
Parameter	thread[i].function[j].calls_per_cycle
Datentyp	UNS32
Datenbereich	0 < P-RTFC-00013 < MAX(UNS32)
Dimension	----
Standardwert	1 *
Anmerkungen	* Allgemein wird <i>calls_per_cycle</i> = 1 gesetzt, bei spezifischen Anwendungen (z.B. im HSC-Bereich) ist eventuell ein mehrfacher Aufruf des Rundlaufteils (BACKGROUND) sinnvoll. Der Parameter muss dann entsprechend den Bedürfnissen angepasst werden. Parametrierbeispiel: <i>thread[0].function[0].calls_per_cycle 1</i>

### 4.5.3 Markierungsbit einer Thread-Funktion (P-RTCF-00014)

<b>P-RTCF-00014</b>	<b>Markierungsbit einer Thread-Funktion zur Diagnose</b>
Beschreibung	Zur Diagnose mit dem Oszilloskop besteht die Möglichkeit, Anfang und Ende einer Thread-Funktion durch Setzen eines Bit auf der Parallelschnittstelle anzuzeigen. Das Zeitverhalten des Scheduling kann somit aufgezeichnet werden.
Parameter	thread[i].function[j].trace_bit
Datentyp	UNS16
Datenbereich	0 ... 7 0: Keine Markierung gesetzt
Dimension	----
Standardwert	0
Anmerkungen	Parametrierbeispiel: <i>thread[0].function[0].trace_bit 1</i>

### 4.5.4 Zuweisung des Threads an einen CPU-Kern (P-RTCF-00015)

<b>P-RTCF-00015</b>	<b>Zuweisung des Threads an einen CPU-Kern</b>
Beschreibung	<p>Mit diesem Parameter kann der Thread an einen bestimmten Kern der CPU zugewiesen werden. Der Thread wird dann ausschließlich auf diesem Kern ausgeführt.</p> <p>Soll der Thread einem Kern zugewiesen werden, der nicht existiert, wird der Fehler ID 1000180 ausgegeben. Die Zuweisung an einen oder mehrere Prozessorkerne wird dann vom Betriebssystem übernommen.</p> <p>Kann die Zuweisung aus einem anderen Grund nicht durchgeführt werden, wird ebenfalls eine Fehlermeldung mit ID 1000181 ausgegeben.</p>
Parameter	thread[i].cpu
Datentyp	UNS16
Datenbereich	1 ... 7
Dimension	----
Standardwert	1
Anmerkungen	Parametrierbeispiel: <i>thread[0].cpu 1</i>

## 4.6 Kontextinformation eines Threads (P-RTCF-00017)

P-RTCF-00017	Kontextinformation eines Threads
Beschreibung	Dieser Parameter ordnet den Thread einem Kontext zu. Um den GEO bzw. SDA Task eines Kanals dem gleichen Kontext zuzuordnen, müssen die einen Kanalparameter P-CHAN-00410 bzw. P-CHAN-00411 entsprechend gesetzt werden.
Parameter	thread[i].context_info
Datentyp	UNS32
Datenbereich	0 <= thread[i].context_info
Dimension	----
Standardwert	0
Anmerkungen	Parametrierbeispiel: <i>thread[0].context_info 1</i>

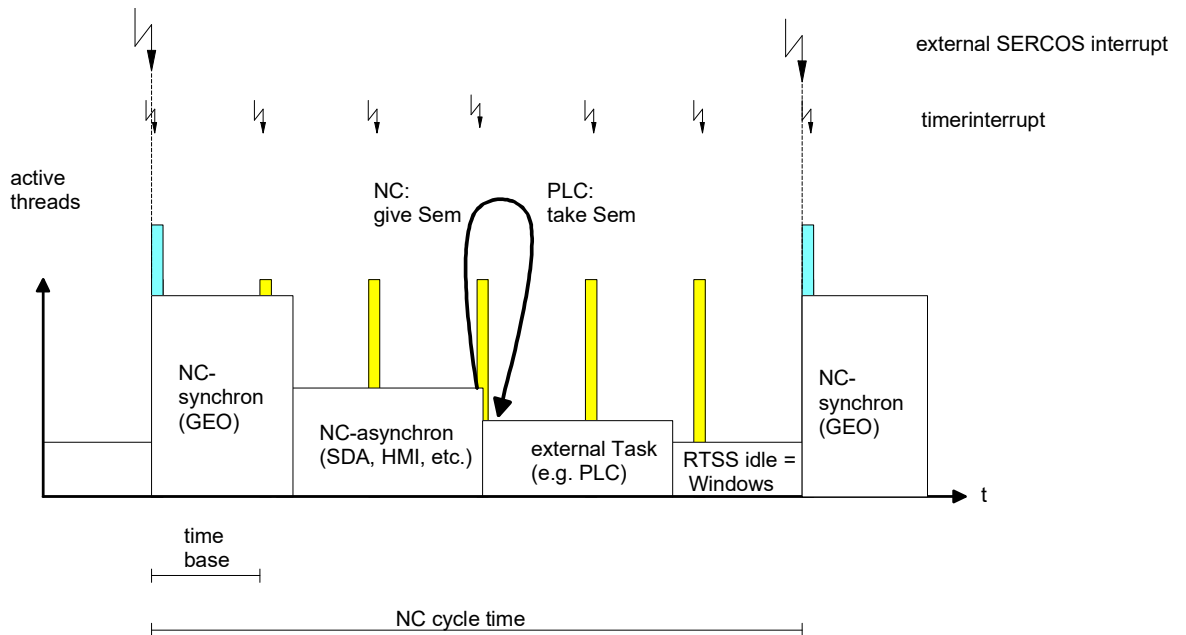
## 4.7 Schedule (P-RTCF-00018)

P-RTCF-00018	Schedule
Beschreibung	Definiert die Reihenfolge, in der Einlesen von Achs-Istwerten. Ausgabe von Achs-Sollwerten und Interpolation abgearbeitet werden, siehe CNC-Scheduling.
Parameter	thread[i].schedule
Datentyp	SGN32
Datenbereich	STANDARD COMPLETE SWITCHED
Dimension	----
Standardwert	STANDARD
Anmerkungen	Parametrierbeispiel: <i>thread[1].schedule STANDARD</i>

## 5 Externe Threads

### 5.1 Allgemeine Beschreibung und Überblick

Zur Synchronisation der NC mit einem oder mehreren externen Thread(s) werden benannte Semaphoren eingerichtet. Pro Zyklus werden diese Semaphoren einmalig gegeben, wodurch die darauf wartenden externen Threads aufgeweckt werden und ihre Arbeit verrichten können. Je nach Priorität der externen Threads reihen sich diese in das Gesamt-Scheduling ein.



**Abb. 3: Synchronisation mit einem externen Thread**

Da Threads nicht prozessübergreifend suspendiert werden können, müssen sich externe Threads kooperativ verhalten. Je nach Priorität des externen Threads geht die Laufzeit des externen Threads zu Lasten des asynchronen NC-Threads oder zu Lasten der garantierten Windows-Zeit.

## 5.2 Spezifische Parameter externer Threads (external\_thread[i].\*)

Strukturname	Index
external_thread[i]	i = 0 ... 3 (Maximale Anzahl externer Threads 4, applikationsspezifisch)

### 5.2.1 Zykluszeit eines externen Threads (P-RTCF-00006)

P-RTCF-00006	Zykluszeit eines externen Threads
Beschreibung	Mit diesem Parameter wird festgelegt, ob der externe Thread in jedem oder nur in jedem n-ten Intervall durch den Scheduler aufgerufen wird. Deshalb muss der Wert des Parameters als ganzzahliges Vielfaches der Zykluszeit gewählt werden. Ist das nicht der Fall, wird der Wert auf das nächste ganzzahlige Vielfache der NC-Zykluszeit aufgerundet.
Parameter	external_thread[i].cycle
Datentyp	UNS32
Datenbereich	n * cycle_time (P-RTCF-00002 [▶ 13])
Dimension	----
Standardwert	0
Anmerkungen	Parametrierbeispiel: <i>external_thread[0].cycle 2000</i>

### 5.2.2 Name der Semaphore eines externen Threads (P-RTCF-00007)

P-RTCF-00007	Name der Semaphore eines externen Threads
Beschreibung	Der Parameter legt den Namen des Semaphors eines externen Threads fest.
Parameter	external_thread[i].semaphore_name
Datentyp	STRING
Datenbereich	Maximal 127 Zeichen
Dimension	----
Standardwert	*
Anmerkungen	* Hinweis: Der Standardwert der Variablen ist ein Leerstring. Parametrierbeispiel: <i>external_thread[0].semaphore_name external_sps</i>

### 5.2.3

### Beispiel einer Rtconf.lis für eine SERCOS Applikation

```
# *****
#
# *****
# RTX [0; 127] 127 highest
# VXWORKS [0; 255] 0 highest
# INTIME [0; 255] 0 highest
#
interrupt_source 2 # 1: internal Timer, 2: IRQ, 3: Semaphore
cycle_time 2000 # Zykluszeit in micro s
time_slice 500
windows_time 500 # 0 turned off
#
thread[0].name GEO
thread[0].cycle 2000
thread[0].priority 115 # 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].cycle 2000
thread[1].priority 105 # 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 2
#
thread[2].name MMI_DRIVER
thread[2].cycle 6000
thread[2].priority 95 # LOWEST
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].cycle 4000
thread[3].priority 90
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
#
external_thread[0].semaphore_name external_sps
external_thread[0].cycle 2000
#
Ende
```



Alternativ kann z.B. die Kommunikation auch zusammen mit dem Rundlaufteil aufgerufen werden. Dann hat die `rtconf.lis` folgendes Aussehen:

```
# *****
#
# *****
# RTX [0; 127] 127 highest
# VXWORKS [0; 255] 0 highest
# INTIME [0; 255] 0 highest
#
interrupt_source 2 # 1 == internal Timer, 2 == external
cycle_time 2000 # Zykluszeit in micro s
time_slice 500
windows_time 500 # 0 turned off
#
thread[0].name GEO
thread[0].cycle 2000
thread[0].priority 115 # 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].cycle 2000
thread[1].priority 105 # 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 2
thread[1].function[1].name task_com
thread[1].function[1].calls_per_cycle 1
thread[1].function[1].trace_bit 4
#
thread[2].name MMI_DRIVER
thread[2].cycle 6000
thread[2].priority 95 # LOWEST
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
# task_com nach thread[1] verlagert:
# thread[3].name COM
# thread[3].cycle 4000
# thread[3].priority 90
# 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
#
external_thread[0].semaphore_name external_sps
external_thread[0].cycle 2000
#
Ende
```

## 6 Anhang

### 6.1 Anregungen, Korrekturen und neueste Dokumentation

Sie finden Fehler, haben Anregungen oder konstruktive Kritik? Gerne können Sie uns unter [documentation@isg-stuttgart.de](mailto:documentation@isg-stuttgart.de) kontaktieren. Die aktuellste Dokumentation finden Sie in unserer Onlinehilfe (DE/EN):



QR-Code Link: <https://www.isg-stuttgart.de/documentation-kernel/>

Der o.g. Link ist eine Weiterleitung zu:

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



#### Hinweis

##### Mögliche Änderung von Favoritenlinks im Browser:

Technische Änderungen der Webseitenstruktur betreffend der Ordnerpfade oder ein Wechsel des HTML-Frameworks und damit der Linkstruktur können nie ausgeschlossen werden.

Wir empfehlen, den o.g. „QR-Code Link“ als primären Favoritenlink zu speichern.

##### PDFs zum 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](mailto:documentation@isg-stuttgart.de)

## Stichwortverzeichnis

### P

---

P-RTCF-00001 .....	13
P-RTCF-00002 .....	13
P-RTCF-00003 .....	14
P-RTCF-00004 .....	15
P-RTCF-00005 .....	15
P-RTCF-00006 .....	23
P-RTCF-00007 .....	23
P-RTCF-00008 .....	17
P-RTCF-00009 .....	17
P-RTCF-00010 .....	18
P-RTCF-00011 .....	18
P-RTCF-00012 .....	19
P-RTCF-00013 .....	19
P-RTCF-00014 .....	20
P-RTCF-00015 .....	20
P-RTCF-00017 .....	21
P-RTCF-00018 .....	21



© Copyright  
ISG Industrielle Steuerungstechnik GmbH  
STEP, Gropiusplatz 10  
D-70563 Stuttgart  
Alle Rechte vorbehalten  
[www.isg-stuttgart.de](http://www.isg-stuttgart.de)  
[support@isg-stuttgart.de](mailto:support@isg-stuttgart.de)

