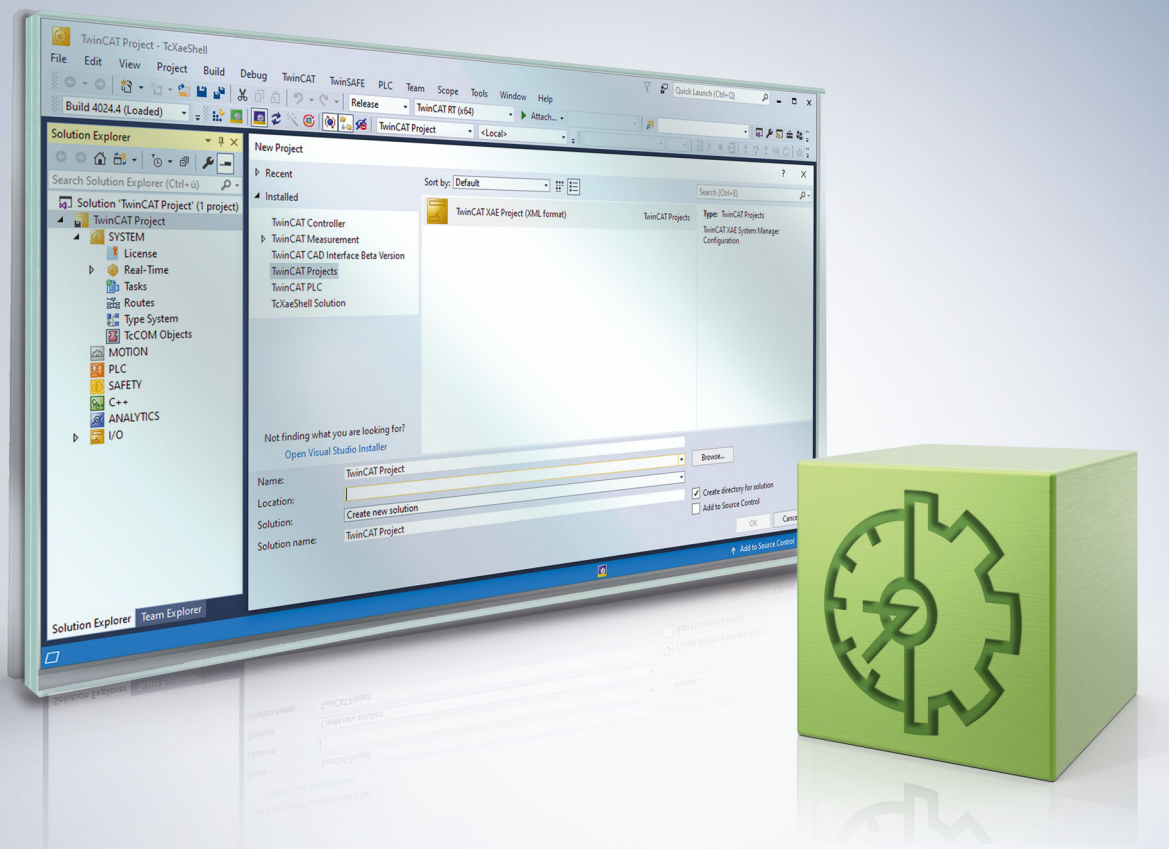
BECKHOFF New Automation Technology

Handbuch | DE

TE1010

TwinCAT 3 | Realtime Monitor



08.05.2023 | Version: 1.3.2

Inhaltsverzeichnis

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.
Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.
Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentliche Dokumentation zu verwenden.

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

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.
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.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.
Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

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

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

⚠ GEFAHR

Akute Verletzungsgefahr!

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

⚠ WARNUNG

Verletzungsgefahr!

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

⚠ VORSICHT

Schädigung von Personen!

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

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

Tipp oder Fingerzeig

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem https://www.beckhoff.de/secguide.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter https://www.beckhoff.de/secinfo.

2 Übersicht

Der TwinCAT 3 Realtime Monitor ermöglicht eine präzise Diagnose und Optimierung des Laufzeitverhaltens von Tasks in der TwinCAT-3-Runtime. Er bietet eine grafische Darstellung der zeitlichen Abarbeitung von Echtzeit-Tasks und deren Module über alle Rechenkerne hinweg. Zudem können durch entsprechende Instrumentalisierung der Steuerungssoftware auch benutzerdefinierte Abarbeitungsprozesse und deren Abhängigkeiten grafisch dargestellt werden.

Der Realtime Monitor macht das Zeitverhalten der Steuerungssoftware auf einem Zielsystem vollständig transparent und ermöglicht eine umfassende zeitliche Analyse. Damit unterstützt er sowohl die Fehlerdiagnose als auch die zeitliche Optimierung der Konfiguration, insbesondere auf Multicore-Systemen.

Installation

Die Installation erfolgt über einen separaten Installer. Die Freischaltung der Lizenz erfolgt wie bei TwinCAT 3 üblich.

Voraussetzungen

Der Realtime Monitor kann nur zur Diagnose von TwinCAT 3.1 Laufzeiten ab der TwinCAT 3.1 Version 4024.0 verwendet werden.

Er eignet sich für Windows 10-basierte Zielsysteme, nicht aber für Zielsysteme auf der Basis von Windows CE.

Lizenzierung

Der TwinCAT 3 Realtime Monitor (TE1010) ist ein Engineering Produkt. Die Lizenzierung erfolgt also ausschließlich auf dem Engineering System.

Für dieses Produkt ist keine 7-Tage-Testlizenz verfügbar.

3 Grundlagen

Das folgende Kapitel beschreibt die Grundlagen, welche vor der Verwendung des TwinCAT 3 Realtime Monitors gelesen werden sollten.

3.1 TwinCAT 3 Echtzeit

Entsprechend der Norm DIN 44300 ist die Echtzeit bzw. vielmehr der Echtzeitbetrieb definiert als: „Echtzeitbetrieb ist ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind, derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind.“.
Mit anderen Worten bedeutet dies, dass die Ausgabewerte eines Anwenderprogramms, berechnet basierend auf dem inneren Zustand und den Eingabewerten, innerhalb einer definierten und garantierten Zeit zur Verfügung stehen. Diese definierte Zeit wird auch Zykluszeit genannt.

Das Anwendungsprogramm selbst kann aus mehreren Programmbausteinen bestehen, die wiederum andere Programme, Funktionsbausteine etc. aufrufen (siehe auch Norm IEC 61131-3). Die Programmbausteine können Echtzeit-Tasks zugeordnet werden, welche diese wiederum mit einer zu definierenden Zykluszeit und einer definierten Priorität aufrufen.

Die TwinCAT 3 Echtzeit ist eine Echtzeiterweiterung, welche in der aktuellen TwinCAT 3.1 Version unter den Microsoft Windows Betriebssystemen ab Windows 7 verwendet werden kann. Um den beschriebenen Anforderungen an eine Steuerung von industriellen Prozessen gerecht zu werden, unterstützt die TwinCAT 3 Echtzeit die folgenden Eigenschaften:

- Echtzeitfähiges Scheduling
- Parallele Abarbeitung von Prozessen
- Direkter Hardwarezugriff

Darüber hinaus gehend bietet die TwinCAT 3 Echtzeit auch Multicore-Support, um den immer weiter steigenden Anforderungen an eine performante und flexible/erweiterbare Steuerungsplattform gerecht zu werden. Die verfügbaren Rechenkerne können dabei entweder exklusiv für TwinCAT genutzt werden oder sie werden mit Windows geteilt. Im Folgenden werden die Kerne daher als "isolated" oder "shared" bezeichnet.

Echtzeitfähiges Scheduling

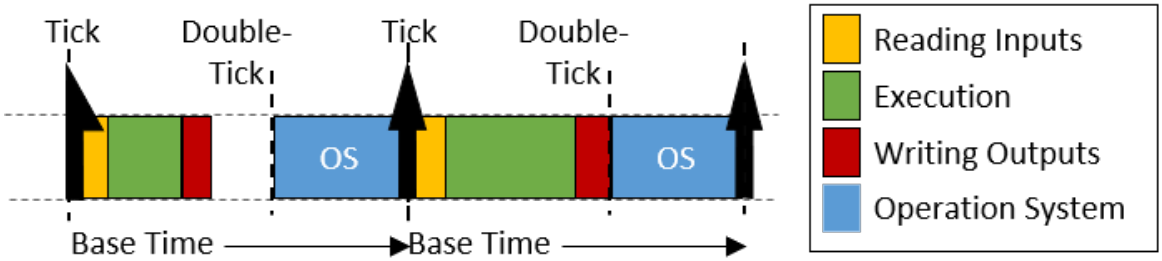
Die TwinCAT 3 Echtzeit arbeitet mit dem Doppeltick-Verfahren. Das bedeutet, dass das Umschalten in den Echtzeitmodus und wieder zurück jeweils von einem Interrupt ausgelöst wird. Der Interrupt beim Umschalten in den Echtzeitmodus startet gleichzeitig auch das Scheduling. Nach einer einstellbaren Zeitdauer, spätestens aber nach 90% der eingestellten Zykluszeit, schaltet TwinCAT auf „shared“-Kernen in den Nicht-Echtzeitmodus zurück, damit das Gastbetriebssystem genügend Rechenzeit erhält, um seinerseits die notwenigen Antwortzeiten für Hardware-Funktionen etc. einzuhalten. Eine Ausnahme bilden hier die isolierten Kerne.

Als Scheduling wird der (System-)Prozess bezeichnet, welcher die Abarbeitungsreihenfolge und den Abarbeitungszeitpunkt der einzelnen Tasks, basierend auf der definierten Zykluszeit und der definierten Priorität bestimmt. Die strenge Einhaltung des Abarbeitungszeitpunkts sorgt dafür, dass die oben beschriebene Einhaltung der Echtzeit gewährleistet wird.

Angestoßen durch einen synchronen Basis-Tick auf allen Echtzeitkernen, wird in der TwinCAT 3 Echtzeit das Scheduling für jeden Echtzeitkern unabhängig berechnet. Damit ist garantiert, dass Echtzeit-Tasks, welche auf verschiedenen Kernen laufen, sich nicht beeinflussen. Sofern dies nicht durch die Verwendung von Verriegelungen explizit im Anwenderprogramm programmiert wurde.

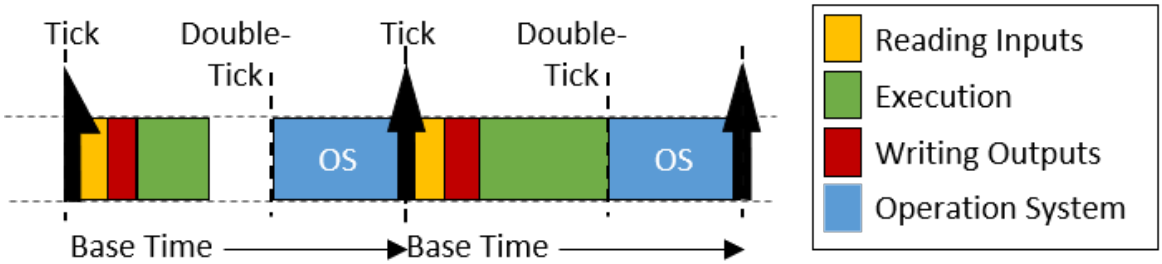
Ein Scheduling, bei dem die Priorität eines Tasks anhand seiner Zykluszeit abgeleitet wird, bezeichnet man auch als Ratenmonotones Scheduling. Die TwinCAT 3 Echtzeit aktiviert automatisch die Option „Automatic Priority Management“. Da dies nicht für jeden Anwendungsfall die beste Lösung ist, können Sie die Prioritäten manuell anpassen.

Beispielhafte Darstellung des Aufrufs einer SPS-Task



In der Abbildung dargestellt sehen Sie den Aufruf einer SPS-Task. Nachdem der Echtzeit-Tick erfolgt ist, wird vom Scheduler die SPS-Task aufgerufen. Dieser stellt der SPS-Anwendung die aktuellen Eingangswerte zur Verfügung (Input-Update), danach erfolgt die Abarbeitung des Anwendungsprogramms (Cycle-Update) und abschließend das Schreiben der Ergebnisse auf die Ausgänge (Output-Update). Ist dieses beendet, erfolgt das Umschalten in den Nicht-Echtzeit-Mode (Doppeltick). Wie in der Abbildung zu sehen ist, kann die Ausführungsdauer des Anwenderprogramms variieren, je nachdem welcher Code basierend auf dem inneren Zustand des Programms durchlaufen wird. Somit variiert auch der Zeitpunkt, wann die Ausgänge geschrieben werden. Je nachdem welche Task unter Umständen ein Bussystem treibt, kann dies dazu führen, dass das Absenden der Bustelegramme in gleichem Maße variiert.

Beispielhafter Aufruf einer Task mit „IO am Task-Beginn“



Durch die Verwendung der Option „IO am Task-Beginn“ kann die Abarbeitungsreihenfolge innerhalb einer Task dahingehend geändert werden, dass nach dem Lesen der Eingänge direkt die Ausgänge (des vorhergehenden Zyklus) geschrieben werden, bevor die Abarbeitung des Anwendungsprogramms erfolgt. Obwohl das Schreiben der Ausgänge erst einen Zyklus später erfolgt, hat diese Einstellung den Vorteil, dass der Zeitpunkt, wann die Ausgänge auf den Prozess / den Bus geschrieben werden, in jedem Zyklus exakt derselbe ist.

Präemptives Multitasking

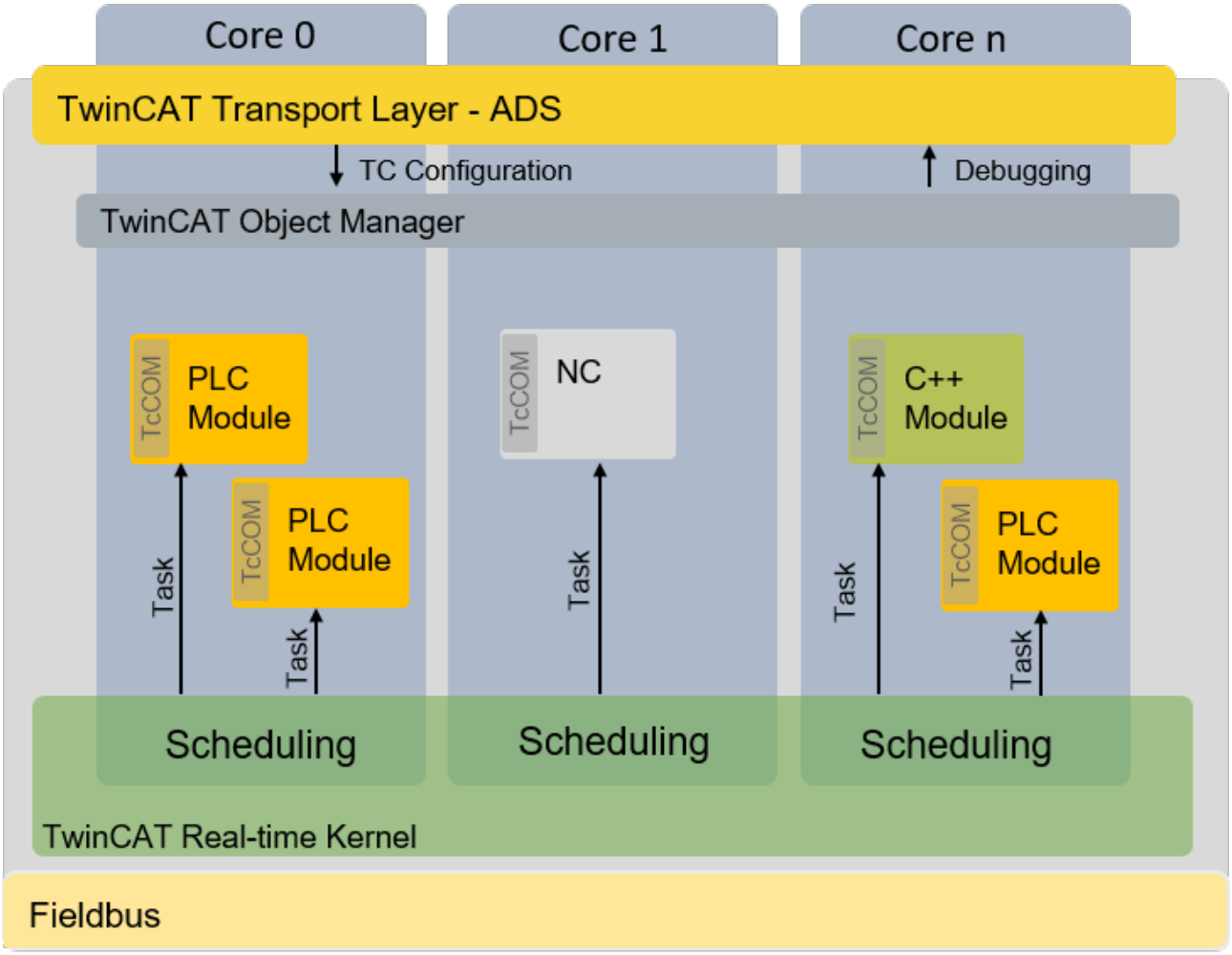
Präemptives Multitasking bedeutet, dass der aktuelle Zustand eines Prozesses (die CPU- und Floatingpoint-Register), bei einer Unterbrechung durch einen Interrupt (z. B. durch höher priore Prozesse), gesichert und der aktuelle Prozess „schlafen gelegt“ wird. Ist dies passiert, bestimmt der Scheduler, anhand der Prioritäten der Tasks, den (neuen) abzuarbeitenden Prozess. Nachdem der zu unterbrechende Prozess beendet wurde, wird der Prozesskontext wiederhergestellt und der „alte“ Prozess fortgesetzt.

Direkter Hardwarezugriff

Um ein deterministisches (reproduzierbares) Echtzeit-Verhalten zu erreichen, benötigt die TwinCAT 3 Echtzeit einen direkten Hardwarezugriff. Damit dies möglich ist, muss die TwinCAT 3 Echtzeit im sogenannten Kernel-Mode von Windows ausgeführt werden. Dadurch ist es u.a. möglich, dass die TwinCAT-Echtzeit direkt auf die Netzwerk-Ports zugreift und Echtzeit-Ethernet-Telegramme (z. B. EtherCAT) versenden und empfangen kann.

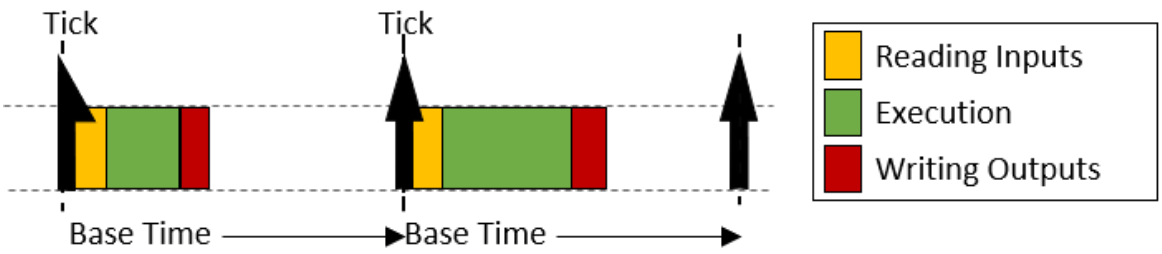
Schematische Darstellung der TwinCAT 3-Laufzeitumgebung

Das folgende Bild stellt den Aufbau der TwinCAT 3.1 Laufzeitumgebung (Runtime), bezogen auf das Scheduling, schematisch dar. Die TwinCAT 3 Laufzeitumgebung ermöglicht das Ausführen von Anwendermodulen in Echtzeit. Ein wesentlicher Teil der TwinCAT 3 Laufzeitumgebung ist somit der Echtzeit-Treiber, welcher auf den für TwinCAT aktivierten Kernen ausgeführt wird und dort das Scheduling übernimmt. Letzteres erfolgt auf den einzelnen Kernen unabhängig voneinander.



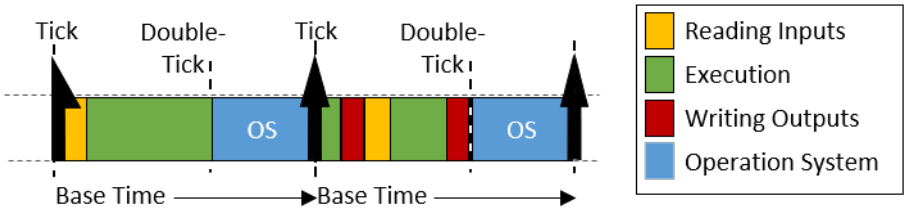
Isolierte Kerne

Wie unter TwinCAT 3 Echtzeit [▶ 9] beschrieben, verwendet TwinCAT ein Doppeltick-Verfahren, damit zu einem festgelegten Zeitpunkt in den Nicht-Echtzeitmodus zurückgeschaltet wird. Beim Umschalten zwischen Echtzeit-Modus und Nicht-Echtzeit-Modus erfolgt, wie unter TwinCAT 3 Echtzeit [▶ 10] beschrieben, ein Restaurieren des vorgehenden Prozesszustands. Je nachdem wie intensiv die Echtzeit- und Nicht-Echtzeit-Programme den Speicher und insbesondere den Cache auslasten, braucht das Wiederherstellen Zeit. Um diese zeitlichen Effekte zu beseitigen, erlaubt es die TwinCAT 3.1 Echtzeit, Kerne vom Gastbetriebssystem zu isolieren. Dadurch ist ein Zurückschalten nicht mehr erforderlich, was sowohl in mehr Rechenzeit für das Echtzeit-Anwenderprogramm resultiert als auch in einer besseren Echtzeit-Güte (geringerer Jitter) durch die Vermeidung von zeitlichen Effekten beim Wiederherstellen das „alten“ Prozesszustands.

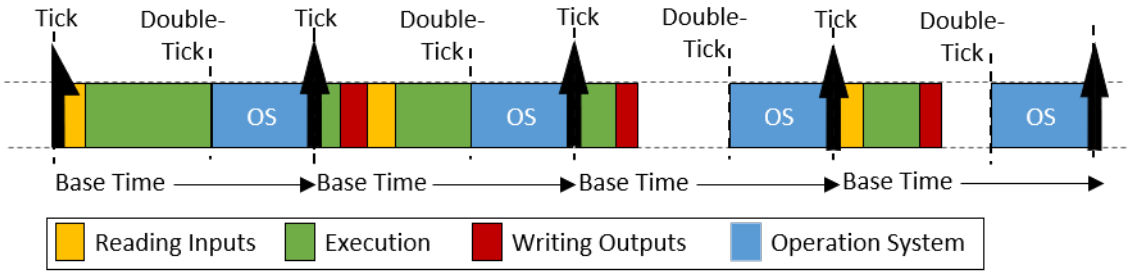


Verhalten bei Zykluszeitüberschreitung

Wird die definierte Zykluszeit eines Tasks überschritten, wird im nächsten Zyklus die Abarbeitung des „alten“ Zyklus fortgesetzt. Zudem wird der Überschreitungszähler der Task nach oben gesetzt. Nach der fertigen Abarbeitung des alten / vorangegangenen Zyklus, wird sofort versucht die Taskabarbeitung des aktuellen Zyklus zu starten. Wird diese innerhalb dieses Zyklus fertig gestellt, erfolgt die weitere Abarbeitung wie oben gezeigt.



Wird auch der zweite direkt darauffolgende Zyklus überschritten (wobei es hierbei unerheblich ist, ob es sich noch um die Abarbeitung des 1. Zyklus oder bereits die Abarbeitung des 2. Zyklus handelt), wird die aktuelle Bearbeitung fertig ausgeführt und das nächste Starten der Abarbeitung der Task startet erst zum nächstmöglichen geplanten Zyklusstart. Es gehen hierbei also unter Umständen mehrere Zyklen verloren. Der Überschreitungszähler wird auch hierbei entsprechend hochgezählt.

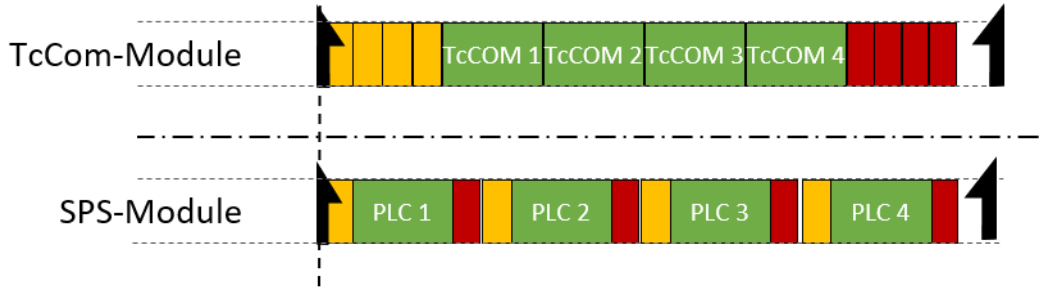


Unterschiede in der Abarbeitung zwischen SPS- zu „TcCom“-Laufzeitmodulen

Die Abarbeitungsreihenfolge einer TwinCAT-Task, bezogen auf die Ausführung von Laufzeitmodulen, besteht aus der folgenden Sequenz:

1. Umkopieren der Eingänge auf die Prozessabbilder der von ihr aufgerufenen Laufzeitmodule.
2. Ausführen der Module entsprechend der Sort-Order (in aufsteigender Reihenfolge).
3. Ausgangs-Update, welches die Ausgänge entsprechend bereitstellt. Treibt diese Task einen EtherCAT-Feldbus, wird der Frame während des Ausgangsprozessabbildes bereitgestellt und versendet.
4. Post-Zyklusupdate: Wird u. a. für das Anstoßen des Zyklusupdates verwendet, wenn die Option „IO am Taskbeginn“ aktiv ist.

Werden Laufzeitmodule einer Task hinzugefügt, „melden“ diese sich an den jeweiligen Aufrufen der Task an. Die einzige Ausnahme sind SPS-Laufzeitmodule. Aus Kompatibilitätsgründen zu TwinCAT 2 erfolgt durch die SPS-Laufzeitmodule direkt das Updaten der Ein- und Ausgänge. Der Unterschied zwischen den beiden Verhaltensweisen wird in der folgenden Abbildung dargestellt:



Zu sehen sind jeweils 4 Laufzeitmodule. Standard-TwinCAT 3 –Laufzeitmodule melden sich bei den entsprechenden Methoden-Aufrufen der Task an. Das bedeutet, alle Ein- (gelb) und Ausgangsupdates (rot) werden von der Task angestoßen und erfolgen direkt nacheinander zu Beginn bzw. am Ende der Taskabarbeitung. Kommunizieren zwei dieser Module über ein Mapping miteinander, so erhalten diese die jeweils aktuellen Werte erst im nächsten Zyklus.

Die SPS-Laufzeitmodule führen eigenständig ein Ein- und Ausgangsupdate durch. Kommunizieren zwei SPS-Laufzeiten miteinander, so bekommt das Laufzeitmodul, welches als zweites ausgeführt wird, direkt die aktuellen Werte vom ersten Laufzeitmodul. Somit ist in der Kommunikationsrichtung 1. Laufzeitmodul -> 2. Laufzeitmodul kein Zyklusversatz, in die andere Richtung aber schon.

3.2 Darstellung im Realtime Monitor

Vereinfacht erklärt, ermöglicht der TwinCAT 3 Realtime Monitor die Darstellung von gruppierten Events. Zur Vermeidung von Bedeutungsüberschneidungen mit dem TwinCAT Eventlogger und den darin geloggten Nachrichten bzw. Alarmen, wird im Kontext des TwinCAT 3 Realtime Monitors von (Zeit-) Marken gesprochen.

Diese Marken können verwendet werden, um das zeitliche Verhalten von Tasks oder Nutzer-Prozessen / Abläufen darzustellen. Dazu werden den Marken eine ID, ein Markentyp, ein Kontext und ein Zeitstempel mitgegeben. Zusätzlich kann bei Bedarf noch ein als UINT formatiertes Nutzer-definiertes Datum mitgegeben werden, um ggf. zusätzliche Informationen mit in die Darstellung im Realtime Monitor einzubringen (z. B. Fehlernummer, Zustand einer State Machine etc.).

Marken-ID:

Die Marken-ID dient zur Identifizierung der dargestellten Task / des dargestellten Prozesses. Mit anderen Worten sollten alle Marken, welche dieselbe Task / denselben Prozess betreffen, dieselbe Marken-ID verwenden.

Markentyp:

Der TwinCAT 3 Realtime Monitor ermöglicht die Darstellung von Ereignissen oder Prozessen / Abläufen aufgetragen über die Zeit. Für die Darstellung von Prozessen / Abläufen werden diese als Sequenz markiert. Eine Sequenz wiederum kann in ein oder mehrere Intervalle unterteilt werden. Marken können entsprechend typisiert werden, um den Start bzw. das Ende von Sequenzen oder Intervallen zu definieren. Zusätzlich können diese auch Ereignisse innerhalb einer Anwendung über die Zeit mit darstellen. Es wird daher zwischen den folgenden Markentypen unterschieden:

1. Marke:
   Die Marke kann verwendet werden, um ein Ereignis festzuhalten, z. B. den Zeitpunkt eines Alarms, den Wechsel eines Zustandes etc.
2. Sequenz-Start:
   Ein Sequenz-Start gibt den Zeitpunkt an, wenn einer Task / einem Prozess erlaubt ist loszulaufen (durch höher priore Tasks / Prozesse geschieht dies u. U. erst verzögert).
3. Intervall-Start:
   Ein Intervall-Start gibt den Zeitpunkt an, wenn eine Task / ein Prozess tatsächlich startet. Aufgrund von Unterbrechungen etc. kann es innerhalb einer Sequenz mehrere Intervall-Starts geben.
4. Intervall-Stopp:
   Ein Intervall-Stopp gibt den Zeitpunkt an, wenn eine Task / ein Prozess nicht mehr ausgeführt wird. Dies kann z. B. aufgrund von Unterbrechungen durch höher priore Tasks oder durch noch nicht erfüllte Abhängigkeiten geschehen.
5. Sequenz-Stopp:
   Ein Sequenz-Stopp gibt den Zeitpunkt an, ab welchem eine Task nicht mehr laufen darf, bzw. ein Prozess beendet ist.

Kontext:

Ein Kontext beschreibt eine Zusammenfassung von Marken bzw. Markengruppen.

Für die System-Tasks, werden alle Tasks, die auf einem Kern abgearbeitet werden, zu einem Kontext zusammengefasst (z. B. Kern 0). Ein solcher (Echtzeit)-Kontext bildet somit das Sheduling innerhalb eines Echtzeitkerns ab. Für diese Echtzeit-Kontexte gilt, dass zu jeder Zeit immer nur genau eine der einem Kontext zugeordneten Tasks aktiv ist. Für Nutzerspezifische Marken-Gruppen gilt diese Einschränkung nicht.

Bei der Verwendung der einfachen Marken (durch Verwendung des FB_Mark), werden die nutzerspezifischen Markengruppen automatisch anhand ihrer Anwendungs-Ports gruppiert. So werden z. B. alle Marken, die aus einem SPS-Projekt mit dem Port 851 heraus abgelegt werden, einem Kontext mit der Context-Id 851 (hexadezial 0x353) zugeordnet.

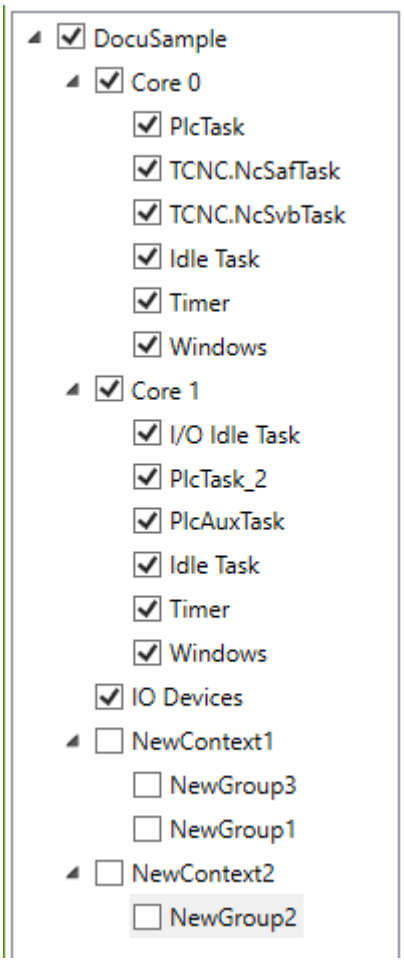
Bei der Verwendung der komplexeren Marken (durch Verwendung des Bausteins FB_RTMon_LogMarkBase [▶ 40]), können selber Kontexte (also Zusammenhänge) definiert werden. Dies könnte beispielhaft eine Gruppierung nach Prozessart oder nach Maschinen-Modulen (Funktionseinheiten) sein.

Darstellung in der Baumansicht:

Wie bereits beschrieben, verwenden alle Marken, die dieselbe Task / denselben Prozess beschreiben, dieselbe Marken-ID. Diese Marken werden zu einer Markengruppe zusammengefasst und erhalten einen Eintrag in der Baumansicht des TwinCAT 3 Realtime Monitors.

Für die System-Tasks wird automatisch ein Eintrag mit dem entsprechenden Namen der Task im Baum angelegt.

Für Nutzer-bezogene Markengruppen, welche z. B. Prozesse beschreiben, muss dies manuell erfolgen. Im Baum erscheint automatisch für jede erkannte nutzerspezifische Markengruppe ein Eintrag **NewGroup**, der anhand der Marken-ID (entspricht der Group-ID im Eigenschaftenfenster der Gruppe) identifiziert werden kann. Diese Gruppe kann entsprechend umbenannt werden (siehe Kontextknoten [▶ 33]).



Wie unter Darstellung im Realtime Monitor [▶ 13] beschrieben, werden die einzelnen Markengruppen zu Kontexten zusammengefasst. Dies geschieht für die System-Tasks, sowie bei der Verwendung der einfachen Marken automatisch. Bei der Verwendung der erweiterten Marken geschieht dies anhand der im Anwendungs-Code übergebenen Context-ID.

Die Benennung der Marken-IDs (Gruppen-IDs) sowie der Kontexte kann für die spätere Wiederverwendung exportiert, bzw. importiert werden.

In der Chart-Darstellung, wird eine Marken-Gruppe (also alle Marken einer Task / eines Prozesses) innerhalb einer Zeile dargestellt. Näheres hierzu unter Darstellung im Realtime Monitor [▶ 15].

Darstellung im Chart:

Symbole in der chart-Darstellung:

	Sequenz-Start
	Sequenz Stopp
	Zeigt ein Intervall-Start oder ein Intervall-Stopp an.
	Marke

Beispielhafte Darstellung:

Die folgende Darstellung zeigt bespielhaft ein mögliches zeitliches Verhalten einer Task. Diese erhält zu einem Zeitpunkt (1) anhand der eingestellten Zykluszeit die „Erlaubnis“ zu laufen. Aufgrund von fehlenden Abhängigkeiten oder aufgrund von noch aktiven höher prioren Tasks läuft diese tatsächlich aber erst zum Zeitpunkt (2) los. Zum Zeitpunkt (3) wurde eine Marke übermittelt. Dies kann sowohl ein „System-Event“ als auch eine nutzerdefinierte Marke sein. Genauer Informationen erfährt man per Tooltip auf der Marke. Die Marke selbst hat keinen Einfluss auf das zeitliche Verhalten der Task / des Prozesses. Zum Zeitpunkt (4) wird die Task unterbrochen (wieder zum Beispiel durch eine Verriegelung oder eine höher priore Task). Zum Zeitpunkt (5) läuft die Task weiter. Zum Zeitpunkt (6) ist die Task beendet.

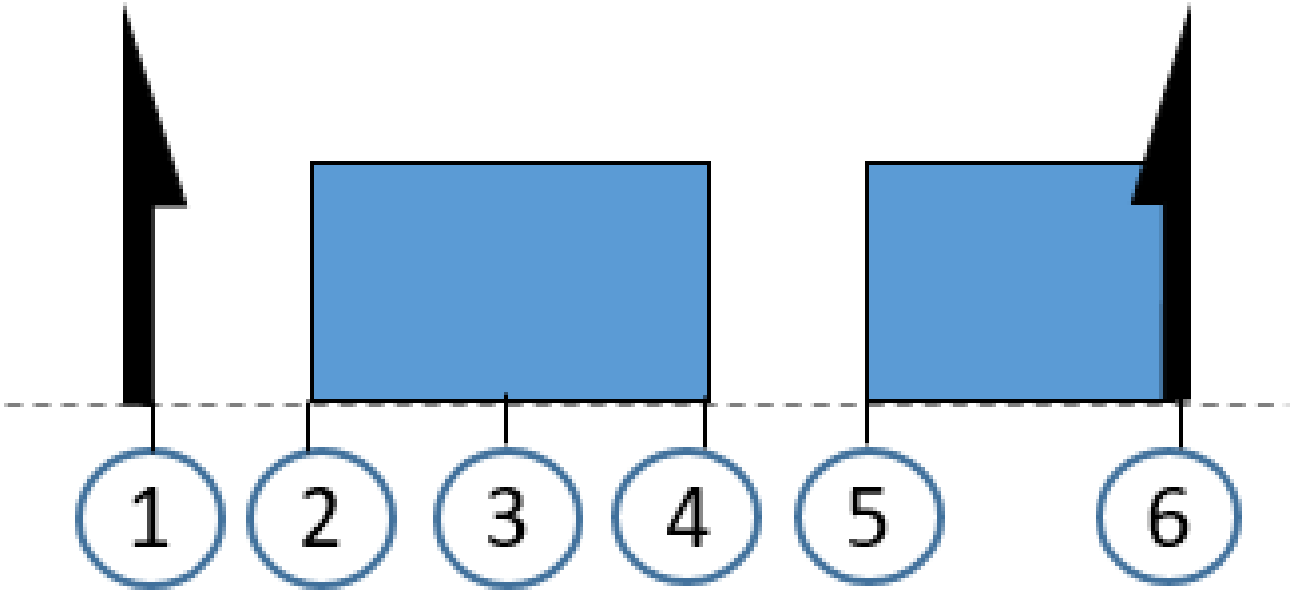
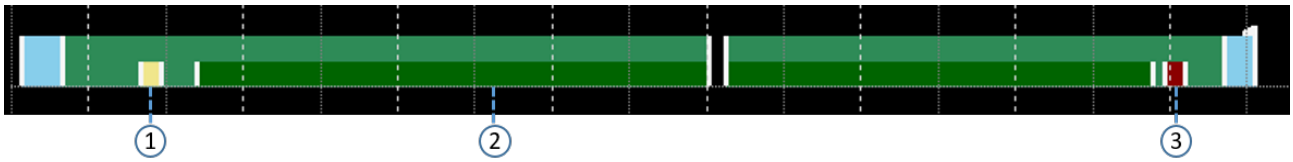


Abbildung der Abarbeitung eines SPS-Laufzeitmoduls:

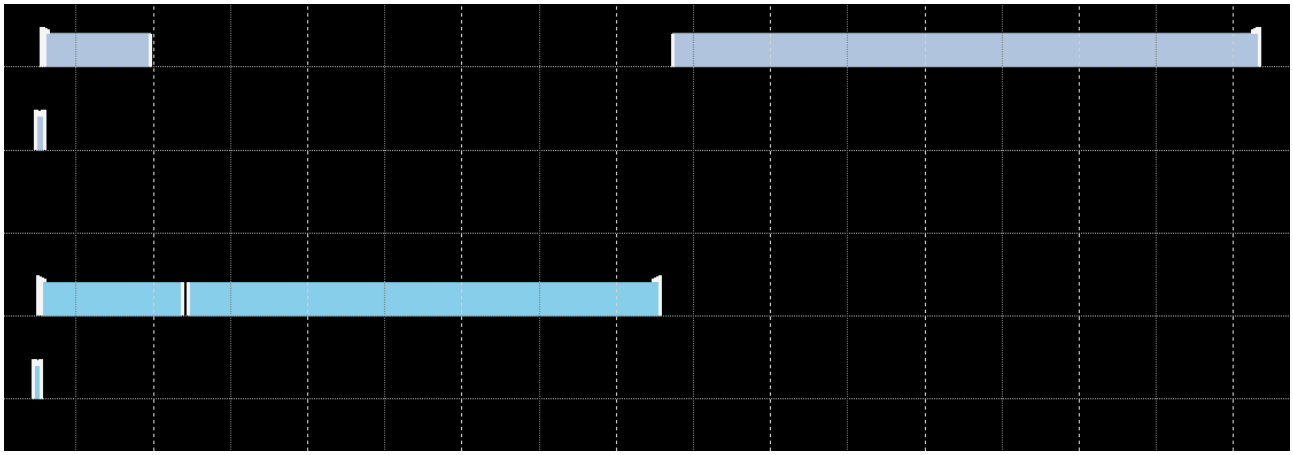
Wie im Absatz TwinCAT 3 Echtzeit [▶ 12] beschrieben, ruft jedes SPS-Laufzeitmodul das Update der Ein- und Ausgänge selbst auf. Die komplette Abarbeitung der SPS findet im Cyclic-Update der sie aufrufenden Task statt. Aus diesem Grund wird die Abarbeitung der SPS, sofern das detaillierte Logging aktiviert ist, überlagert im Cyclic-Update einer Task abgebildet. In der folgenden Abbildung wird dies beispielhaft gezeigt. Der Zeitpunkt (1) zeigt die Ausführung des Eingangsupdates des SPS-Laufzeitmodules. Im Bereich (2) findet die zyklische Abarbeitung des SPS-Codes statt, welche in dem hier gezeigten Beispiel von einer anderen Task unterbrochen wird. Nach der fertigen Abarbeitung findet zum Zeitpunkt (3) das Ausgangsupdate des SPS-Laufzeitmoduls statt. Die Task selbst, führt im hier gezeigten Beispiel kein Ein- oder Ausgangsupdate durch.



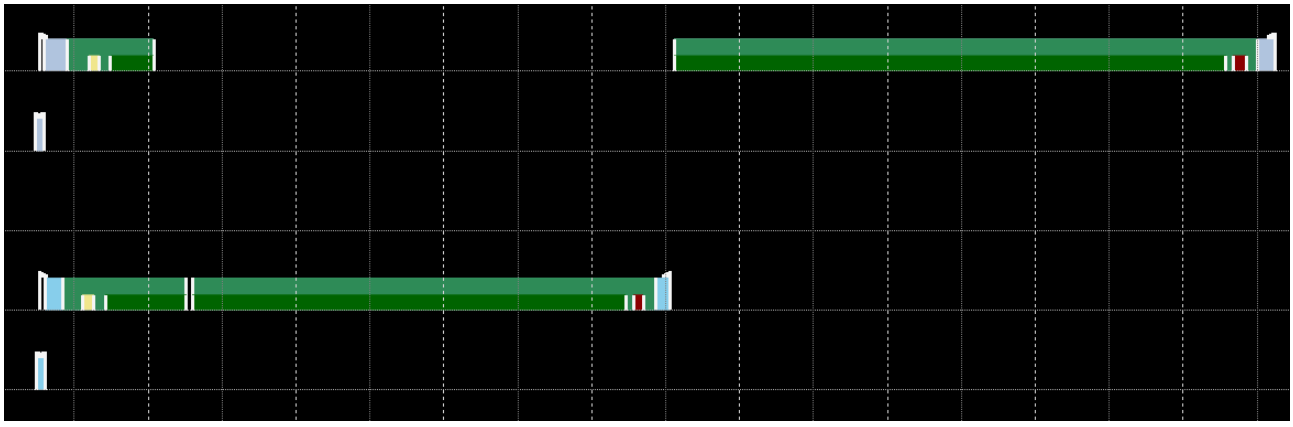
detailliertes Logging:

Die Option „detailedLogging“ (siehe Projektknoten [▶ 32] bzw. Kontextknoten [▶ 33]) erlaubt es, für Echtzeittasks eine detaillierte Darstellung der Ausführung auch innerhalb einer Task zu erhalten. Die folgenden beiden Abbildungen machen den Unterschied sichtbar.

Standard Logging aktiviert:



DetailedLogging aktivert:



Darstellung der Task-Referenzen:

Die Option **Show Task Reference** (siehe Markengruppen-Element [▶ 34]) ermöglicht es, die Zuordnung von Anwenderprozessen zu den Tasks auf denen sie ausgeführt werden, im TwinCAT 3 Realtime Monitor sichtbar zu machen. Dargestellt wird dies durch gestrichelte Linien. In der folgenden Abbildung sieht man die Zuordnung des orange dargestellten Anwenderprozesses zur einer SPS-Task.

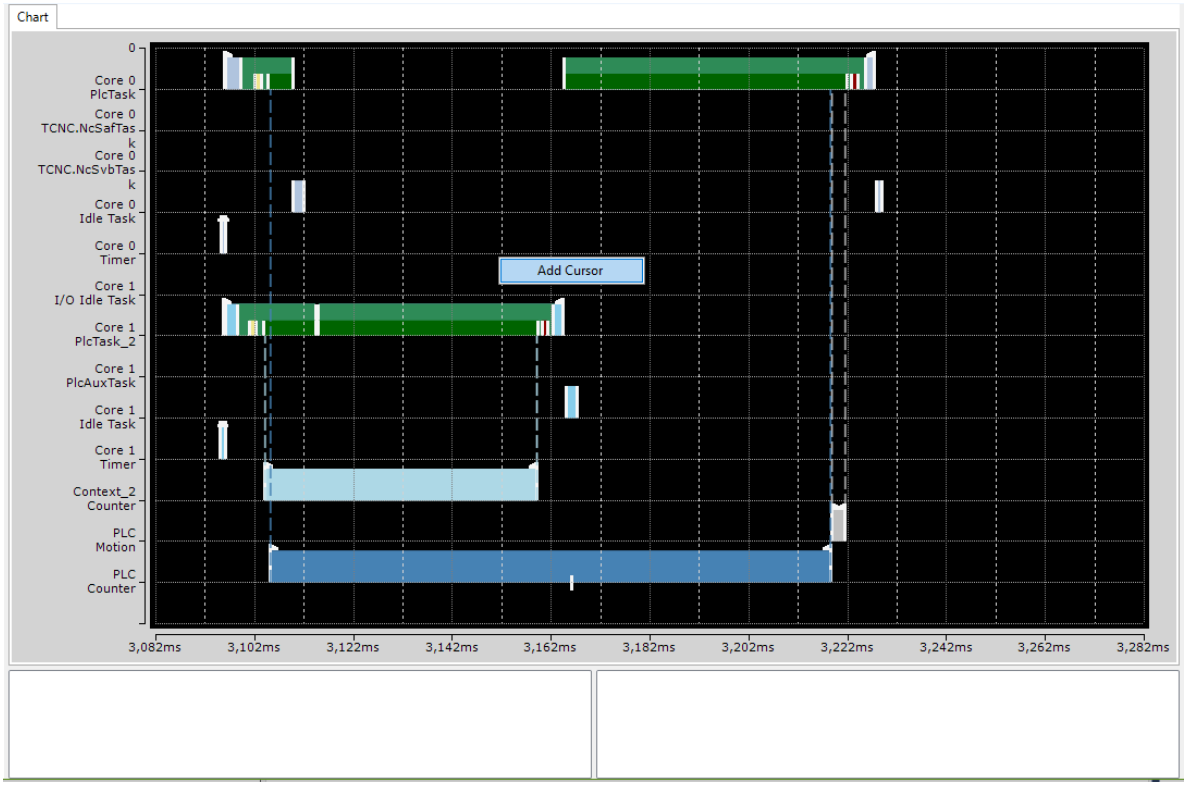
3.3 Verwendung von Cursors

Um Zeiten zu messen bzw. um alle (System-)Events die zu einem Zeitpunkt auftreten darstellbar zu machen, können auch im TwinCAT 3 Realtime Monitor Cursors verwendet werden.

Hinzufügen von Cursors

Zum Hinzufügen eines Cursors gehen Sie wie folgt vor:

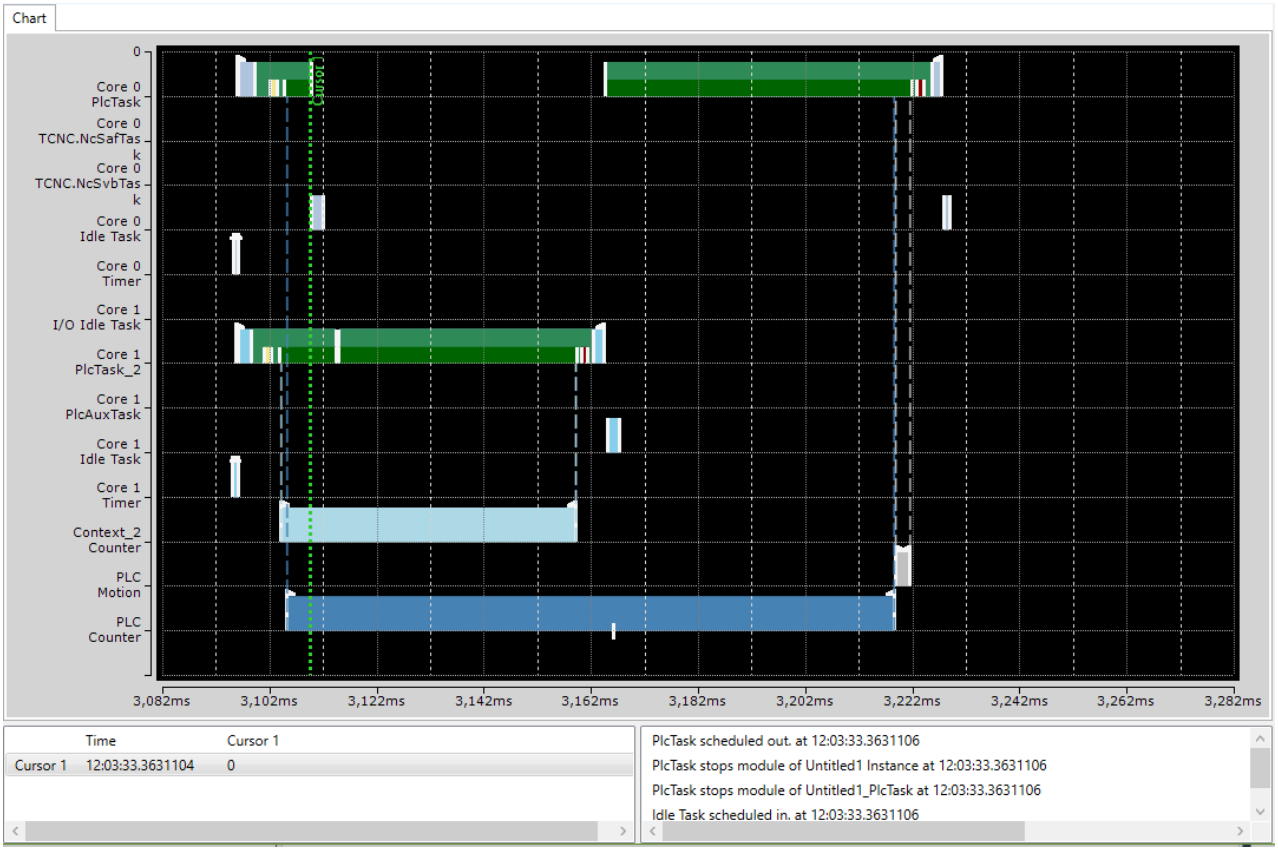
1. Klicken Sie per Rechtsklick innerhalb des Darstellungsbereichs des Charts.
   ⇨ Es öffnet sich ein Kontext-Menü, welches den Befehl **Add Cursor** sowie für alle bereits existierenden Cursors einen Befehl enthält, um diese zu löschen.



⇨

2. Klicken Sie auf **Add Cursor.**

⇨ Ein neuer Cursor wird in der Mitte des Charts angelegt.

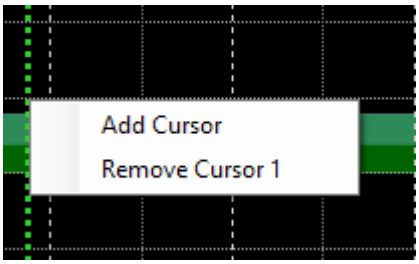


Löschen von Cursors

Zum Löschen eines Cursors gibt es die folgenden beiden Möglichkeiten:

Innerhalb des Charts

1. Klicken Sie per Rechtsklick innerhalb des Darstellungsbereichs des Charts
   ⇨ Es öffnet sich ein Kontext-Menü, welches für alle bereits existierenden Cursors einen Befehl enthält, um diese zu löschen.



2. Verwenden Sie den Befehl **Remove Cursor** des Cursors, den sie löschen wollen.

⇨ Der Cursor ist gelöscht.

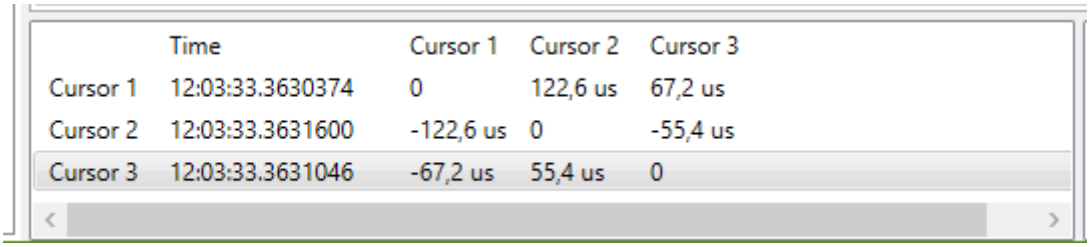
Innerhalb des Cursor-Fensters

1. Klicken Sie per Rechtsklick auf den Cursor, der gelöscht werden soll.
   ⇨ Es öffnet sich ein Kontext-Menü mit dem Befehl den Cursor zu entfernen.
2. Verwenden Sie den Befehl **Remove Cursor** um diesen Cursor zu löschen.

⇨ Der Cursor ist gelöscht.

Navigieren mithilfe der Cursors

Alle erstellten Cursors werden im Cursor-Fenster angezeigt.

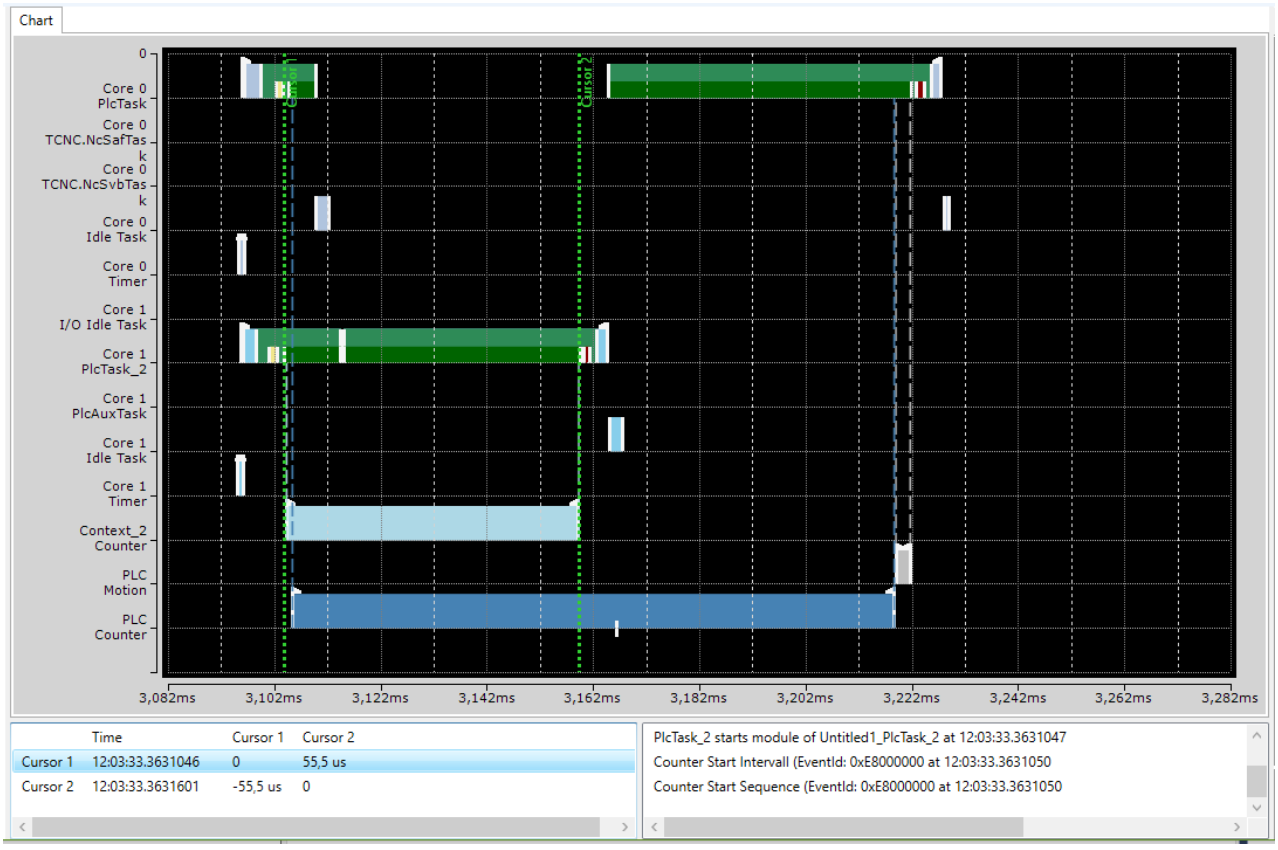


	Time	Cursor 1	Cursor 2	Cursor 3
Cursor 1	12:03:33.3630374	0	122,6 us	67,2 us
Cursor 2	12:03:33.3631600	-122,6 us	0	-55,4 us
Cursor 3	12:03:33.3631046	-67,2 us	55,4 us	0

Ein Doppel-Klick auf einen Cursor sorgt dafür, dass die Darstellung innerhalb des Charts genau an die Stelle springt, an welcher der Cursor steht. Der Cursor wird mittig im Darstellungsbereich angezeigt.

Messen von Zeiten

Die Cursors können verwendet werden, um Ausführungszeiten von Prozessen oder den Zeitpunkt des Auftretens eines Anwenderereignisses exakt zu bestimmen. Hierzu bewegen Sie je einen Cursor an einen für Sie relevanten Zeitpunkt innerhalb der Darstellung. Der Cursor „Rastet“ automatisch bei Events ein und stellt die zu dem Zeitpunkt auftretenden Events für den aktiven Cursor im Cursor-Fenster dar. In der folgenden Abbildung ist dies für den Cursor 1 das Anwendungsereignis **Counter Start Intervall**.



Möchten Sie die Dauer des Prozesses „Context_2_Counter“ messen, gehen Sie wie folgt vor:

1. Erstellen Sie einen Cursor für den Prozess-Start oder verwenden Sie einen bereits existierenden.
2. Bewegen Sie den Cursor auf die Sequenz- /-Intervall-Start-Markierung des Prozesses „Context_2_Counter“ so dass im Event-Fenster dieses Ereignis angezeigt wird (siehe Abbildung oben).
3. Gehen Sie mit einem weiteren Cursor analog vor und bewegen Sie diesen auf die Sequenz- / Intervall-Stopp-Markierung des Prozesses „Context_2_Counter“.
   ⇨ Im Cursor-Fenster werden nun in einer tabellarischen Darstellung die Differenzen zwischen allen existierenden Cursors angezeigt.
4. Lesen Sie den Wert für den von Ihnen verwendeten Cursor direkt aus der tabellarischen Darstellung ab. Für das hier aufgezeigte Beispiel wäre das eine Ausführungsdauer von 55,5µs.

Eigenschaften von Cursors

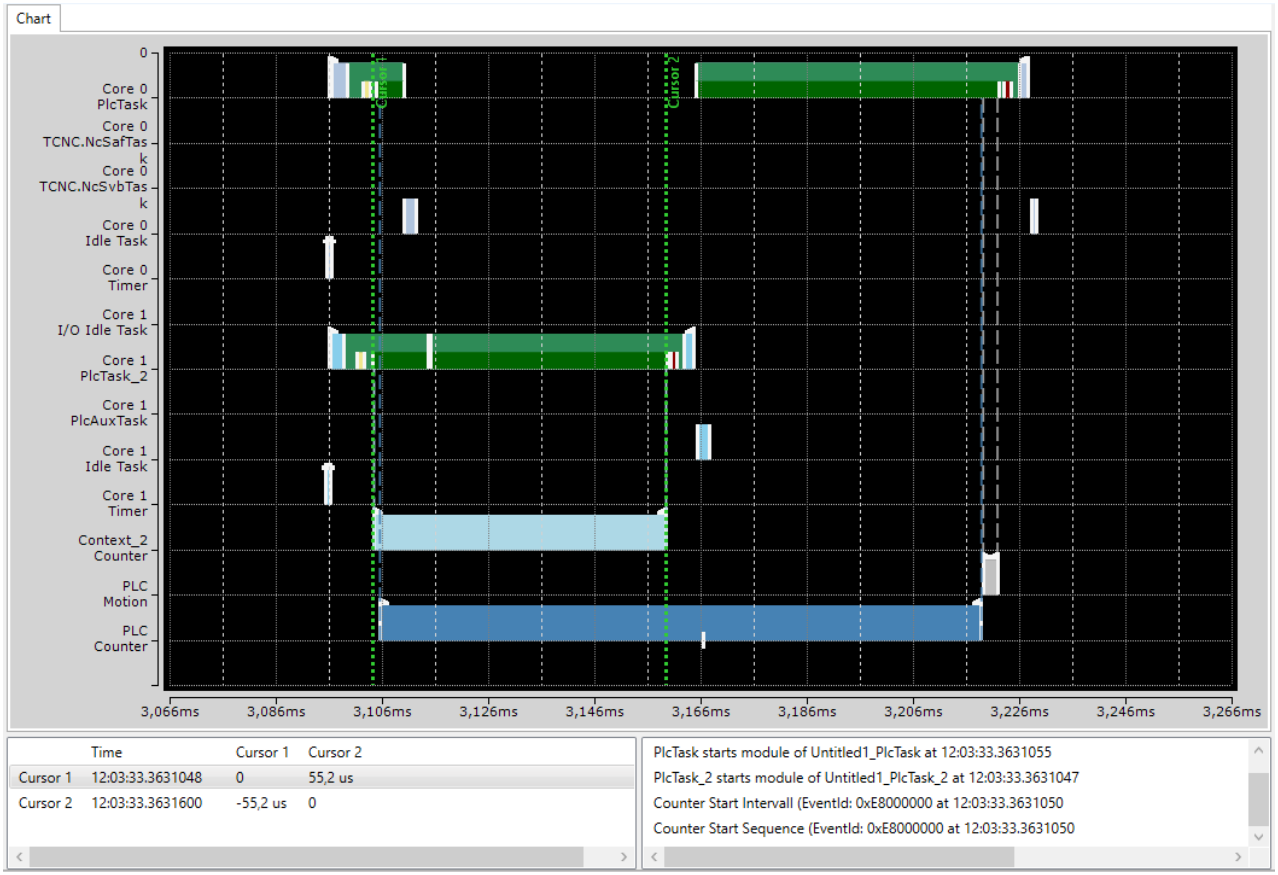
Die folgenden Eigenschaften sind für Cursors verfügbar.

Eigenschaft	Bedeutung
Cursor Info	
Color	Erlaubt das Umstellen der Farbe des aktiven Cursors.
Text	Zeigt den Text an, der am Cursor dargestellt wird.
Information	
TriggerCursor	Schaltet die Eigenschaft TriggerCursor ein, durch welche ein Cursor im Trigger-Mode an derselben Stelle im Chart-Fenster stehen bleibt und nicht an einen Zeitpunkt geheftet wird (und damit aus dem Darstellungsbereich verschwindet).

Das Event-Fenster

Im Event-Fenster werden für den jeweils aktiven Cursor alle Events aufgeführt, die zu diesem Zeitpunkt stattfinden. Für den Cursor 1 in der folgenden Abbildung sind das die folgenden Ereignisse:

- Die Task PlcTask beginnt mit der Abarbeitung des Laufzeitmoduls Untitled1.
- Die Task PlcTask_2 beginnt ebenfalls mit der Abarbeitung des Laufzeitmoduls Untitled2.
- Der Anwendungsprozess Counter startet sowohl die Sequenz als auch das Intervall.

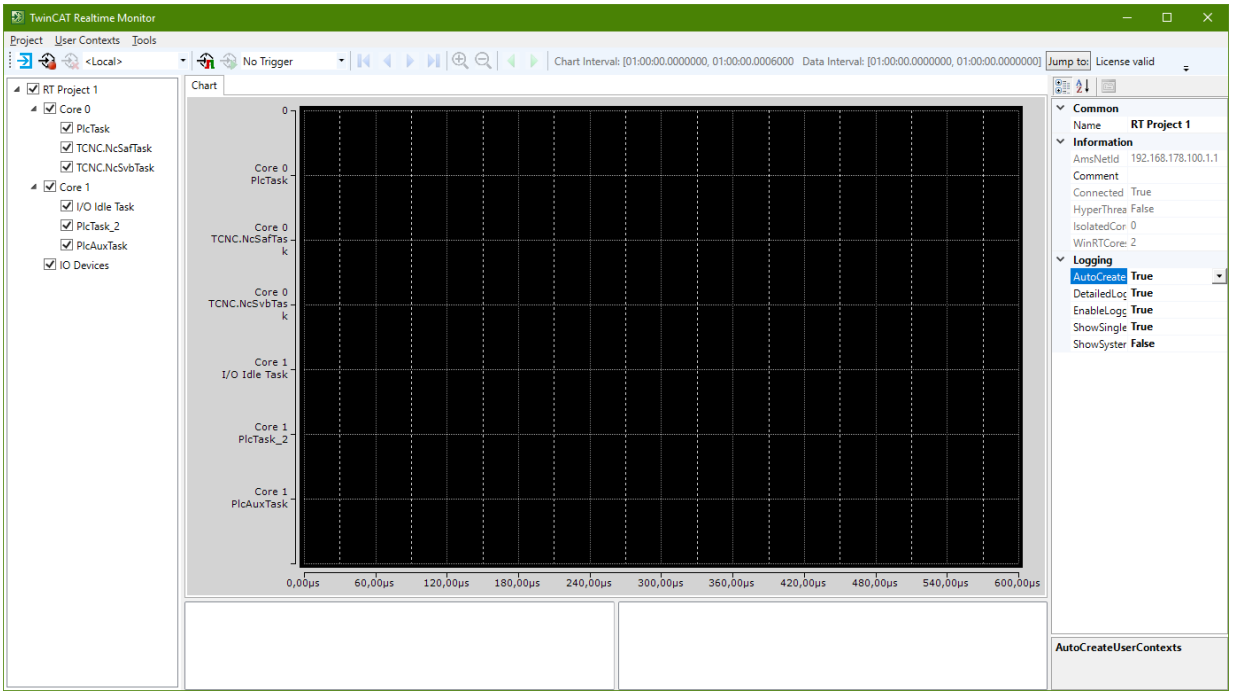


4 Quickstart

Das folgende Kapitel soll einen leichten Einstieg in die Verwendung des TwinCAT 3 Realtime Monitors ermöglichen.

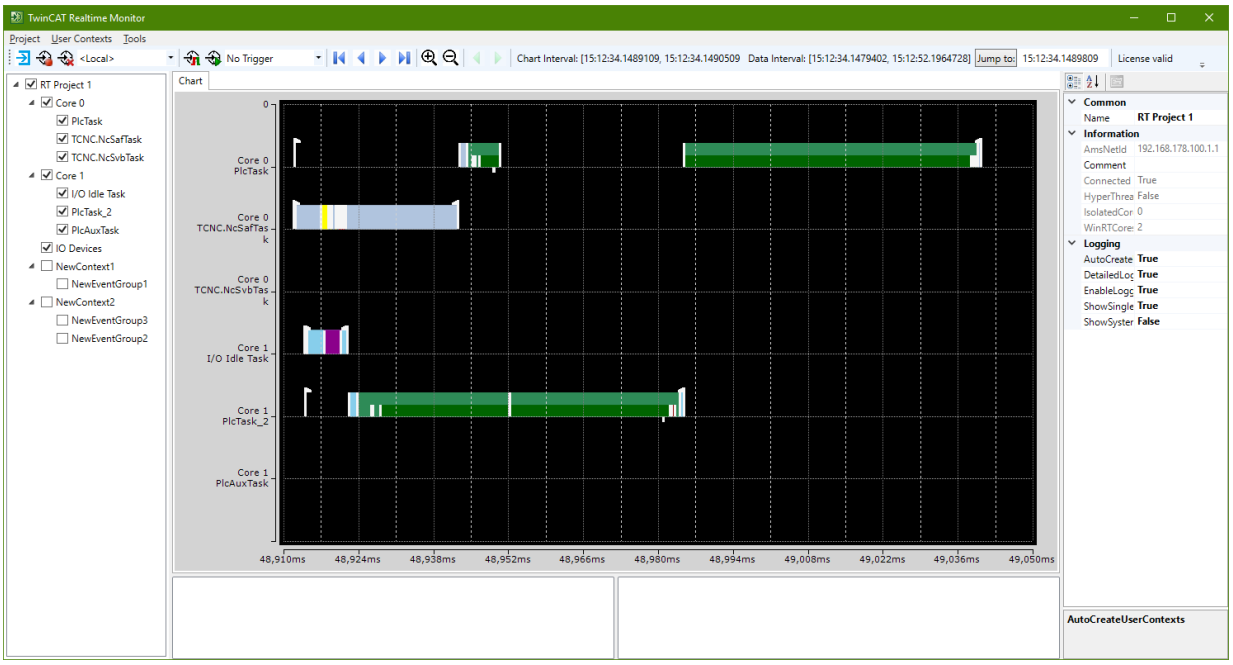
✓ Ausgangspunkt ist ein laufendes Projekt auf einer TwinCAT 3.1 Runtime der Version 3.1.4024.0 oder neuer.

1. Öffnen Sie den Realtime Monitor.
2. Legen Sie ein neues Projekt an.
   Hierzu verwenden Sie die Option **New Project** im Menü **Project** des TwinCAT 3 Realtime Monitors. Das Ändern des Projektnamens, kann über die Projekteigenschaften erfolgen (siehe Projektknoten [▶ 32]).
3. Wählen Sie nun das Zielsystem aus, welches Sie analysieren wollen. Dies erfolgt über die Toolbar des TwinCAT 3 Realtime-Monitors.
4. Es erscheint eine Abfrage, ob Sie die Konfiguration vom Zielsystem laden wollen. Bestätigen Sie diese mit „Ja“
   ⇨ Die aktive Konfiguration des Zielsystems wurde geladen und die Kontexte werden hierarchisch als Baum dargestellt.
5. Wählen Sie die Tasks aus, die Sie im TwinCAT Realtime Monitor dargestellt haben möchten.



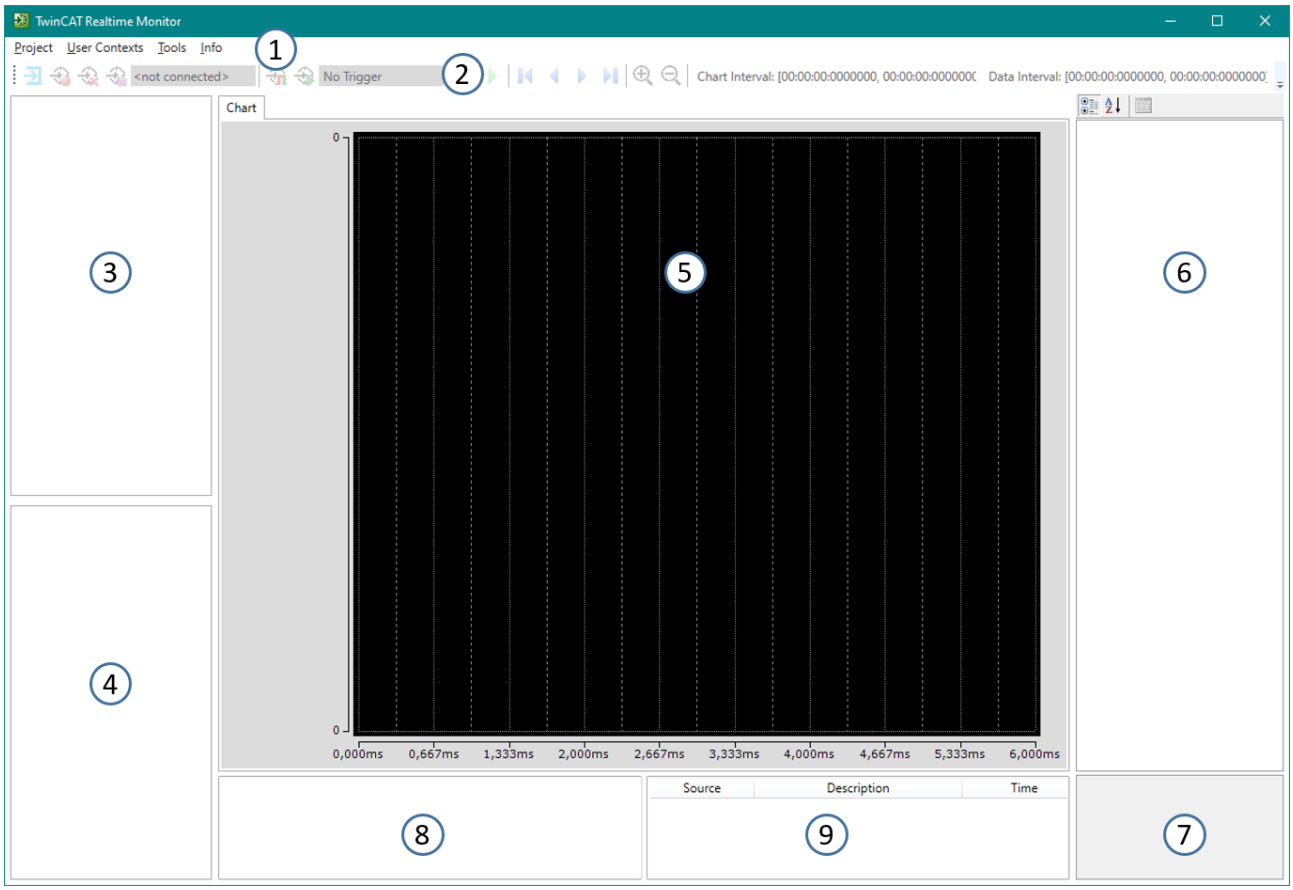
6. Markieren Sie in der Baumansicht das Projekt und setzen Sie im Eigenschaften-Fenster die Option **Detailed Logging** auf „True“ (siehe auch Markengruppen-Element [▶ 34]).
7. Wenn Sie zudem die Nutzer-Kontexte automatisch auslesen wollen, die in einem Applikationsprogramm evtl. gesetzt wurden, setzen Sie die Option **AutoCreateUserContexts** ebenfalls auf „True“.
8. Betätigen Sie den **Start Log**-Button in der Toolbar des TwinCAT 3 Realtime Monitors.

⇨ Die Aufnahme des Echtzeitverhaltens beginnt.



5 Referenz, Benutzeroberfläche

Die Benutzeroberfläche des TwinCAT 3 Realtime Monitors besteht aus den folgenden Komponenten:



1	Menüleiste
2	Symbolleiste
3	Projekt-Baum
4	Aufzeichnungsliste (Recordings)
5	Anzeigefenster
6	Eigenschaftenfenster
7	Beschreibungsanzeige der ausgewählten Eigenschaft
8	Cursor-Fenster
9	Event-Fenster

Projekt-Baum: Im Projekt-Baum des TwinCAT 3 Realtime Monitors werden die verschiedenen Zeitkontexte angezeigt.

5.1 Menüleiste

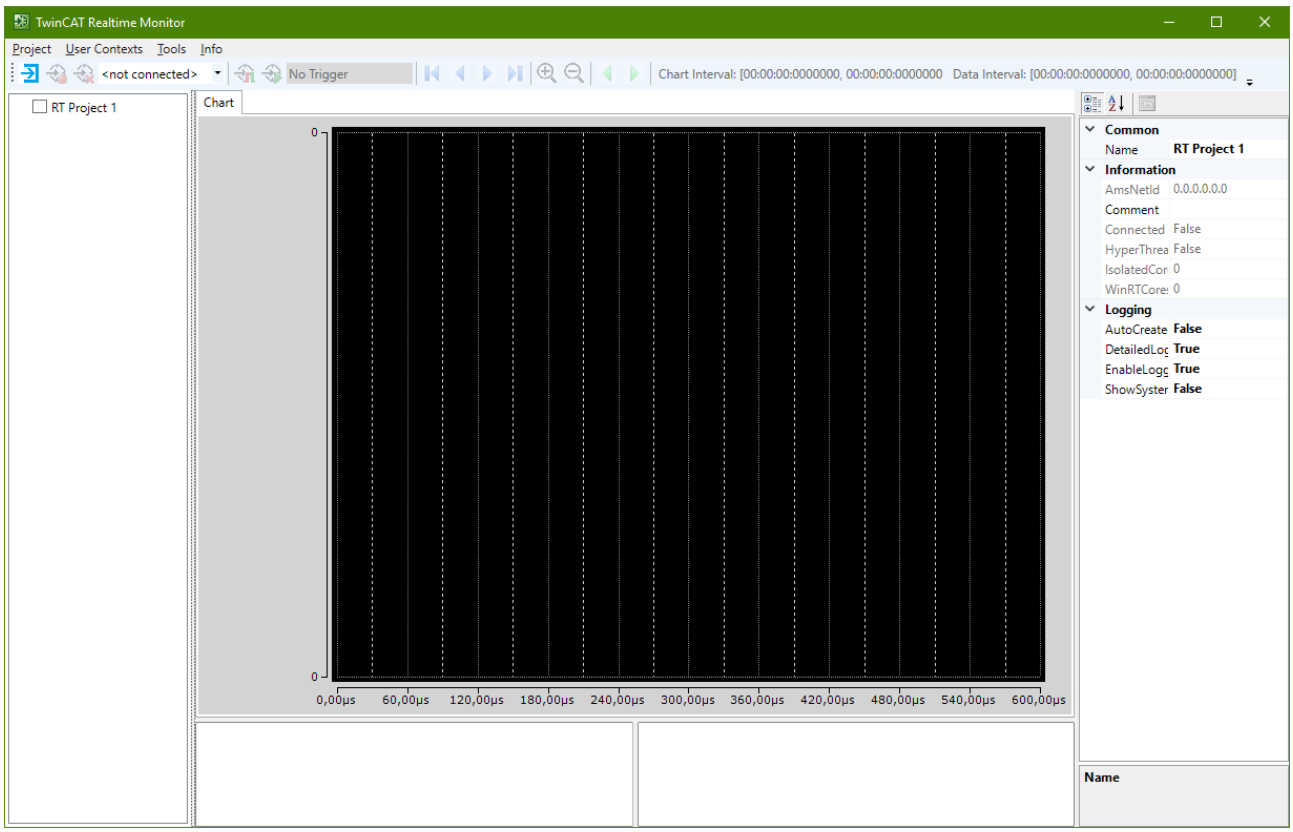
5.1.1 Project

New Project

Funktion: Der Befehl erzeugt ein neues TwinCAT 3 Realtime Monitor-Projekt.

Aufruf: Menü **Project** > **New Project**

Nachdem ein neues Projekt angelegt wurde, stellt sich der TwinCAT 3 Realtime Monitor wie folgt dar:



Open Project

Funktion: Der Befehl öffnet ein bestehendes TwinCAT 3 Realtime Monitor Projekt.

Aufruf: Menü **Project** > **Open Project**

Save Project

Funktion: Der Befehl speichert ein bestehendes TwinCAT 3 Realtime Monitor Projekt.

Aufruf: Menü **Project** > **Save Project**

Save Project as

Funktion: Der Befehl speichert ein bestehendes TwinCAT 3 Realtime Monitor Projekt unter einem zu definierenden Namen.

Aufruf: Menü **Project** > **Save Project as**

Save Project to archive

Funktion: Der Befehl speichert ein bestehendes TwinCAT 3 Realtime Monitor Projekt inklusive der darin enthaltenen Aufzeichnungen in eine Zip-Datei.

Aufruf: Menü **Project** > **Save Project to archive**

Close Project

Funktion: Der Befehl schließt ein bestehendes TwinCAT 3 Realtime Monitor Projekt.

Aufruf: Menü **Project** > **Close Project**

Import Recording

Funktion: Der Befehl importiert Aufzeichnungsdaten und fügt die Aufzeichnung am Ende der Aufzeichnungsliste ein.

Aufruf: Menü **Project** > **Import Recordings**

5.1.2 User Contexts

Import User Contexts

Funktion: Der Befehl importiert bestehende Nutzer-Kontexte in ein TwinCAT 3 Realtime Monitor Projekt. Sollten bereits (automatisch) gefundene Kontexte im Projekt enthalten sein, welche dieselben Event-Gruppen und Event-IDs enthalten, wird der Anwender gefragt, ob diese durch die gespeicherten Namen und Einstellungen ersetzt werden sollen.

Aufruf: Menü **User Contexts** > **Import User Contexts**

Export User Contexts

Funktion: Der Befehl exportiert bestehende Nutzer-Kontexte aus einem geöffneten TwinCAT 3 Realtime Monitor Projekt.

Aufruf: Menü **User Contexts** > **Export User Contexts**

Scan User Contexts

Funktion: Der Befehl scannt nach bestehenden Nutzer-Kontexten und fügt diese in ein TwinCAT 3 Realtime Monitor Projekt ein.

Aufruf: Menü **User Contexts** > **Scan User Contexts**

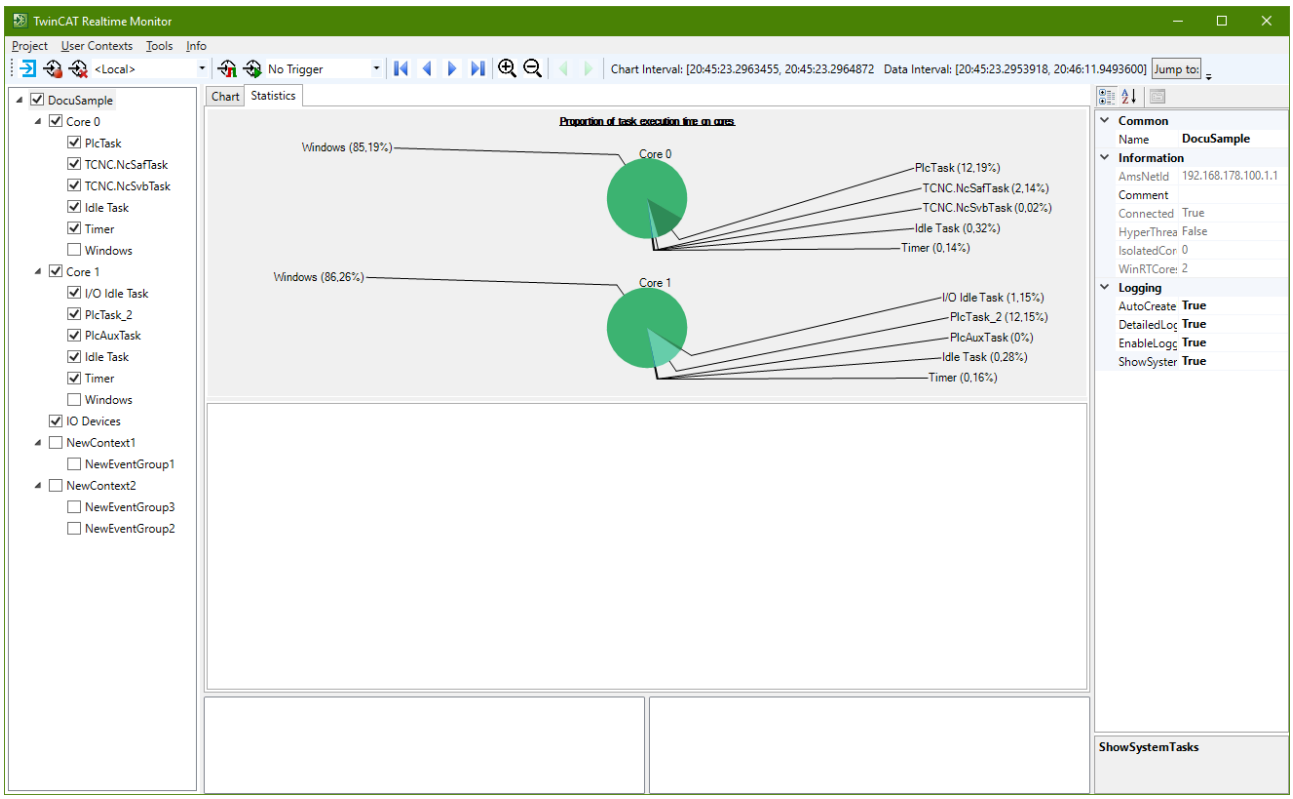
5.1.3 Tools

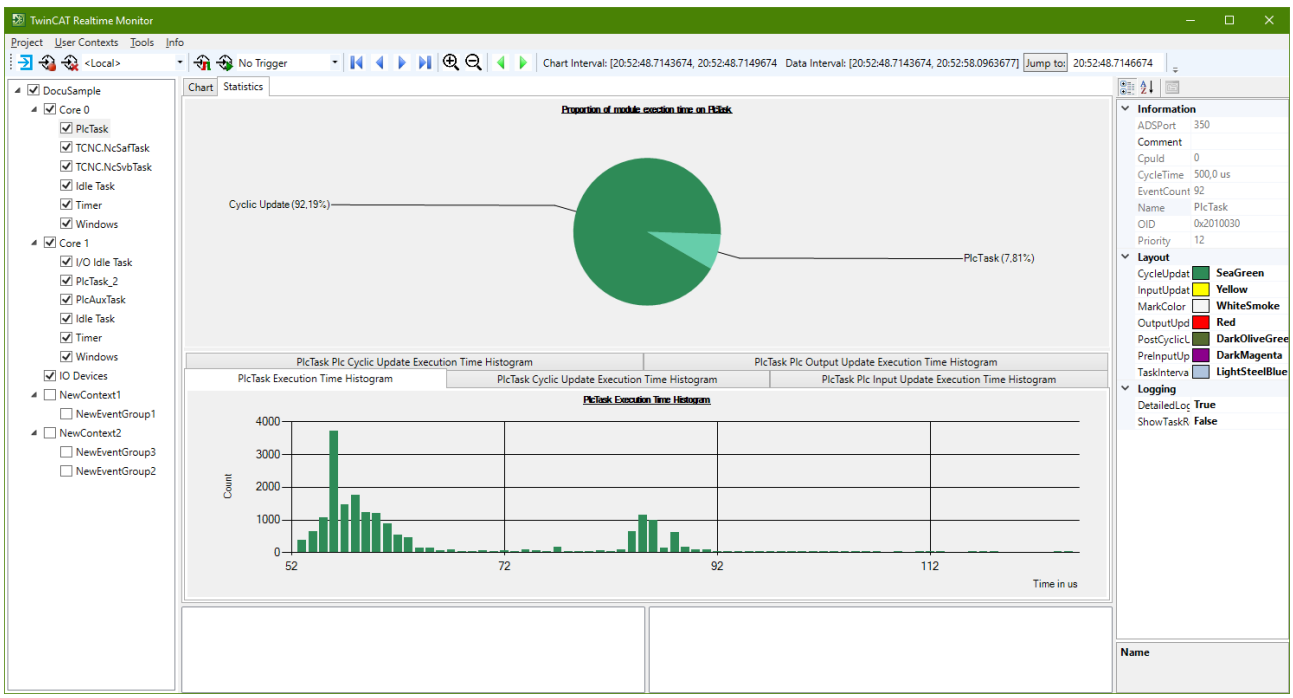
Create Statistics

Funktion: Der Befehl wertet die mit dem TwinCAT 3 Realtime Monitor aufgenommenen Marken aus und generiert eine Statistik. Diese wird im Reiter **Statistics** angezeigt.

Aufruf: Menü **Tools** > **create Statistics**

Beispiele für eine generierte Statistik:





Export Statistics

Funktion: Der Befehl exportiert die ausgewählte Statistik in eine CSV-Datei.

Aufruf: Menü **Tools** > **Export Statistics**

Set Trigger Prelude

Funktion: Der Befehl legt den Vorlauf eines Triggers fest. Es stehen die Werte 1s, 10s, 30s und 1min zur Verfügung. Ab der Version 1.2 des Realtime Monitors finden Sie diese Einstellung unter den Eigenschaften des Realtime Monitor Projektknotens (siehe Projektknoten [▶ 32]).

Aufruf: Menü **Tools** > **Set Trigger Prelude**

5.1.4 Info

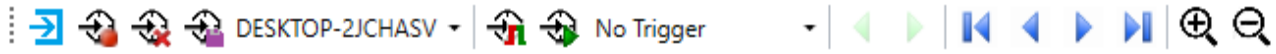
Realtime Monitor

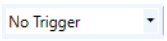
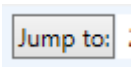
Funktion: Der Befehl öffnet ein Dialog-Fenster, welches die Versionsnummer der installierten TwinCAT 3 Realtime Monitor Version aufzeigt.

Aufruf: Menü **Info** > **Realtime Monitor**

5.2 Symbolleiste - Realtime Monitor Toolbar

Die TwinCAT 3 Realtime Monitor Toolbar stellt die folgenden Befehle zur Verfügung.



	Laden der Projekt-Konfiguration vom eingestellten Zielsystem
	Starten der Aufnahme
	Stoppen der Aufnahme
	Löschen der dargestellten Daten und löschen der aufgenommen Daten
	Speichern der aufgenommen Daten innerhalb des Projekts
	Auswahl des Zielsystems
	Start des Triggerns auf Live-Daten
	Start des Triggerns auf aufgenommene Daten
	Auswahl eines Triggers
	Manueller Sprung zum nächsten Trigger-Event
	Manueller Sprung zum vorhergehenden Trigger-Event
	Sprung zum Anfang der Darstellung
	Bewegung der Darstellung nach links
	Bewegung der Darstellung nach rechts
	Sprung zum Ende der Darstellung
	Zoom In
	Zoom Out
Chart Interval	Zeitintervall des im aktuellen Ausschnitt dargestellten Bereiches
Data Interval	Zeitintervall der aufgenommenen Daten
	Sprung zum Zeitpunkt der im Auswahlfeld dahinter angegebenen wird
	Eingabefeld zur Eingabe eines Zeitpunktes

5.3 Projektbaum

Der Projektbaum stellt alle Markengruppen und ihre Zuordnung zu Kontexten hierarchisch dar. Für die System-Tasks wird automatisch ein Eintrag mit dem entsprechenden Namen der Task im Baum angelegt. System-Tasks werden nach Ihrer Zuordnung zu Rechenkernen zu entsprechenden Kontexen zusammengefasst.

Für nutzerbezogene Markengruppen wird ebenfalls ein Eintrag im Projektbaum erzeugt. Die Zuordnung zu Kontexten entsteht - je nach verwendetem Aufruf - im Anwenderprogramm (siehe FB_RTMon_LogMark [▶ 37] oder FB_RTMon_LogMarkBase [▶ 40]), entweder bezogen auf den ADS-Port des Anwenderprogramms oder anhand einer nutzerdefinierten Kontext-ID.

Erzeugt werden die nutzerbezogenen Knoten entweder manuell durch die Verwendung der Kontext-Menü-Einträge (siehe Projektbaum [▶ 29]) oder automatisch sofern die Option **AutoCreateUserContexts** (siehe Projektknoten [▶ 32]) aktiviert oder die Option User Contexts [▶ 25] aufgerufen wurde.

Die Benennung der nutzerbezogenen Knoten mit sprechenden Namen erfolgt anhand ihrer Eigenschaften-Seite (siehe Kontextknoten [▶ 33] bzw. Markengruppen-Element [▶ 34]).

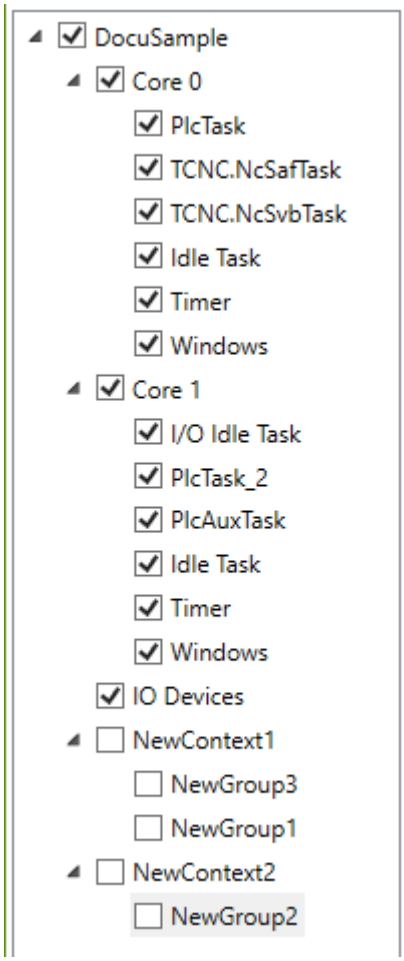
Kontext-Menü-Einträge im Projektbaum

Die folgende Tabelle zeigt alle Kontextmenü-Einträge im Projektbaum (und den Knotentyp auf dem diese zur Verfügung stehen) auf.

Befehl	Knotentyp	Bedeutung
Add New User Context	Projektknoten	Fügt einen nutzerbezogenen Kontext hinzu.
Import User Context	Projektknoten	Importiert einen nutzerbezogenen Kontext inklusiver aller Unterelemente.
Add New User Group	nutzerbezogener Kontext-Knoten	Fügt eine nutzerbezogene Markengruppe hinzu.
Remove User Context	nutzerbezogener Kontext-Knoten	Löscht einen nutzerbezogenen Kontext.
Export User Context	nutzerbezogener Kontext-Knoten	Exportiert einen nutzerbezogenen Kontext inklusive aller Unterelemente.
Remove User Group	nutzerbezogener Markengruppen-Knoten	Löscht eine nutzerbezogene Markengruppe.

Beispiel:

Die folgende Abbildung zeigt die Darstellung eines Projektbaums, wie er nach dem Starten der Aufzeichnung automatisch erzeugt wird (mit aktivierter Option **AutoCreateUserContexts**). Neben den System-Tasks verteilt auf die Rechenkerne Core 0 und Core 1 werden auch 3 nutzerbezogene Markengruppen erzeugt, die hier aber noch nicht benannt wurden.



5.4 Aufzeichnungsliste

In einem Realtime Monitor Projekt können mehrere Aufzeichnungen verwaltet werden. Diese werden in der Aufzeichnungsliste dargestellt.

Kontextmenü-Einträge in der Aufzeichnungsliste

Die folgende Tabelle zeigt alle Kontextmenü-Einträge:

Befehl	Bedeutung
Load Recording to Chart	Lädt die ausgewählte Aufzeichnung in das Anzeigefenster.
Remove Recording from List	Entfernt die Aufzeichnung aus der Liste. Die Aufzeichnungsdatei bleibt auf der Festplatte erhalten.
Delete Recording File	Entfernt die Aufzeichnung aus der Liste und löscht die entsprechende Datei von der Festplatte.

Speichern einer Aufzeichnung

Nachdem Sie eine Aufzeichnung mit dem Realtime Monitor abgeschlossen haben, können Sie die Daten durch den Button  (**Save collected data within project**) in der Symbolleiste [▶ 27] als Aufzeichnung (Recording) innerhalb des Projekts speichern.

Hinzufügen einer Aufzeichnung

Um bereits aufgenommene Daten zu einem Realtime-Monitor-Projekt hinzuzufügen, verwenden Sie den Menüeintrag „Import Recording“ im Menü *Projekt*. Die Aufzeichnung wird dann am Ende der Liste hinzugefügt und entsprechend aufsteigend nummeriert.

Die Datendateien des Realtime Monitors enthalten nur den Namen des Projektes. Die Datendateien selbst haben keinen Namen, sondern nur einen Zeitstempel. Werden Aufzeichnungen also aus dem Projekt entfernt und später wieder hinzugefügt, unterscheiden sich die angezeigten Namen (z. B. die angezeigte Aufzeichnungsnummer). Der Start- und Endzeitpunkt einer geladenen Aufzeichnung wird in den Eigenschaften des Projektes angezeigt (siehe Projektknoten [▶ 32]).

Entfernen einer Aufzeichnung

Soll eine Aufzeichnung aus dem Projekt entfernt aber nicht von der Festplatte gelöscht werden, wählen Sie diese Aufzeichnung in der Aufzeichnungsliste aus, so dass sie markiert ist. Anschließend verwenden Sie den Kontext-Menü-Eintrag „Remove Recording from List“.

Löschen einer Aufzeichnung

Soll eine Aufzeichnung sowohl aus dem Projekt als auch von der Festplatte gelöscht werden, wählen sie diese Aufzeichnung in der Aufzeichnungsliste aus, so dass diese markiert ist. Anschließend verwenden Sie den Kontextmenü-Eintrag „Delete Recording File“.

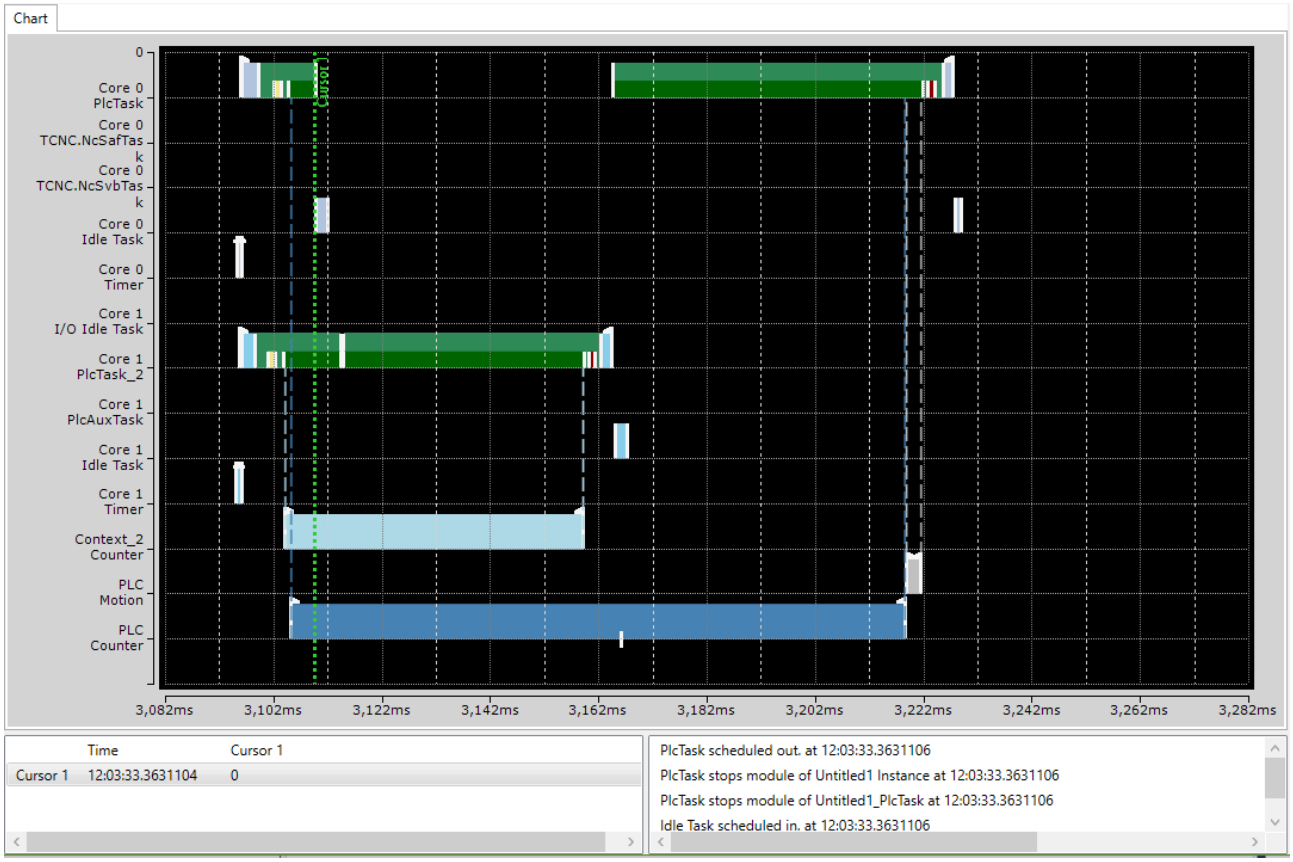
Voraussetzung:

Die Aufzeichnungsfunktionalität steht auf der Realtime Monitor Version 1.2 zur Verfügung.

5.5 Anzeigefenster

Im Anzeigefenster (Chart) werden die (Zeit-)Marken, sortiert nach den einzelnen Markengruppen, aufgetragen über der Zeit dargestellt.

Mithilfe der Funktionen in der Symbolleiste (siehe Symbolleiste - Realtime Monitor Toolbar [▶ 27]) bzw. analog auch unter Verwendung der Maus bzw. ähnlicher Bediengeräte, kann innerhalb der Darstellung des Anzeigefensters navigiert bzw. die Darstellung vergrößert / verkleinert werden.



Per Kontext-Menü-Einträge ist es im Anzeigefenster möglich, Cursors zu setzen, zu löschen oder zu verschieben um Zeitmessungen oder Analysen vorzunehmen (siehe hierzu Verwendung von Cursors [▶ 17]).

5.6 Eigenschaften-Fenster

Im Eigenschaften-Fenster werden die Eigenschaften des jeweils aktiven (markierten) Elements des Projektbaums aufgezeigt.

Die im Bereich **Logging** aufgeführten Eigenschaften gelten dabei immer auch für alle Unterelemente des Baumes. Der Wert **Different Settings** zeigt an, dass sich die Werte der Unterelemente unterscheiden. Durch ein Ändern des Wertes, werden dann auch die Werte der Unterelemente mit geändert.

5.6.1 Projektknoten

Folgende Einstellungen stehen auf dem Projekt-Knoten des TwinCAT 3 Realtime Monitors zur Verfügung:

Eigenschaft	Bedeutung
Common	
Comment	Kommentar/Bemerkung zum Projekt
Name	Name des Projektes
Information	
AmsNetId	AmsNetId des Zielsystems
Connected	Verbindungsstatus zum Zielsystem
HyperTreading	Zeigt an, ob Hyperthreading aktiv ist.
Isolated Cores	Zeigt die Anzahl der im Projekt verwendeten isolierten Kerne an.
WinRTCores	Zeigt die Anzahl der im Projekt verwendeten Windows-Echtzeit-Kerne an
XAR Version	Zeigt die TwinCAT Version des Zielsystems an.
Live Trigger	
Store Trigger Data	Speichert Trigger-Daten als Aufzeichnung (Recording).
Trigger Prelude	Definiert die Zeit in Sekunden vor einem Trigger Event (Trigger-Vorlauf).
Trigger Samples	Definiert die Anzahl der Trigger Samples, die gespeichert werden sollen.
Loaded Recording	
Loaded Recording	Name der aktuell geladenen Aufzeichnung
Recording End	Endzeitpunkt der Aufzeichnung
Recording Start	Startzeitpunkt der Aufzeichnung
Logging	
DetailedLogging	Schaltet das detaillierte Logging ein.
EnableLogging	Aktiviert das Logging.
ReduceOnZoom	Reduziert die Darstellungstiefe beim Heraus-Zoomen (direkt nebeneinander liegende Marken werden als ein Balken zusammengefasst), um die Performance zu steigern.
ShowSystemTasks	Zeigt auch die System-Tasks an.

Die im Bereich **Logging** aufgeführten Eigenschaften gelten immer auch für alle Unterelemente. Mit anderen Worten gelten diese Eigenschaften auf Projekt-Ebene für das gesamte Realtime Monitor Projekt. Steht als Wert hinter einer der Eigenschaften aus dem Bereich Logging ein „Different Settings“ bedeutet dies, dass sich die Werte in den einzelnen Unterknoten unterscheiden. Durch ein Ändern des Wertes auf Projektebene werden die Werte für alle Unterelemente gesetzt.

5.6.2 Kontextknoten

Folgende Einstellungen stehen auf dem Kontext-Knoten des TwinCAT 3 Realtime Monitors zur Verfügung. Diese unterscheiden sich nach Echtzeitkontexten (hier entspricht der Kontext einem Rechnerkern) und Anwendungskontexten.

Echtzeitkontext:

Eigenschaft	Bedeutung
Information	
BaseTime	Basis-Zeit des Kerns
Comment	Kommentar
DefaultCore	Zeigt an, ob es sich um den Default-Kern handelt
Id	Zeigt die ID des Kerns
Name	Zeigt den Namen des Kerns
RT_Percentage	Zeigt die eingestellte maximale Echtzeitauslastung
Type	Zeigt den Typ des Kerns an (WindowsRT/ Isolierter Kern)
Logging	
DetailedLogging	Schaltet das detaillierte Logging an
EnableLogging	Schaltet das Logging an / aus

Anwendungskontext:

Eigenschaft	Bedeutung
Information	
Comment	Kommentar
ContextId	ContextId, die bei den Marken übergeben wurde
Name	Name des Kontexts

Bei der Verwendung des Bausteins FB_RTMon_LogMark [▶ 37] wird als ContextId automatisch die Port-Nummer des SPS-Laufzeitmoduls eingestellt.

5.6.3 Markengruppen-Element

Folgende Einstellungen stehen auf den Markengruppen- / Prozess-Knoten des TwinCAT 3 Realtime Monitors zur Verfügung. Diese unterscheiden sich nach Echtzeit-Tasks und Anwendungs-Prozessen / -Marken.

Echtzeit-Tasks:

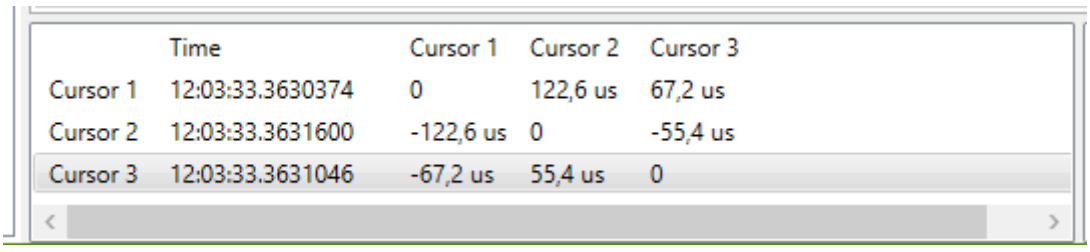
Eigenschaft	Bedeutung
Information	
ADSPort	ADS-Port der Task
Comment	Kommentar
CpuId	CpuId auf dem die Task ausgeführt wird
CycleTime	Zykluszeit der Task
EventCount	Anzahl der Ausführungen (innerhalb der Aufnahmezeit)
Name	Name der Task
OID	ObjektId der Task
Priority	Eingestellte Priorität
Layout	
CycleUdateColor	Farbe des CycleUpdates einer Task (Standard: grün)
InputUpdateColor	Farbe des InputUpdates der Task (Standard: gelb)
MarkColor	Marken-Farbe (Standard: weiß)
OutputUpdateColor	Farbe des OutputUpdates der Task (Standard: rot)
PostCyclicUpdateColor	Farbe des PostCyclicUpdats der Task (Standard: dunkelgrün)
PreInputUpdateColor	Farbe des PreInputUpdates der Task (Standard Magenta)
TaskIntervallColor	Farbe der Taskintervall-Markierung (Standard: hellblau)
Logging	
DetailedLogging	Aktivieren des detaillierten Loggings
ShowTaskReference	Anzeigen der Task-Referenzen

Anwender-Prozesse:

Eigenschaft	Bedeutung
Information	
Comment	Kommentar
EventCount	Anzahl der Ausführungen (innerhalb der Aufnahmezeit)
GroupId	ID der Markengruppe / des Prozesses
Name	Name des darzustellenden Prozesses
Layout	
EventIntervallColor	Farbe für das Intervall / das aktive Ausführen des Prozesses (Standard: blau)
MarkColor	Farbe der Marken (Standard: weiß)
Logging	
ShowSingleMarker	Aktiviert das Anzeigen einzelner Marken
ShowTaskReference	Aktiviert das Anzeigen der Task-Referenzen (Zuordnung der Prozess-Marken zu Echtzeittasks durch gestrichelte Linien)

5.7 Cursor-Fenster

Alle erstellten Cursors werden im Cursor-Fenster angezeigt.



	Time	Cursor 1	Cursor 2	Cursor 3
Cursor 1	12:03:33.3630374	0	122,6 us	67,2 us
Cursor 2	12:03:33.3631600	-122,6 us	0	-55,4 us
Cursor 3	12:03:33.3631046	-67,2 us	55,4 us	0

Ein Doppel-Klick auf einen Cursor sorgt dafür, dass die Darstellung innerhalb des Charts genau an die Stelle springt, an welcher der Cursor steht. Der Cursor wird mittig im Darstellungsbereich angezeigt.

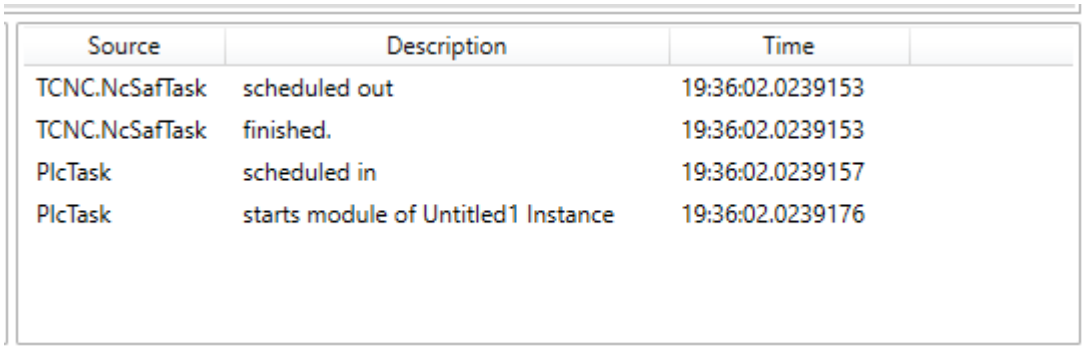
Mithilfe des Kontexmenü-Eintrags **Remove Cursor** ist es möglich, den jeweils markierten Cursor zu löschen.

Die Verwendung von Cursors ist ausführlich beschrieben unter Verwendung von Cursors [▶ 17].

5.8 Event-Fenster

Im Event-Fenster werden für den jeweils aktiven Cursor alle Events aufgeführt, die zu diesem Zeitpunkt stattfinden. Für den Cursor 1 in der folgenden Abbildung sind das die folgenden Ereignisse:

- Die NC-SAF-Task wird beendet.
- Die NC-SAF-Task ist beendet.
- Der Sheduler startet die PlcTask.
- Die Task PlcTask beginnt mit der Abarbeitung des Laufzeitmoduls Untitled1.



Source	Description	Time
TCNC.NcSafTask	scheduled out	19:36:02.0239153
TCNC.NcSafTask	finished.	19:36:02.0239153
PlcTask	scheduled in	19:36:02.0239157
PlcTask	starts module of Untitled1 Instance	19:36:02.0239176

6 SPS API

6.1 Funktionsbausteine

6.1.1 FB_RTMon_LogMark


```
FUNCTION_BLOCK FB_RTMon_LogMark
VAR_INPUT
    nContextId        : UINT := TwinCAT_SystemInfoVarList._AppInfo.AdsPort;
    bLogCallingTask   : BOOL := TRUE; // specifies whether a reference to the calling task should be
set with each mark
END_VAR
VAR_OUTPUT
    bError            : BOOL;          // TRUE if an error occurred
    hrErrorCode       : HRESULT;      // outputs the error code which occurred
END_VAR
```

Beschreibung:

Der FB_RTMon_LogMark ist ein erweiterter Funktionsbaustein, der das Setzen von „einfachen“ (Zeit-) Marken ermöglicht.

Bei „einfachen“ Marken wird automatisch der Kontext des aufrufenden Anwenderprogramms verwendet. Die möglichen Markentypen (Sequenz-Start & -Stopp, Intervall-Start & Stopp bzw. Marke) werden über einzelne Methoden zur Verfügung gestellt. Lediglich die Marken-ID (Markengruppe) muss vom Anwender mit übergeben werden. Diese wird verwendet um den Prozess der dargestellt werden soll zu identifizieren.

Optional steht noch eine Event-ID zur Verfügung, in welcher der Anwender noch ein Anwender-Datum mit übergeben kann (z. B. Zustand einer Statemachine, Fehlermeldung …)

6.1.1.1 LogIntervalStart


```
// Starts logging interval
METHOD LogIntervalStart : HRESULT
VAR_INPUT
    nGroupId    : UINT; // Defines the group to which the interval belongs
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Beschreibung

Die Methode erstellt eine Marke mit einem Intervallstart für die übergebene Marken-ID.

Parameter:

nGroupId: Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.

nEventId: optionale EventId.

6.1.1.2 LogIntervalStop


```
METHOD LogIntervalStop : HRESULT
VAR_INPUT
    nGroupId    : UINT; // Defines the group to which the interval belongs
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Beschreibung

Die Methode erstellt eine Marke mit einem Intervallstopp für die übergebene Marken-ID.

Parameter:

nGroupId: Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.

nEventId: optionale EventId.

6.1.1.3 LogMark


```
// Logs a mark without start/stop
METHOD LogMark : HRESULT
VAR_INPUT
    nGroupId    : UINT; // Defines the group to which the mark belongs
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Beschreibung

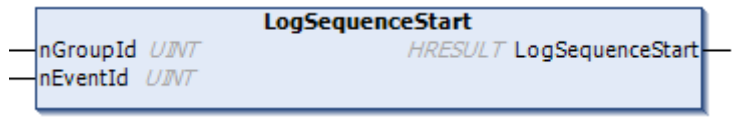
Die Methode erstellt eine Marke für die übergebene Marken-ID. Optional kann die Event-ID benutzt werden, um zwischen verschiedenen Anwender-Events zu unterscheiden bzw. um zusätzliche Daten (als UINT formatiert) mit in der Darstellung des TwinCAT 3 Realtime Monitors anzuzeigen.

Parameter:

nGroupId: Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.

nEventId: optionale EventId.

6.1.1.4 LogSequenceStart




```
// Starts logging sequence
METHOD LogSequenceStart : HRESULT
VAR_INPUT
    nGroupId    : UINT; // Defines the group to which the sequence belongs
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Beschreibung

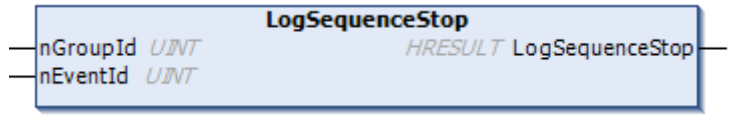
Die Methode erstellt eine Marke mit einem Sequence-Start für die übergebene Marken-ID.

Parameter:

nGroupId: Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.

nEventId: optionale EventId.

6.1.1.5 LogSequenceStop




```
// Stops logging sequence
METHOD LogSequenceStop : HRESULT
VAR_INPUT
    nGroupId    : UINT; // Defines the group to which the sequence belongs
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Beschreibung

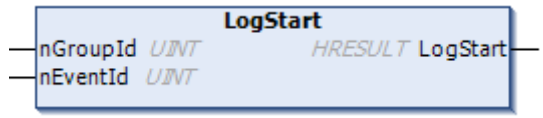
Die Methode erstellt eine Marke mit einem Sequence-Stopp für die übergebene Marken-ID.

Parameter:

nGroupId: Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.

nEventId: optionale EventId.

6.1.1.6 LogStart




```
// Starts logging sequence and interval
METHOD LogStart : HRESULT
VAR_INPUT
    nGroupId    : UINT; // Defines the group to which the sequence and intervall belong
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Beschreibung

Die Methode erstellt eine Marke mit einem Sequence- und Intervall-Start für die übergebene Marken-ID.

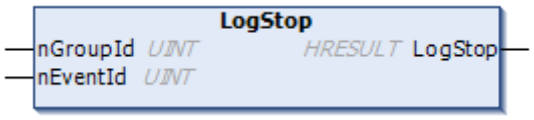
Somit bildet diese Marke den Zeitpunkt eines Prozesses ab, in dem er sofort aktiv / gestartet ist.

Parameter:

nGroupId: Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.

nEventId: optionale EventId.

6.1.1.7 LogStop




```
// Stops logging sequence and interval
METHOD LogStop : HRESULT
VAR_INPUT
    nGroupId    : UINT; // Defines the group to which the sequence and intervall belong
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Beschreibung

Die Methode erstellt eine Marke mit einem Sequence- und Intervall-Stopp für die übergebene Marken-ID.

Somit bildet diese Marke den Zeitpunkt eines Prozesses ab, in dem er direkt beendet wird.

Parameter:

nGroupId: Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.

nEventId: optionale EventId.

6.1.2 FB_RTMon_LogMarkBase


```
FUNCTION_BLOCK FB_RTMon_LogMarkBase
VAR_INPUT
END_VAR
VAR_OUTPUT
    bError        : BOOL;        // TRUE if an error occurred
    hrErrorCode   : HRESULT;     // outputs the error code which occurred
END_VAR
```

Beschreibung:

Der FB_RTMon_LogMarkBase ist ein Basis-Funktionsbaustein, der das Setzen von (Zeit-) Marken ermöglicht.

Im Gegensatz zum Funktionsbaustein FB_RTMon_LogMark [▶ 37], muss hier die Context-ID selbst übergeben werden. Mit dieser ist es möglich, die darzustellenden Prozesse zu gruppieren (z. B. nach Prozessart oder Funktionseinheit).

Optional steht noch eine Event-ID zur Verfügung, in der der Anwender noch ein Anwender-Datum mit übergeben kann (z. B. Zustand einer Statemachine, Fehlermeldung).

6.1.2.1 LogMark


```
METHOD LogMark : HRESULT
VAR_INPUT
    nContextId : UINT;     // defines the context
    nGroupId   : UINT;     // defines the group inside the context
```

```
    nEventId   : UINT;     // defines the specific event inside the group
    nMarkCtrl  : UDINT;    // mask for mark options (listed in TcMarkOption)
END_VAR
```

Beschreibung

Die Methode erstellt eine Marke für die übergebene Markengruppen-ID. Der Markentyp wird anhand des Parameters nMarkCtrl (siehe TcMarkOption [▶ 42]) übergeben.

Optional kann die Event-ID benutzt werden, um zwischen verschiedenen Anwender-Events zu unterscheiden bzw. um zusätzliche Daten (als UINT formatiert) mit in der Darstellung des TwinCAT 3 Realtime Monitors anzuzeigen.

Parameter:

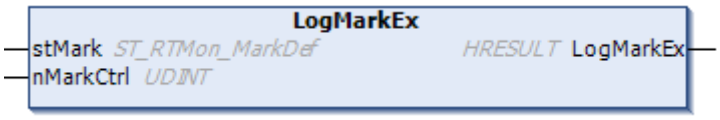
nContextId: definiert die Kontext-Id, unter der die Marke im TwinCAT 3 Realtime Monitor gruppiert werden soll.

nGroupId: Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.

nEventId: optionale EventId.

nMarkCtrl: definiert den Markentyp.

6.1.2.2 LogMarkEx




```
METHOD LogMarkEx : HRESULT
VAR_INPUT
    stMark       : ST_RTMon_MarkDef;
    nMarkCtrl    : UDINT; // mask for mark options (listed in TcMarkOption)
END_VAR
```

Beschreibung

Die Methode erstellt eine Marke. Die Markendefinition erfolgt anhand des Datentyps ST_RTMon_MarkDef [▶ 41]. Der Markentyp wird anhand des Parameters nMarkCtrl (siehe TcMarkOption [▶ 42]) übergeben.

Parameter:

stMark: Übergabeparameter für eine definierte Marke, die geschrieben werden soll.

nMarkCtrl: definiert den Markentyp.

6.2 Datentypen

6.2.1 ST_RTMon_MarkDef

Datentyp, welcher eine Marke repräsentiert.

```
// defines a mark
TYPE ST_RTMon_MarkDef:
STRUCT
    nContextId : UINT;      // defines the context
    nGroupId : UINT;       // defines the group inside the context
    nEventId : UINT;       // defines the specific event inside the group
END_STRUCT
END_TYPE
```

Beschreibung

Mithilfe dieses Datentyps ist es möglich, eine generische Marke (noch ohne deren Typ) zu definieren. Diese wird dann in der Methode LogMarkEx [▶ 41] des Funktionsbausteins FB_RTMon_LogMarkBase [▶ 40] zusätzlich zum Markentyp mit übergeben.

nContextId: Mithilfe der ContextId können Markengruppen, also darzustellende Prozesse, gruppiert werden (z. B. nach Prozessart oder Funktionseinheit).

nGroupId: Definiert den darzustellenden Prozess / das darzustellende Prozessereignis.

nEventId: Optionales Anwenderdatum. Es kann verwendet werden, um z. B. den Zustand einer Statemachine oder Errorcodes im TwinCAT 3 Realtime Monitor darzustellen.

Sowohl die ContextId als auch die GroupId können im TwinCAT 3 Realtime Monitor mit Namen versehen werden. Diese können über die Funktionen User Contexts [▶ 25] bzw. User Contexts [▶ 25] exportiert bzw. importiert werden, sodass sie auch für die weitere Aufzeichnung zur Verfügung stehen.

6.3 Globale Konstanten

6.3.1 TcMarkOption

Die Konstanten in dieser globalen Variablenliste definieren die möglichen Markentypen (siehe Darstellung im Realtime Monitor [▶ 13]).

```
VAR_GLOBAL CONSTANT
    Start             : UDINT := 16#E0000000;
    Stop              : UDINT := 16#C0000000;
    SequenceStart     : UDINT := 16#A0000000;
    SequenceStop      : UDINT := 16#80000000;
    IntervalStart     : UDINT := 16#60000000;
    IntervalStop      : UDINT := 16#40000000;
    RefToCaller       : UDINT := 16#08000000; // reference to caller
END_VAR
```

Zusätzlich zu den Markentypen ist die Option RefToCaller definiert, die es ermöglicht, dass im TwinCAT3 Realtime Monitor die Task-Referenzen angezeigt werden können. Soll diese Option eingeschaltet werden, muss diese mit dem gewünschten Markentyp per ODER verknüpft werden.

Beispiel:

```
fbLogMark.LogMarkEx(markCounter, TcMarkOption.Start OR TcMarkOption.RefToCaller);
```

Das Beispiel zeigt das Setzen einer Marke „markCounter“ mit dem Markentyp „Start“ und der Option „RefToCaller“.

Sollen die Task-Referenzen im TwinCAT3 Realtime Monitor angezeigt werden, muss die Option **Show Task Reference** (siehe Markengruppen-Element [▶ 34]) eingeschaltet sein.

7 C++ API

7.1 Datentypen

7.1.1 TcMark16

Datentyp, der eine Marke repräsentiert.

```
typedef struct {
USHORT ContextId;
USHORT GroupId;
USHORT EventId;
} TcMark16;
```

Beschreibung:

Mithilfe dieses Datentyps ist es möglich, eine generische Marke (noch ohne deren Typ) zu definieren.

ContextId: Mithilfe der ContextId können Markengruppen, also darzustellende Prozesse, gruppiert werden (z. B. nach Prozessart oder Funktionseinheit).

GroupId: Definiert den darzustellenden Prozess / das darzustellende Prozessereignis.

EventId: Optionales Anwenderdatum. Es kann verwendet werden, um z. B. den Zustand einer Statemachine oder Errorcodes im TwinCAT 3 Realtime Monitor darzustellen.

Sowohl die ContextId als auch die GroupId können im TwinCAT 3 Realtime Monitor mit Namen versehen werden. Diese können über die Funktionen User Contexts [▶ 25] bzw. User Contexts [▶ 25] exportiert bzw. importiert werden, so dass sie auch für die weitere Aufzeichnung zur Verfügung stehen.

7.2 Klassen

7.2.1 CTcLogMark

```
CTcLogMark(USHORT nContextId, ITComObjectServer* ipSrv = NULL);
```

Beschreibung:

Die Klasse CTcLogMark ist eine C++-Klasse die es ermöglicht, (Zeit-)Marken aus C++-Applikationscode heraus zu setzen, dass diese mit dem TwinCAT 3 Realtime Monitor dargestellt werden können.

7.2.1.1 LogIntervalStart

```
virtual HRESULT LogIntervalStart(USHORT GroupId, USHORT EventId);
```

Beschreibung:

Die Methode erstellt eine Marke mit einem Intervallstart für die übergebene Marken-ID.

Parameter:

GroupId: Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.

EventId: optionale EventId.

7.2.1.2 LogIntervalStop

```
virtual HRESULT LogIntervalStop(USHORT GroupId, USHORT EventId);
```

Beschreibung:

Die Methode erstellt eine Marke mit einem Intervallstopp für die übergebene Marken-ID.

7.2.1.3 LogMark

```
virtual HRESULT LogMark(USHORT GroupId, USHORT EventId, ULONG CtrlId);
```

Beschreibung:

Die Methode erstellt eine Marke für die übergebene Markengruppen-ID. Der Markentyp wird anhand der Konstanten aus der TcLogMark.h (siehe Konstanten [▶ 45]) übergeben.

Optional kann die Event-ID benutzt werden, um zwischen verschiedenen Anwender-Events zu unterscheiden bzw. um zusätzliche Daten (als USHORT formatiert) in der Darstellung des TwinCAT 3 Realtime Monitors anzuzeigen.

7.2.1.4 LogMarkEx

```
virtual HRESULT LogMarkEx(TcMark16* pMark, ULONG CtrlId);
```

Beschreibung

Die Methode erstellt eine Marke. Die Markendefinition erfolgt anhand des Datentyps TcMark16 [▶ 43]. Der Markentyp wird anhand der Konstanten aus der TcLogMark.h (siehe Konstanten [▶ 45]) übergeben.

7.2.1.5 LogSequenceStart

```
virtual HRESULT LogSequenceStart(USHORT GroupId, USHORT EventId);
```

Beschreibung:

Die Methode erstellt eine Marke mit einem Sequence-Start für die übergebene Marken-ID.

7.2.1.6 LogSequenceStop

```
virtual HRESULT LogSequenceStop(USHORT GroupId, USHORT EventId);
```

Beschreibung:

Die Methode erstellt eine Marke mit einem Sequence-Stopp für die übergebene Marken-ID.

7.2.1.7 LogStart

```
virtual HRESULT LogStart(USHORT GroupId, USHORT EventId);
```

Beschreibung:

Die Methode erstellt eine Marke mit einem Sequence- und Intervall-Start für die übergebene Marken-ID.

Somit bildet diese Marke den Zeitpunkt eines Prozesses ab, in dem er sofort aktiv / gestartet ist.

7.2.1.8 LogStop

```
virtual HRESULT LogStop(USHORT GroupId, USHORT EventId);
```

Beschreibung

Die Methode erstellt eine Marke mit einem Sequence- und Intervall-Stopp für die übergebene Marken-ID.

Somit bildet diese Marke den Zeitpunkt eines Prozesses ab, in dem er direkt beendet wird.

7.2.1.9 SetContextId

```
virtual void SetContextId(USHORT nContextId);
```

Beschreibung:

Mit dieser Methode wird die verwendete Kontext-ID gesetzt.

7.2.1.10 InitLogMark

```
virtual HRESULT InitLogMark(ITComObjectServer* ipSrv);
```

Beschreibung:

Initialisiert die Instanz der CTcLog-Mark-Klasse.

Parameter:

ipSrv: Interface Pointer zum TcObjectServer.

7.2.1.11 ReleaseLogMark

```
virtual HRESULT ReleaseLogMark();
```

Beschreibung:

Gibt die Ressourcen der Instanz der CTcLogMark-Klasse wieder frei.

7.3 Konstanten

Diese Konstanten - definiert in der TcLogMark.h - definieren die möglichen Markentypen [▶ 13].

```
#define TCMARK_START 0xE0000000
#define TCMARK_STOP 0xC0000000
#define TCMARK_SEQ_START 0xA0000000
#define TCMARK_SEQ_STOP 0x80000000
#define TCMARK_IVAL_START 0x60000000
#define TCMARK_IVAL_STOP 0x40000000
#define TCMARK_REF_CALLER 0x08000000
```

8 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Downloadfinder

Unser Downloadfinder beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline:	+49 5246 963-157
E-Mail:	support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline:	+49 5246 963-460
E-Mail:	service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon:	+49 5246 963-0
E-Mail:	info@beckhoff.com
Internet:	www.beckhoff.com

Mehr Informationen:
www.beckhoff.de/te1010

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