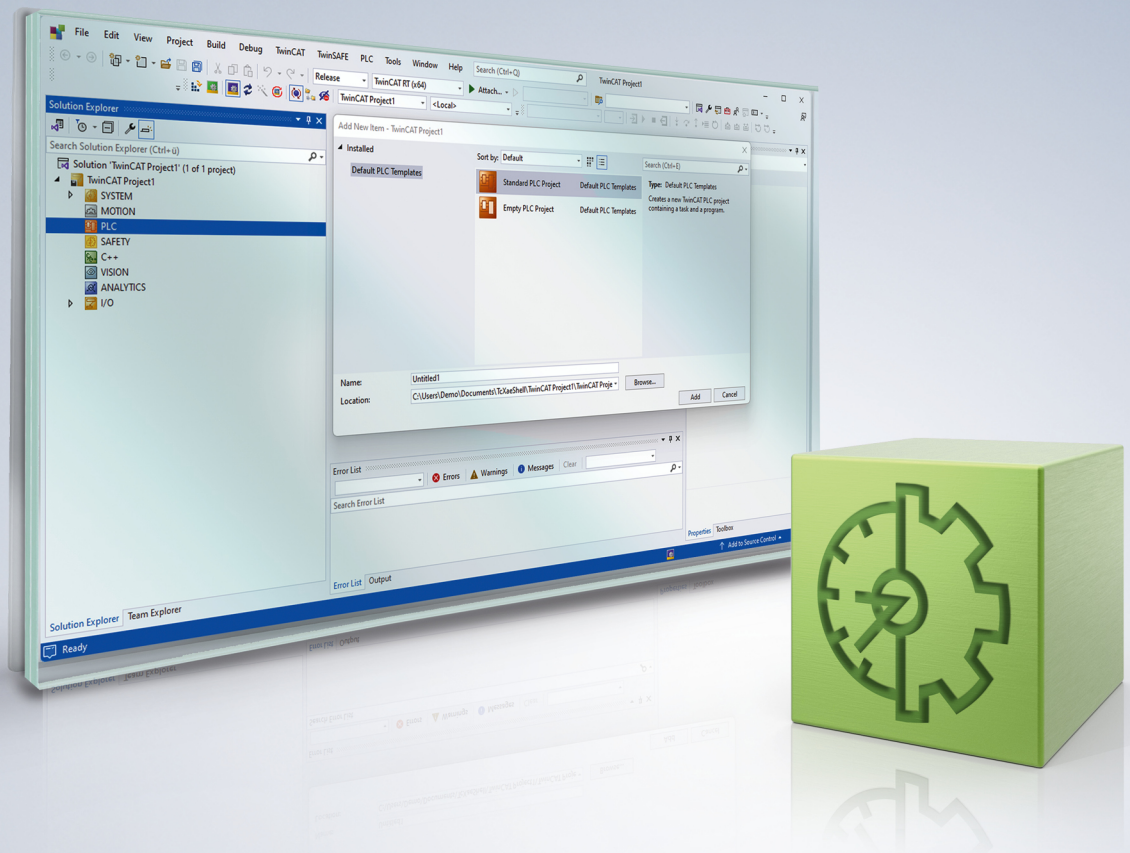
BECKHOFF New Automation Technology

Handbuch | DE

TE1010

TwinCAT 3 | Realtime Monitor



14.11.2025 | Version: 2.2.1

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, stets die aktuell gültige 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 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.

Marken

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® und XTS® 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 Kennzeichnungen führen.

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

Copyright

Fremdmarken

In dieser Dokumentation können Marken Dritter verwendet werden. Die zugehörigen Markenvermerke finden Sie unter: https://www.beckhoff.com/trademarks.

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.
Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

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.

Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

Warnungen vor Personenschäden

⚠ GEFAHR

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

⚠ WARNUNG

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

⚠ VORSICHT

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

Warnung vor Umwelt- oder Sachschäden

HINWEIS

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

Information zum Umgang mit dem Produkt

Diese Information beinhaltet z. B.:
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

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 Runtime. Er bietet eine grafische Darstellung der zeitlichen Abarbeitung von Echtzeittasks 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.

3 Installation

Systemvoraussetzungen

Technische Daten	Beschreibung
Betriebssystem	Windows 10
Zielplattform	Windows
Minimale TwinCAT-Version	TwinCAT 3.1.4024.0

TwinCAT Package Manager: Installation (TwinCAT 3.1 Build 4026)

Eine ausführliche Anleitung zur Installation von Produkten finden Sie im Kapitel Workloads installieren in der Installationsanleitung TwinCAT 3.1 Build 4026.

Installieren Sie den folgenden Workload, um das Produkt nutzen zu können:

- TE 1010 | TwinCAT 3 Realtime Monitor

TwinCAT Setup: Installation (TwinCAT 3.1 Build 2024)

Die Installation erfolgt über einen separaten Installer. Um TwinCAT 3 Realtime Monitor zu installieren, folgen Sie den Anweisungen des Installationsassistenten.

Lizenzierung

Der TwinCAT 3 Realtime Monitor (TE1010) ist ein Engineering Produkt. Die Lizenzierung erfolgt also ausschließlich auf dem Engineering System. Die Beschreibung der Lizenzierung einer Vollversion finden Sie in der Dokumentation „TwinCAT-3-Lizenzierung“.

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

4 Technische Einführung

Das folgende Kapitel beschreibt die technischen Grundlagen, die für die Verwendung des TwinCAT 3 Realtime Monitors hilfreich sind.

4.1 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 mit einer zu definierenden Zykluszeit und einer definierten Priorität vom Scheduler aufgerufen werden.

Die TwinCAT 3 Echtzeit ist eine Echtzeiterweiterung, welche in der aktuellen TwinCAT 3.1 Version unter den Microsoft Windows Betriebssystemen ab Windows 7 sowie unter TwinCAT/BSD und Beckhoff RT Linux® verwendet werden kann. Um den oben 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
- Multicore-Support
- Direkter Hardwarezugriff

Die TwinCAT-3-Multicore-Unterstützung erlaubt es die verfügbaren Rechenkerne entweder exklusiv für TwinCAT oder mit dem entsprechenden Betriebssystem geteilt zu verwenden. Im Folgenden werden die Kerne daher als "isolated" oder "shared" bezeichnet.

Echtzeitfähiges Scheduling

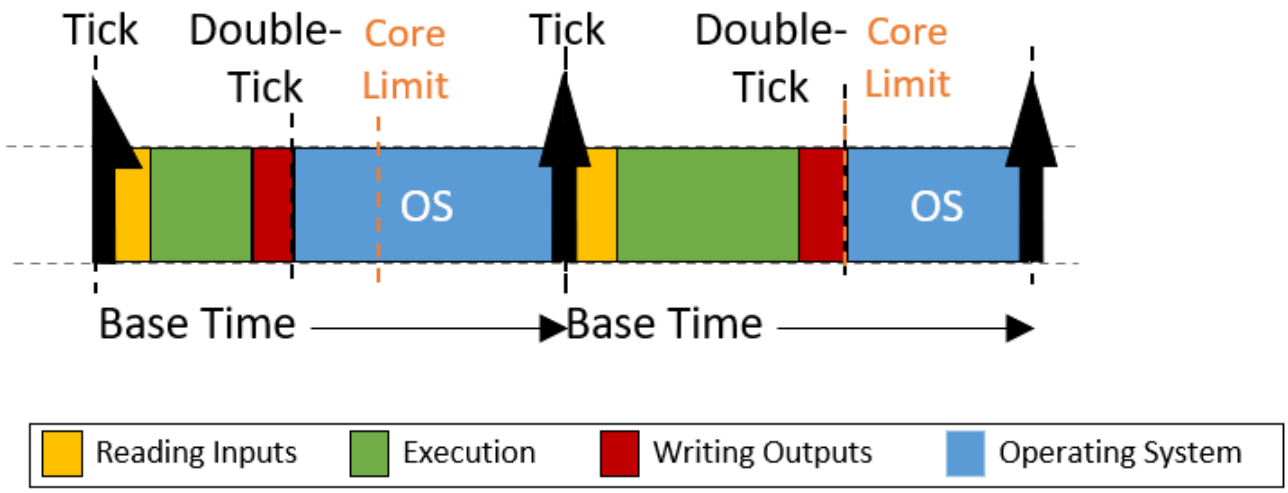
Die TwinCAT 3 Echtzeit arbeitet mit dem Doppeltick-Verfahren. Das bedeutet, dass das Umschalten in den Echtzeitmodus und wieder zurück jeweils von einem Interrupt ausgelöst wird. Der Interrupt beim Umschalten in den Echtzeitmodus startet gleichzeitig auch das Scheduling. Nach einer einstellbaren Zeitdauer, 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.

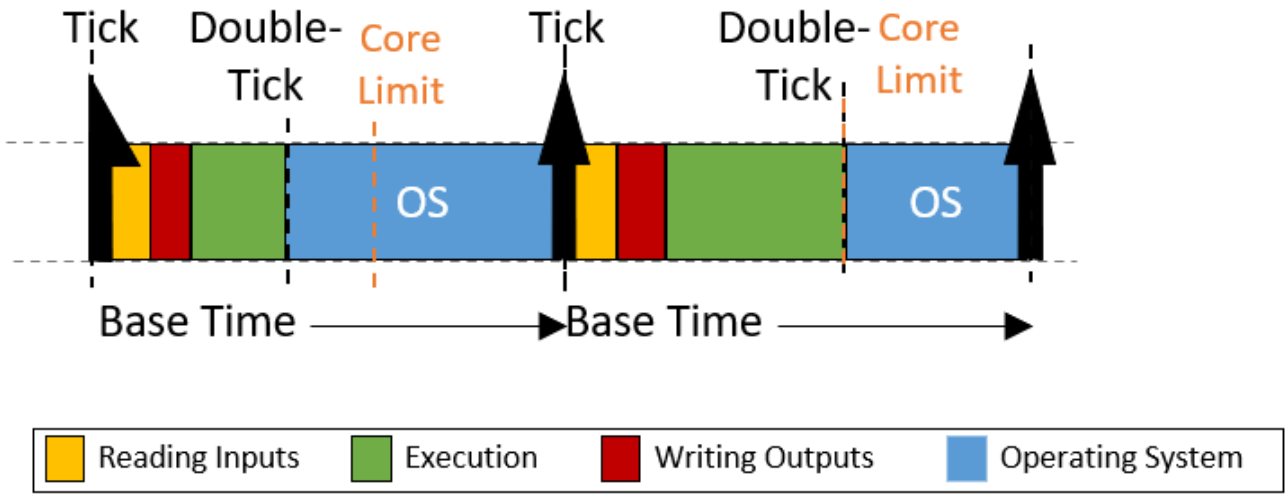
Ein Scheduling, bei dem die Priorität eines Tasks anhand seiner Zykluszeit abgeleitet wird, bezeichnet man auch als Ratenmonotones Scheduling. Die TwinCAT 3 Echtzeit aktiviert automatisch die Option „Automatic Priority Management“. Da dies nicht für jeden Anwendungsfall die beste Lösung ist, können Sie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

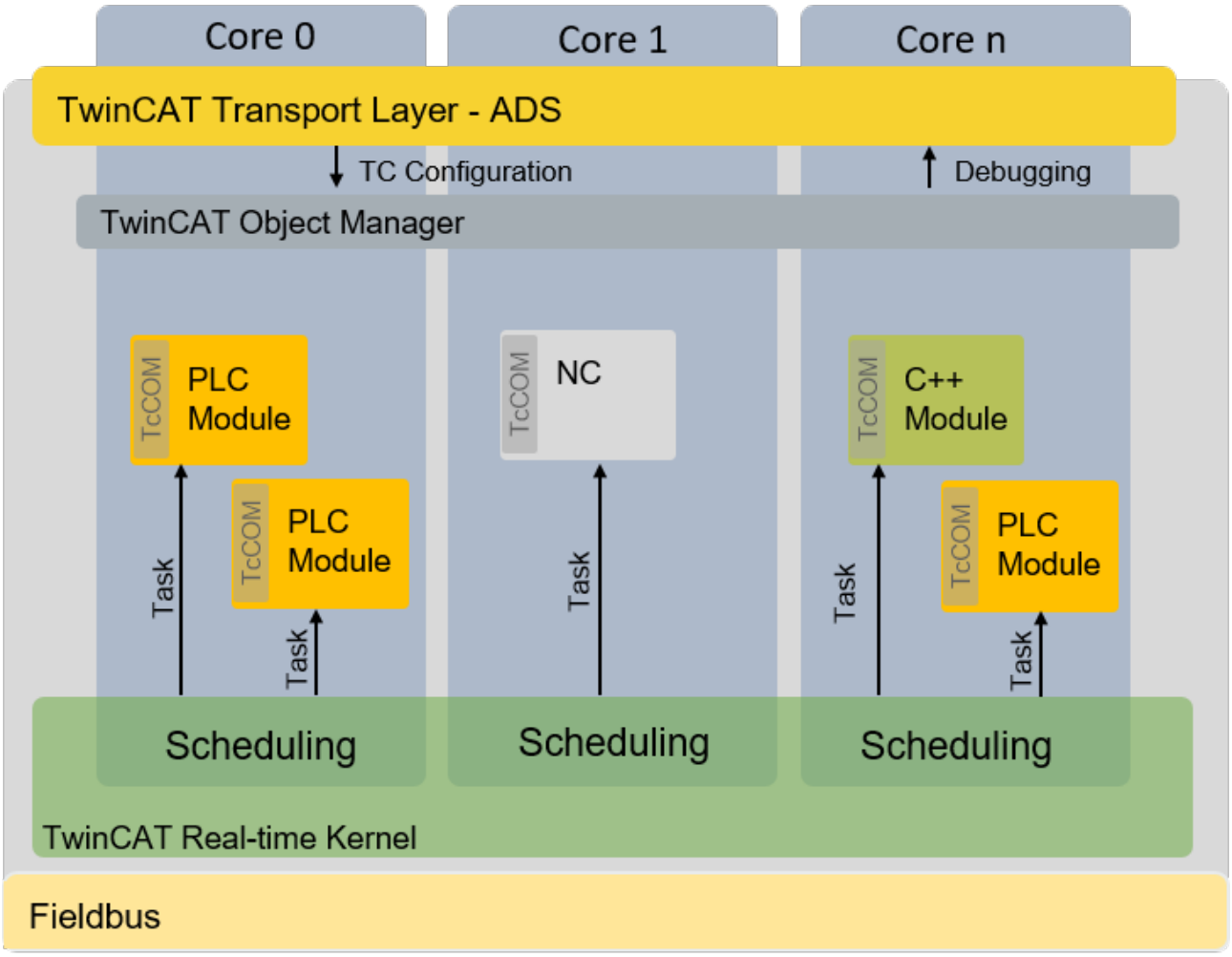
Präemptives Multitasking bedeutet, dass der aktuelle Zustand eines Prozesses (die CPU- und Floatingpoint-Register), bei einer Unterbrechung durch einen Interrupt (z. B. durch höher priore Prozesse), gesichert und der aktuelle Prozess „schlafen gelegt“ wird. Ist dies passiert, bestimmt der Scheduler, 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 Kernel-Mode von Windows bzw. TwinCAT/BSD 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. Unter Beckhoff RT Linux® arbeitet die Echtzeit mit der Echtzeiterweiterung im Usermode. Der direkte Hardware-Zugriff wird über spezielle Netzwerktreiber ermöglicht.

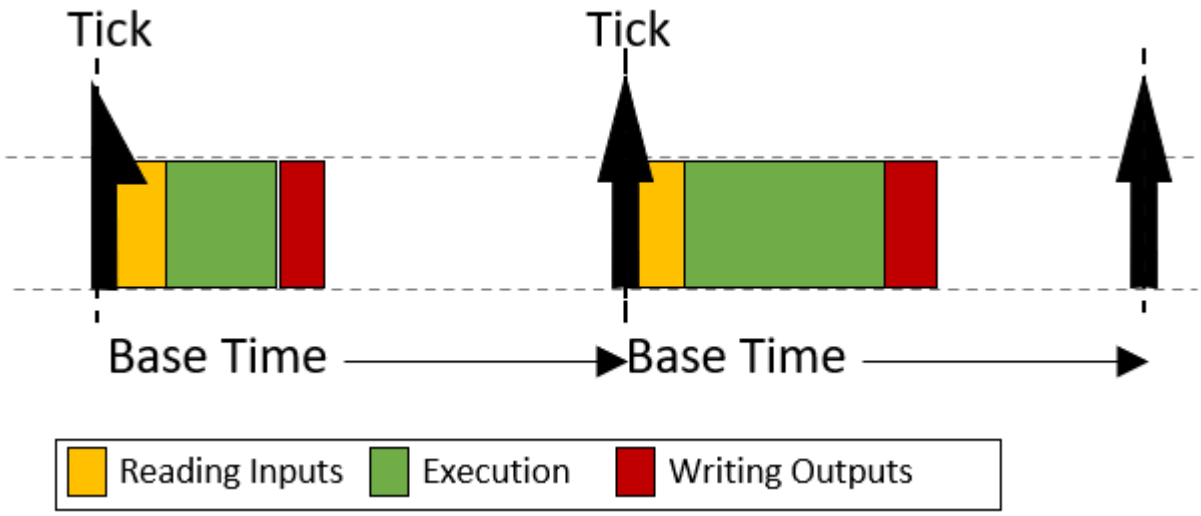
Schematische Darstellung der TwinCAT 3-Laufzeitumgebung

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.



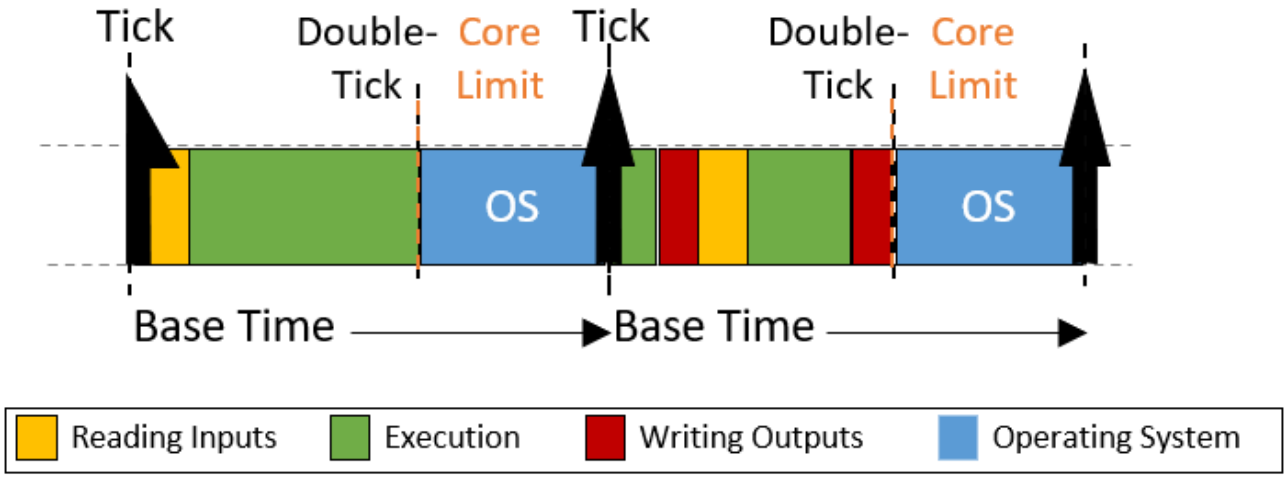
Isolierte Kerne

Wie unter Echtzeit Scheduling [▶ 10] 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 Präemptives Multitasking [▶ 11] 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 des „alten“ Prozesszustands.

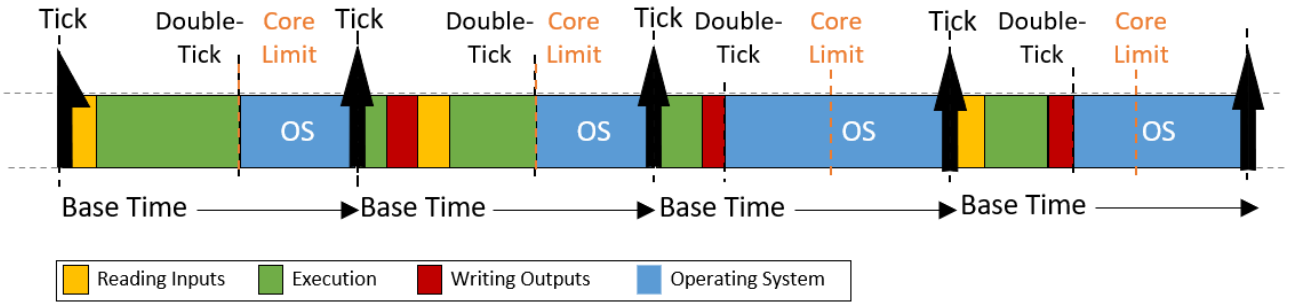


Verhalten bei Zykluszeitüberschreitung

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.

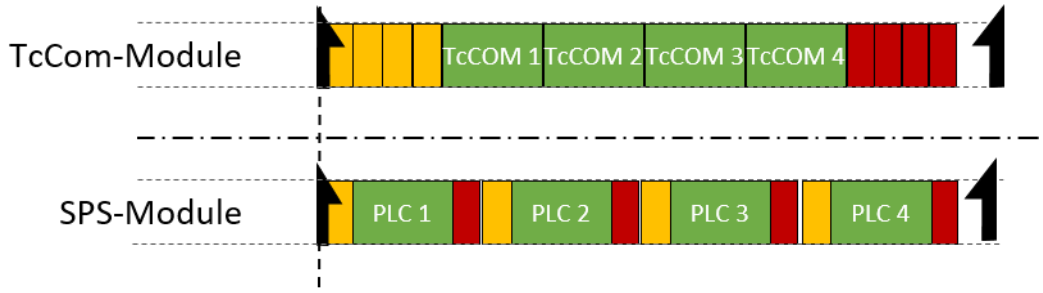


Unterschiede in der Abarbeitung zwischen SPS- zu „TcCom“-Laufzeitmodulen

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.

4.1.1 Tasks

Eine Task ist ein Laufzeitobjekt, welches von einem Scheduler eingeplant und angestoßen werden kann. Bei diesem Objekt melden sich Funktionen bzw. Laufzeitobjekte an, die im Kontext dieses Objekts ausgeführt werden sollen. Den Tasks werden eine Zykluszeit und eine Priorität zugeordnet, anhand derer der Scheduler die Ausführung der Tasks einplant (siehe Echtzeit Scheduling [▶ 10]). Des Weiteren werden jeder Task ein gemeinsamer Speicher/ Adressraum, auf den alle Tasks gemeinsam zugreifen können, und ein zusätzlicher Stackspeicher zugeordnet. Dieser Stackspeicher wird benötigt, um das verschachtelte Ausführen von Funktionen und Unterfunktionen zu erlauben. Der Stack dient auch als Speicher für lokale Variablen. Der Zustand einer Task wird durch diesen Stack und den aktuellen Inhalt der Maschinenregister (Rechenregister) definiert. Bei einem Kontextwechsel bzw. einem Unterbrechen einer Task durch eine höher Priore, wird dieser Zustand gesichert und beim erneuten bzw. weiteren Ausführen der Task wieder hergestellt. Die Größe des Task-Stacks kann in TwinCAT definiert werden (siehe Kapitel Registerkarte Settings der Echtzeit-Einstellungen).

Zur Ablaufsteuerung von mehreren Tasks bietet ein (Echtzeit-)System Funktionen zur Kommunikation und Synchronisation an (siehe Kapitel Multitask-Datenzugriffs-Synchronisation in der SPS).

In den folgenden Kapiteln wird auf die entsprechenden Task-Arten eingegangen.

4.1.1.1 TwinCAT Task

TwinCAT-Standard-Tasks, an welchen sich TwinCAT-Laufzeit-Module anmelden können.

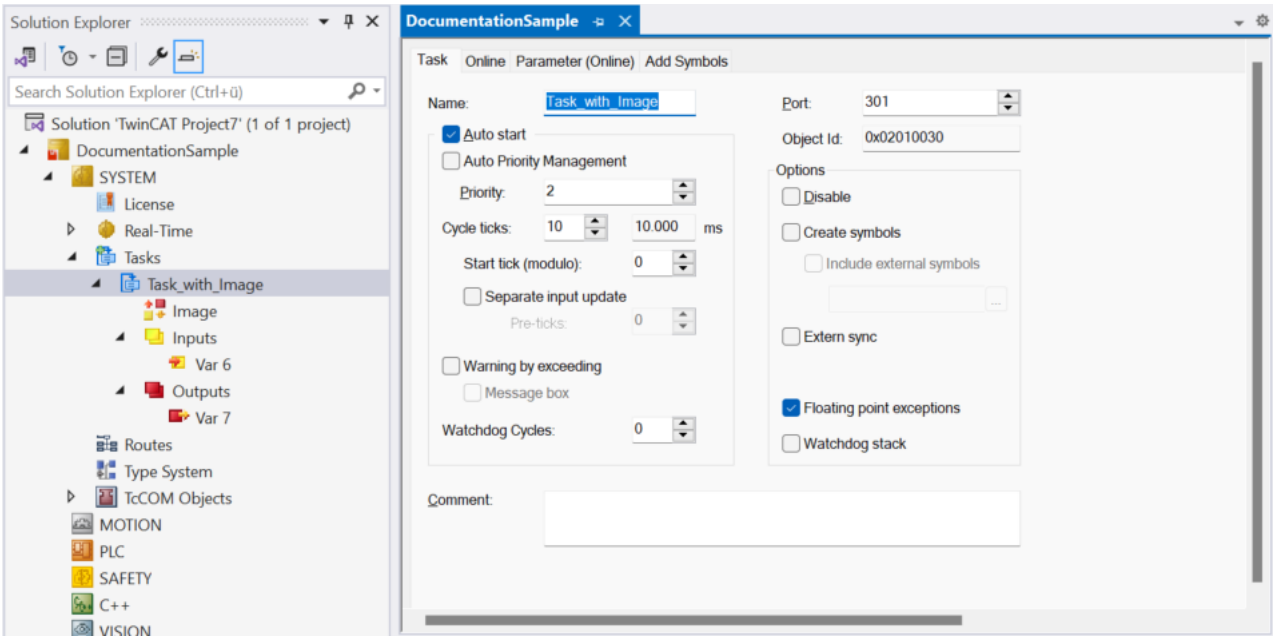
Für die Verwendung der Tasks in einer SPS geschieht dies über die Zuordnung einer Task zu einer Task-Referenz (siehe Taskreferenz erzeugen).

Für TcCom-Module im Allgemeinen erfolgt die Zuordnung des TcCom-Modules zu Tasks über den Reiter **Context** des TcCom-Moduls. (Siehe Regisiterkarte Context im Kapitel TcCom-Module-Handhabung).

Die Einstellungen einer Task sind im Unterkapitel TwinCAT Task in der Dokumentation „Das TwinCAT-Projekt“ beschrieben.

4.1.1.2 TwinCAT Task with Image

Gegenüber einer Standard-Task (siehe Kapitel TwinCAT Task [▶ 15]) hat die **TwinCAT Task with Image** zusätzlich ein eigenes Prozessabbild.

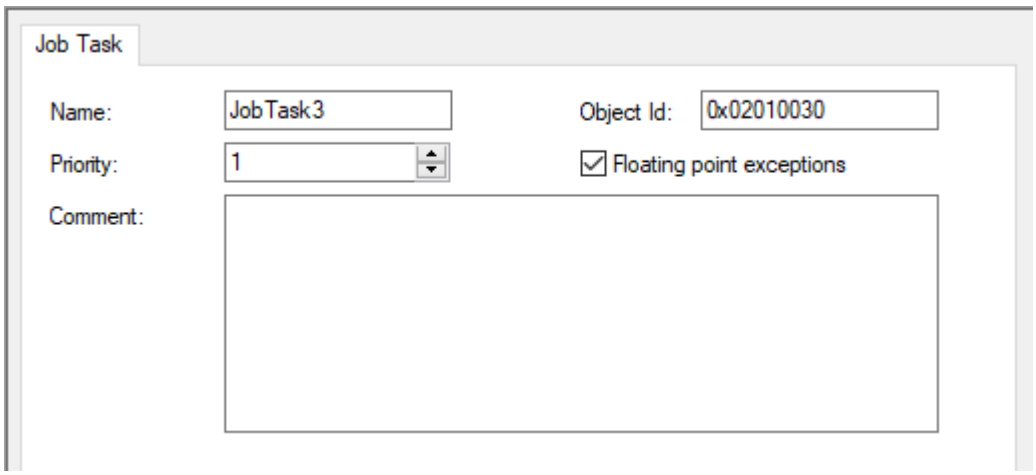


In diesem Prozessabbild können Variablen angelegt werden, die mit anderen Prozessabbildern (z. B. von EtherCAT-Teilnehmern) verknüpft werden. Je nach eingestellter Zykluszeit und Priorität stößt die Task entsprechend das Mapping an. Es ist somit z. B. möglich eine zyklische Bus-Kommunikation zu betreiben, ohne dass eine SPS oder ein anderes Laufzeitmodul benötigt werden. Auf die Variablen des Prozessabbilds kann z. B. über ADS aus dem TwinCAT Scope oder einer Nicht-Echtzeit-Applikation zugegriffen werden.

4.1.1.3 TwinCAT Job Task (Worker Task)

Eine Job Task ist eine Task, die bei Bedarf ausgeführt wird. Sie wird aus einer Applikation heraus aufgerufen und nicht zyklisch ausgeführt. Eine Job Task kann sowohl direkt unter Tasks oder in einem Job Pool erstellt werden. Legen Sie die Job Task direkt unter Tasks an, können dieser direkt Aufgaben (Jobs) aus einer Kundenapplikation heraus übergeben werden.

Legen Sie die Job Task hingegen unter einem Job Pool an, werden die Aufgaben aus der Applikation heraus dem Job Pool übergeben. Dieser weist die Aufgabe dann der Job Task in seinem Pool zu, die als nächstes frei wird.



Name	Name der Job Task
Object Id	Objekt-ID der Job Task
Priority	Priorität der Job Task
Floating point exception	Bestimmt, ob TwinCAT auf Fließkomma-Exceptions prüft oder nicht.
Comment	Optionaler Kommentar zur Job Task

Wählen Sie die Einstellung **Floating point exception** bei der aufrufenden Task und bei der Job Task gleich.

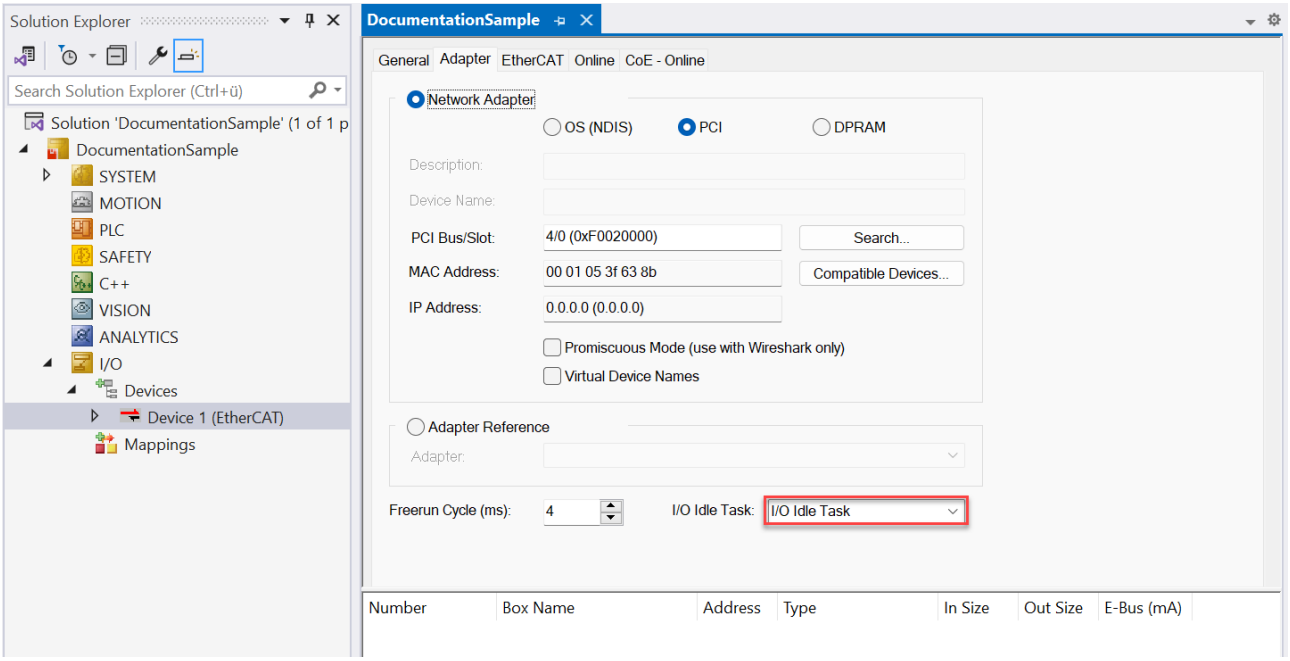
4.1.1.4 IO-Idle Task

Die I/O-Idle Task übernimmt für Feldbusse die azyklische Kommunikation, somit ist sie zuständig für:

- die State Machine der EtherCAT Devices
- CoE- und SoE-Kommunikation
- Parameter lesen bzw. setzen über AOE
- Mailbox-Kommunikation
- Azyklische Diagnose vom EtherCAT (z. B. Statusabfragen)

Die I/O-Idle Task ist eine zyklische Task, wird jedoch nicht für die zyklische I/O-Kommunikation (Prozessdatenaustausch) verwendet. Daher ist es in den meisten Fällen sinnvoll, die Priorität so zu wählen, dass diese nach den SPS- und Motion-Tasks ausgeführt wird. Da es sich zudem um eine azyklische Kommunikation handelt, sind Zykluszeitüberschreitungen dieser Tasks im Allgemeinen nicht von Bedeutung.

Seit der Version TwinCAT 3.1.4026 ist es möglich, mehrere I/O-Idle Tasks zu verwenden (eine pro EtherCAT-Master). Die Auswahl der I/O-Idle Task erfolgt über die Adaptereinstellungen des EtherCAT-Masters.



4.1.1.5 PLC-AuxTask

Die PLC-AuxTask dient der Kommunikation zwischen den SPS-Editoren und den SPS-Laufzeitmodulen. Dies beinhaltet den Download und den Online-Change von SPS-Steuerungscode ebenso wie das Debugging (Monitoring von Werten, Setzen von Breakpoints, etc.). Darüber hinaus verarbeitet die PLC-AuxTask auch die ADS-Nachrichten, die unabhängig von der Entwicklungsumgebung (TcXaeShell) an das Laufzeitsystem gesendet werden (z. B. von einer HMI).

Die PLC-AuxTask ist keine zyklische Task und Sie sollten ihre Priorität kleiner als diese wählen. Somit wird gewährleistet, dass die zyklischen Tasks die PLC-AuxTask unterbrechen können. Im Falle eines Online-Changes werden der neue Code über die PLC-AuxTask auf das Zielsystem gespielt und die Symbole entsprechend generiert, etc.. Lediglich die „kritische Phase“ eines Online-Changes, in der der auszuführende Code getauscht wird, wird so geschützt, dass dieser Vorgang nicht durch die zyklischen Tasks unterbrochen werden kann.

Verwenden mehrere SPS-Laufzeitmodule dieselben SPS-Tasks, können diese sich damit durch einen Online-Change beeinflussen.

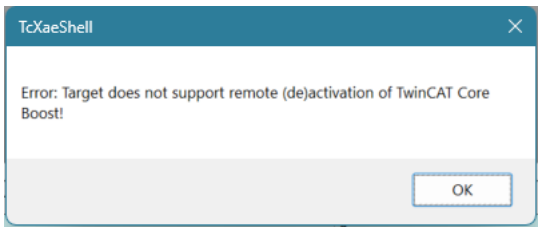
4.1.2 Core Boost

TwinCAT Core Boost

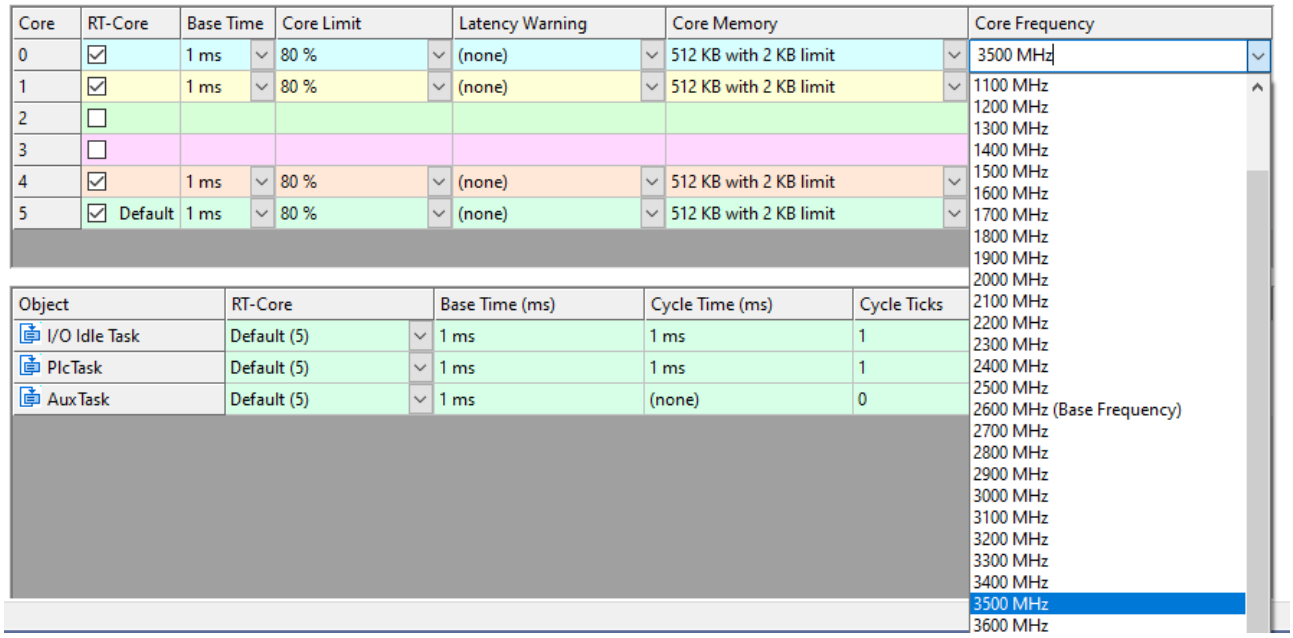
Voraussetzung: Sowohl die Engineering-Umgebung als auch die Laufzeitumgebung müssen mindestens eine TwinCAT-Version 3.1.4026.6 verwenden.

Wird das TwinCAT Feature Core Boost für ein Zielsystem unterstützt und ist aktiv, wird dies nach dem Betätigen des Buttons **Read from Target** in der Auswahlbox **TwinCAT Core Boost** angezeigt. Zum Aktivieren oder Deaktivieren der Funktion muss diese Einstellungen durch den Button **Set on Target** auf dem Zielsystem aktiviert werden. Nach dem Ändern dieser Einstellung muss ein Neustart des Rechners erfolgen.

Wird die TwinCAT Core Boost Funktion von einem Zielsystem nicht unterstützt, erscheint nach dem Betätigen des Buttons **Set on Target** die folgende Mitteilung:



Ist die Einstellung TwinCAT Core Boost gesetzt, ist es möglich für jeden Echtzeitkern eine Taktfrequenz zu definieren. Wird für einen Echtzeitkern keine Taktfrequenz definiert, wird automatisch die **Base Frequency** ausgewählt.



Richtige Auswahl der Core Frequency

TwinCAT überwacht automatisch die Taktfrequenzen der einzelnen Kerne anhand der im System hinterlegten Grenzwerte für die Temperatur der einzelnen Kerne bzw. des Stromverbrauchs der einzelnen Packages. Werden diese Grenzen überschritten, regelt TwinCAT die Taktfrequenzen der einzelnen Kerne entsprechend runter (siehe auch Kapitel Registerkarte Core Boost). Wird TwinCAT gezwungen die Taktfrequenzen von einzelnen Echtzeitkernen herunterzuregeln, kann dies u. U. Einfluss auf das in TwinCAT eingestellte Echtzeitverhalten haben. Die auf diesem Echtzeitkern ausgeführten Tasks haben dann längere Ausführungszeiten, was u. U. zu Zykluszeitüberschreitungen führen kann. Sie sind daher mitverantwortlich die Taktfrequenzen der Echtzeitkerne so zu wählen, dass TwinCAT nicht dauerhaft im Throttling (heruntergeregelt) betrieben wird. Ein dauerhaftes Überschreiten der Grenztemperatur kann dazu führen, dass sich das System abschaltet.

Auswahl der Taktfrequenz für Nichtechtzeitkerne

Nichtechtzeitkerne sind Rechnerkerne, auf denen TwinCAT nicht aktiviert ist und die nicht in der Echtzeit verwendet werden, so dass dort nur vom Betriebssystem angestoßene Prozesse ausgeführt werden. Für die Nichtechtzeitkerne wird die Taktfrequenz vom Betriebssystem je nach Bedarf automatisch ausgewählt. Die maximale Taktfrequenz bis zu der Nichtechtzeitkerne hochtakten können, ist die Basisfrequenz. Ab den Intel® Prozessoren der 12./ 13. Generation ist es für Nichtechtzeitkerne möglich, dass diese bei Bedarf die Taktfrequenz bis zur Core-Boost-Frequenz übertakten. Die Höhe der Core-Boost-Frequenz ist im System hinterlegt und unterscheidet sich je nach Prozessortyp.

Konfigurierbare Taktfrequenzen:

Prozessor	Prozessorgeneration	Basistakt	Konfigurierbarer Core Boost Takt
Core i3-11100HE	Intel® Core™ der 11. Generation	2,40 GHz	4,00 GHz
Core i5-11500HE	Intel® Core™ der 11. Generation	2,60 GHz	4,10 GHz
Core i7-11850HE	Intel® Core™ der 11. Generation	2,60 GHz	4,20 GHz
Core i3-13100E	Intel® Core™ der 13. Generation	3,30 GHz	4,20 GHz
Core i5-13400E	Intel® Core™ der 13. Generation	2,40 GHz	4,10 GHz
Core i7-13700E	Intel® Core™ der 13. Generation	1,90 GHz	4,00 GHz
Core i9-13900E	Intel® Core™ der 13. Generation	1,80 GHz	3,90 GHz

4.1.3 Core Memory

TwinCAT (3.1.4026 und 3.1.4024) ermöglicht Echtzeit-Tasks dynamisch Speicher innerhalb des Echtzeitkontextes zu allokieren. Da Echtzeit-Tasks keine Speicherblöcke direkt vom Betriebssystem allokieren können, bietet TwinCAT verschiedene Speicherpools an. Diese Pools sind vorab vom Betriebssystem allokierte Speicherblöcke. Diese Speicherblöcke werden an den physischen Speicher angepinnt, um ein Paging beim Zugriff innerhalb des Echtzeitkontexts zu vermeiden.

Wie im Kapitel Registerkarte Settings beschrieben, gibt es für TwinCAT mehrere Speicher-Pools. Diese sind:

Executable RT Memory	Ausführbarer Echtzeitspeicher, dieser wird vom TwinCAT System benötigt und steht dem Anwender nicht direkt zur Verfügung
Global RT Memory	Globaler Echtzeitspeicher für Speicherallokationen von TwinCAT-Modulen (SPS und C++) im Echtzeitkontext
Core Memory <n>	Echtzeitspeicher für Speicherallokationen des <n>-ten Echtzeitkernes
Global ADS Memory	Datenspeicher für die ADS-Kommunikation. Aus diesem Speicher werden Allokationen für ADS-Nachrichten zwischen den TwinCAT-Komponenten allokiert.
Tc/OS Memory	Speicher-Pool der Usermode Runtime: die Allokation erfolgt beim Aufstarten einer Usermode Runtime. Von diesem Speicherpool werden die anderen Speicherpools bei Verwendung einer Usermode Runtime bedient.

Die Größe des Globalen RT Memory ist über die Option **Router Memory** in der TwinCAT-3-Engineering-Umgebung für jedes Projekt einstellbar. Seit der TwinCAT-Version 3.1.4026 ist der Datenspeicher für ADS-Nachrichten abgetrennt vom „Router Memory“ und als eigenständiger globaler ADS-Memory vorhanden. Die Größe des globalen ADS-Memory wird ermittelt aus der Größe des Globalen RT-Speichers und entspricht 25 % seiner Größe, maximal aber 32 MB.

Mit der TwinCAT Version 3.1.4026 wurde zudem ein weiterer Speicherpool eingeführt, der Core Memory. Der Core Memory bietet für jeden einzelnen Echtzeitkern einen dedizierten Pool für die Echtzeit. Ist der Core Memory für einen Echtzeitkern konfiguriert, wird zuerst versucht, aus diesem Speicherpool den angeforderten Speicher zur Verfügung zu stellen. Ist dieser nicht ausreichend, wird der Speicher aus dem globalen Echtzeitspeicher verwendet.

Der Core Memory erlaubt damit parallele Speicher-Allokationen von verschiedenen Kernen parallel und unabhängig voneinander zu verarbeiten. Für den Zugriff auf den globalen Echtzeitspeicher müssen parallele Speicherallokationen von verschiedenen Kernen sequenziell bearbeitet werden. Das kann bei vielen Speicherallokationen innerhalb eines Echtzeitzyklus zu Wartezeiten führen und ist damit ein Performance-Engpass, z. B. bei Vision-Anwendungen.

Siehe auch:

- Core Management

4.2 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 Nutzerprozessen/ -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 benutzerdefiniertes Datum mitgegeben werden, um ggf. zusätzliche Informationen 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 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, ab dem 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, ab dem 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, ab dem 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 Scheduling innerhalb eines Echtzeitkerns ab. Für diese Echtzeitkontexte gilt, dass zu jeder Zeit immer nur genau eine der einem Kontext zugeordneten Tasks aktiv ist. Für nutzerspezifische Markengruppen gilt diese Einschränkung nicht.

Bei der Verwendung der erweiterten Marken (durch Verwendung des FB_RTMon_LogMark [▶ 57]), 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 (hexadezimal 0x353) zugeordnet.

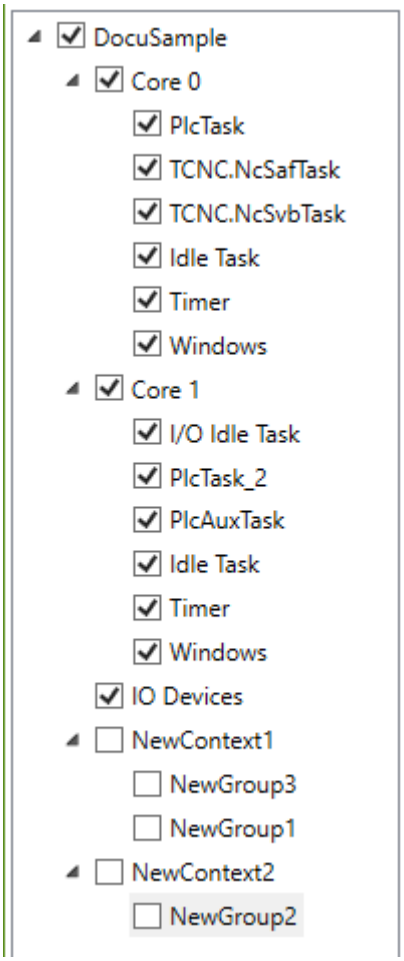
Bei der Verwendung der generischen Marken (durch Verwendung des Bausteins FB_RTMon_LogMarkBase [▶ 62]) können eigene Kontexte (also Zusammenhänge) definiert werden. Dies könnte beispielhaft eine Gruppierung nach Prozessart oder nach Maschinenmodulen (Funktionseinheiten) sein.

Darstellung in der Baumansicht:

Wie bereits beschrieben, verwenden alle Marken, die dieselbe Task oder 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 nutzerbezogene 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 [▶ 53]).



Wie unter Kontext [▶ 20] 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 Markengruppe (also alle Marken einer Task eines Prozesses) innerhalb einer Zeile dargestellt. Näheres hierzu unter Darstellung im Chart.

Darstellung im Chart:

Symbole in der chart-Darstellung:

	Sequenz-Start
	Sequenz Stopp
	Zeigt einen Intervall-Start oder einen 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. Genauere Informationen erfahren Sie per Tooltip auf der Marke. Die Marke selbst hat keinen Einfluss auf das zeitliche Verhalten der Task oder 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.

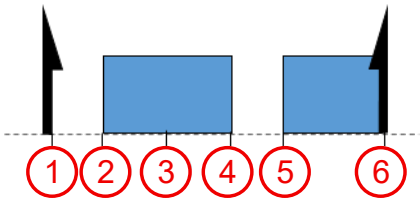
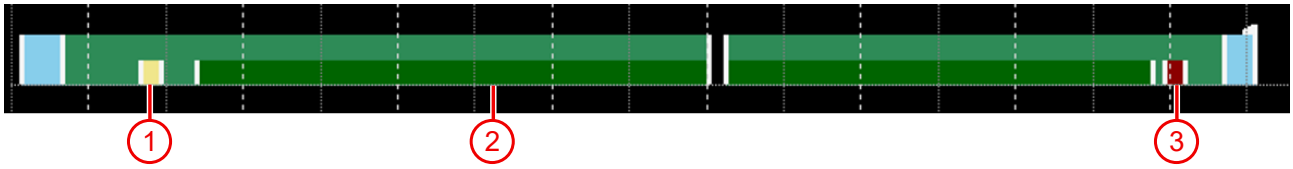


Abbildung der Abarbeitung eines SPS-Laufzeitmoduls:

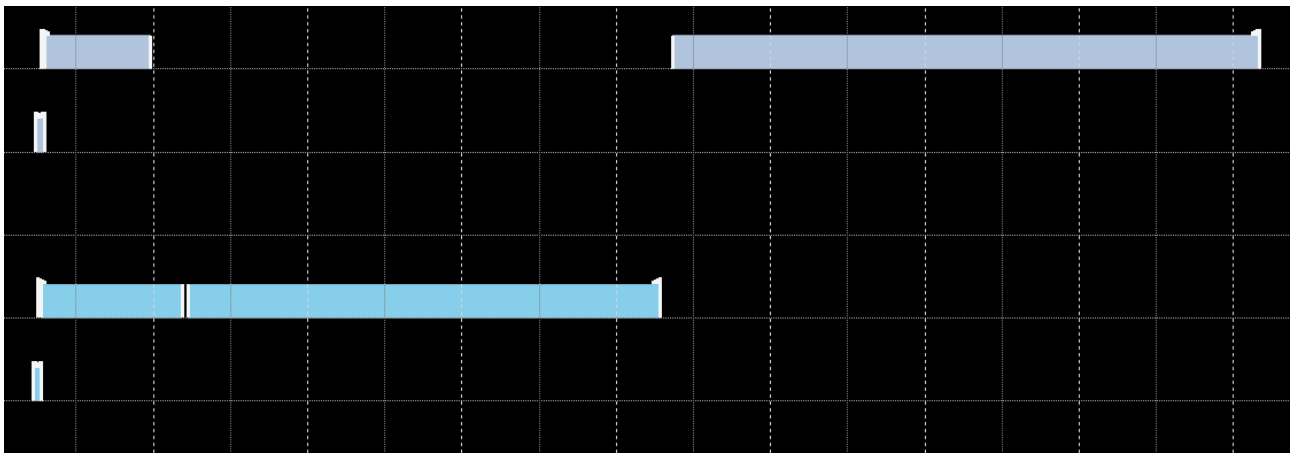
Wie im Absatz Echtzeit Scheduling [▶ 10] beschrieben, ruft jedes SPS-Laufzeitmodul das Update der Ein- und Ausgänge selbst auf. Die komplette Abarbeitung der SPS findet im Cyclic-Update der sie aufrufenden Task statt. Aus diesem Grund wird die Abarbeitung der SPS, sofern das detaillierte Logging aktiviert 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.



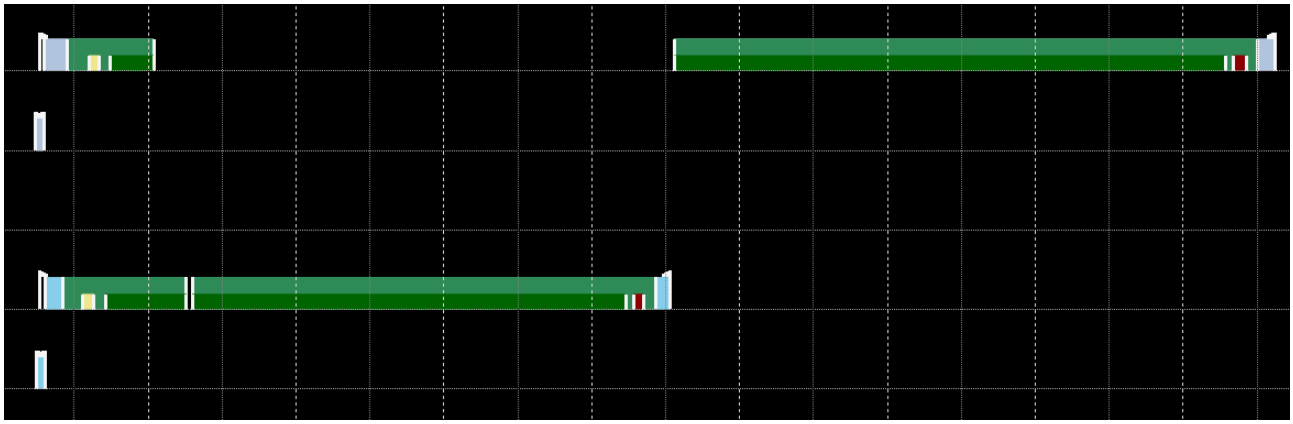
Detailliertes Logging:

Die Option „detailedLogging“ (siehe Projektknoten [▶ 52] bzw. Kontextknoten [▶ 53]) 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:



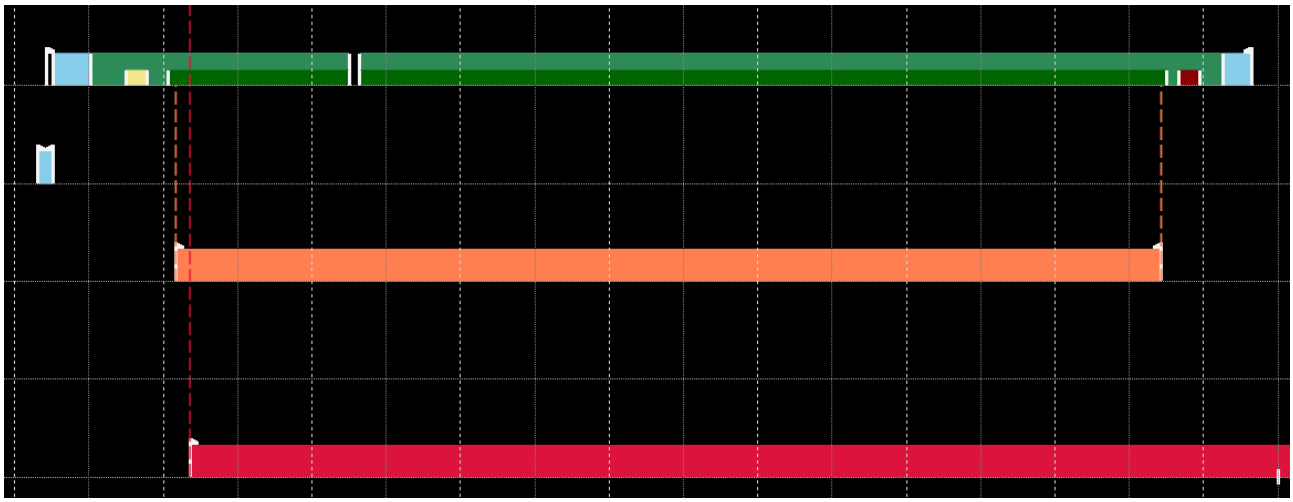
Detailed Logging aktivert:



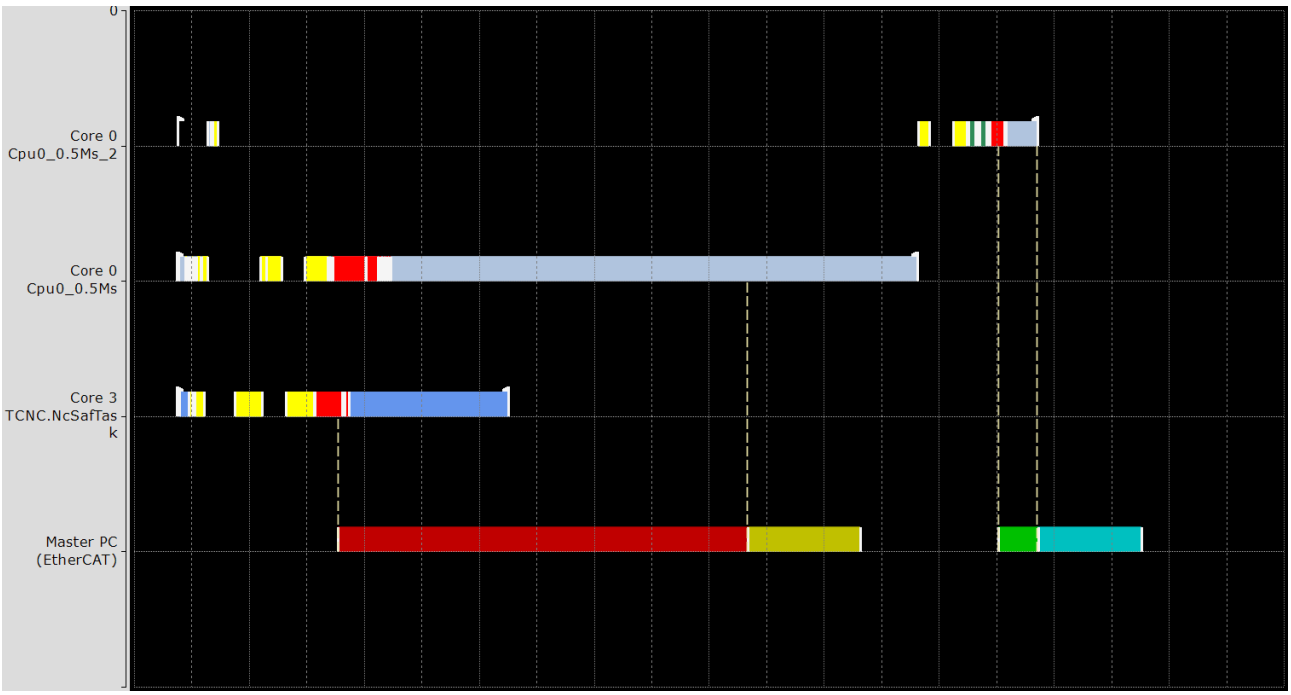
Darstellung der Task-Referenzen:

Anhand von Task-Referenzen kann die Zuordnung von Marken zu den einzelnen Echtzeittasks im TwinCAT 3 Realtime Monitor sichtbar gemacht werden. Dies ermöglicht sowohl der Zuordnung des Absendens der einzelnen EtherCAT Frames als auch die Zuordnung von Nutzer-Kontexten zu Echtzeittasks. Dargestellt werden Task-Referenzen durch gestrichelte Linien.

Die Option **Show Task Reference** (siehe Markengruppen-Element [▶ 54]) schaltet diese Option für Nutzer-Kontexte (z.B. Anwenderprozesse) an. In der folgenden Abbildung sehen Sie die Zuordnung des orange dargestellten Anwenderprozesses zu einer SPS-Task.



Für EtherCAT Frames ist diese Option automatisch aktiviert. Die folgende Abbildung zeigt dies an einem Beispiel. Die farbliche Darstellung der Frames ist dabei dieselbe wie in TwinCAT.



Reihenfolge der Kontexte im Anzeigefenster

Je nach TwinCAT-Projekt, das auf dem zu untersuchenden Zielsystem aktiviert wurde, können sehr viele Kontexte, verteilt auf verschiedenen Kernen, dargestellt werden. Dies kann dazu führen, dass einzelne Zusammenhänge nur schwer sichtbar gemacht werden können. Aus diesem Grund ist es möglich, die Kontexte im Darstellungsfenster beliebig anzuordnen.

1. Deaktivieren Sie die Option **KeepGraphOrder** in den Projekteinstellungen (siehe Projektknoten [▶ 52]).
2. Wählen Sie alle Kontexte im Baum des Projekt-Eintrags ab.
3. Selektieren Sie die Kontexte in der Reihenfolge, in der Sie diese im Darstellungsfenster anzeigen möchten.

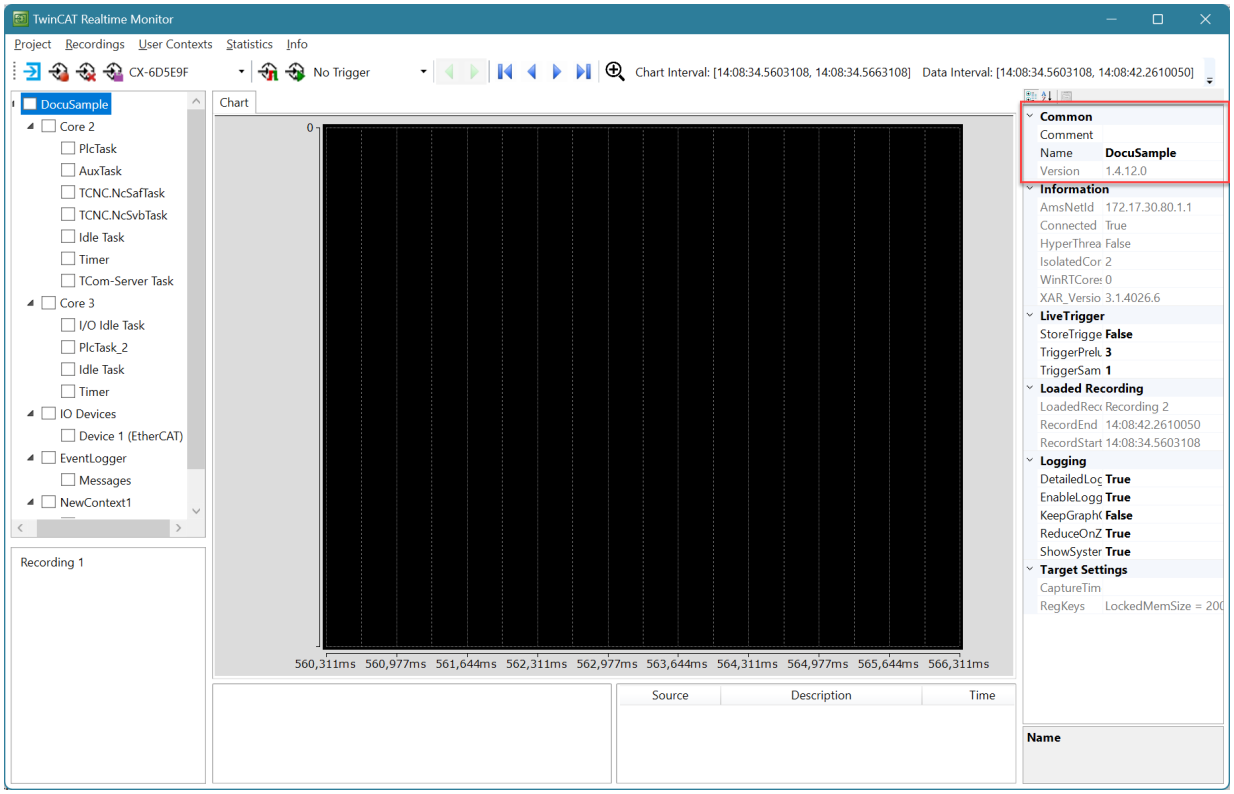
Beispiel: In der vorangegangenen Abbildung wurden alle Kontexte ausgeblendet und in der Reihenfolge wieder angewählt, sodass ohne Überscheidungen leicht ersichtlich ist, welche Task welchen EtherCAT Frame abschickt.

5 Quickstart

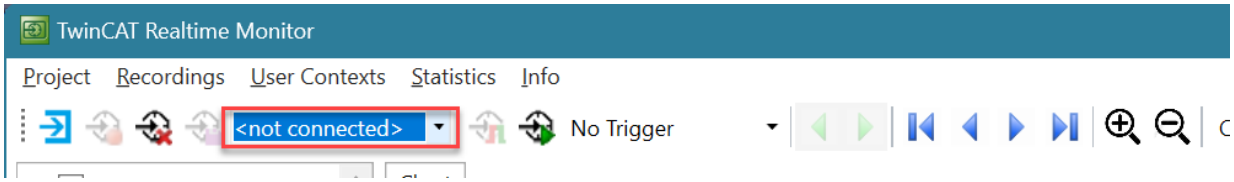
Das folgende Kapitel soll einen leichten Einstieg in die Verwendung des TwinCAT 3 Realtime Monitors ermöglichen.

✓ 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, indem Sie die Option **New Project** im Menü **Project** des TwinCAT 3 Realtime Monitors anklicken. Das Ändern des Projektnamens kann über die Projekteigenschaften erfolgen.



3. Wählen Sie über die Toolbar des Realtime Monitors das Zielsystem aus, welches Sie analysieren wollen.

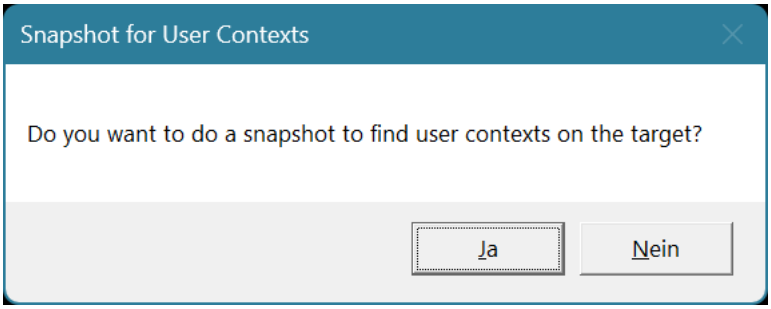


⇨ Es erscheint eine Abfrage, ob Sie die Konfiguration vom Zielsystem laden wollen.

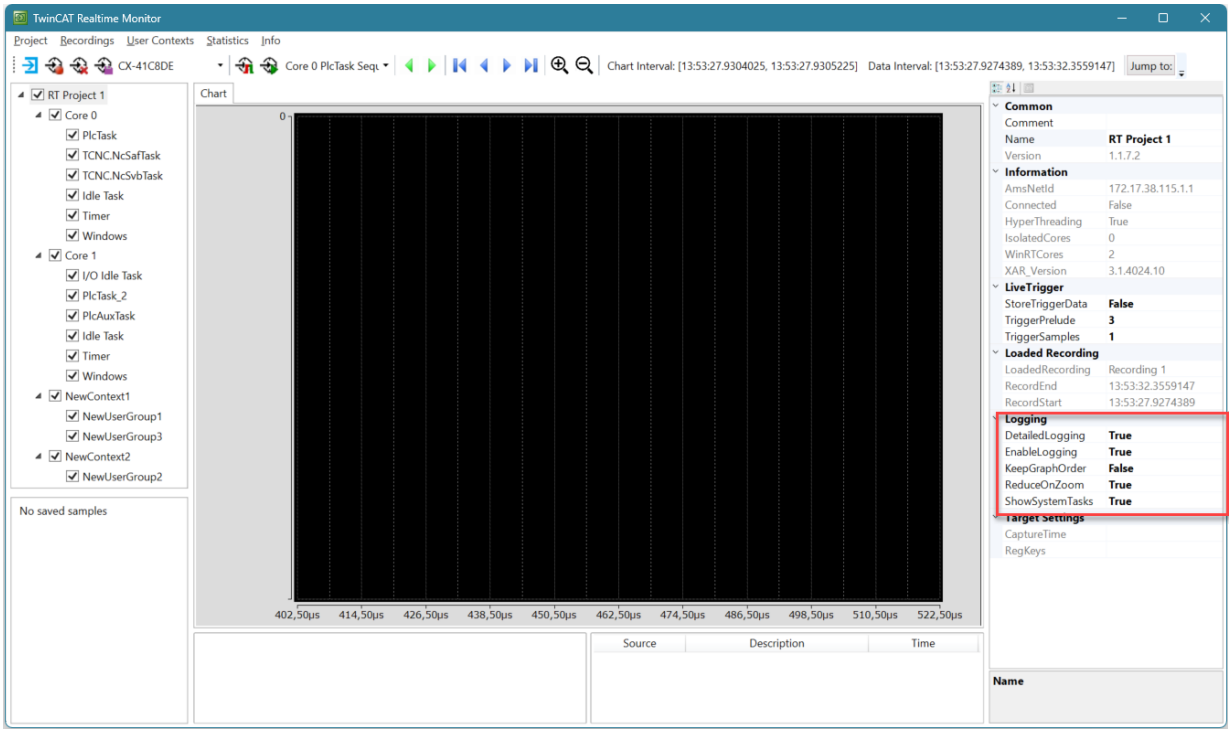
4. Bestätigen Sie mit **Ja**.

⇨ Die aktive Konfiguration des Zielsystems wurde geladen und die Kontexte werden hierarchisch als Baum dargestellt.

⇨ Es erscheint eine Dialogbox mit der Frage, ob der Realtime Monitor ein kurzes Snapshot (Kurzzeitaufnahme) vom Zielsystem durchführen soll, um Nutzerkontexte automatisch zu erkennen und anzulegen.



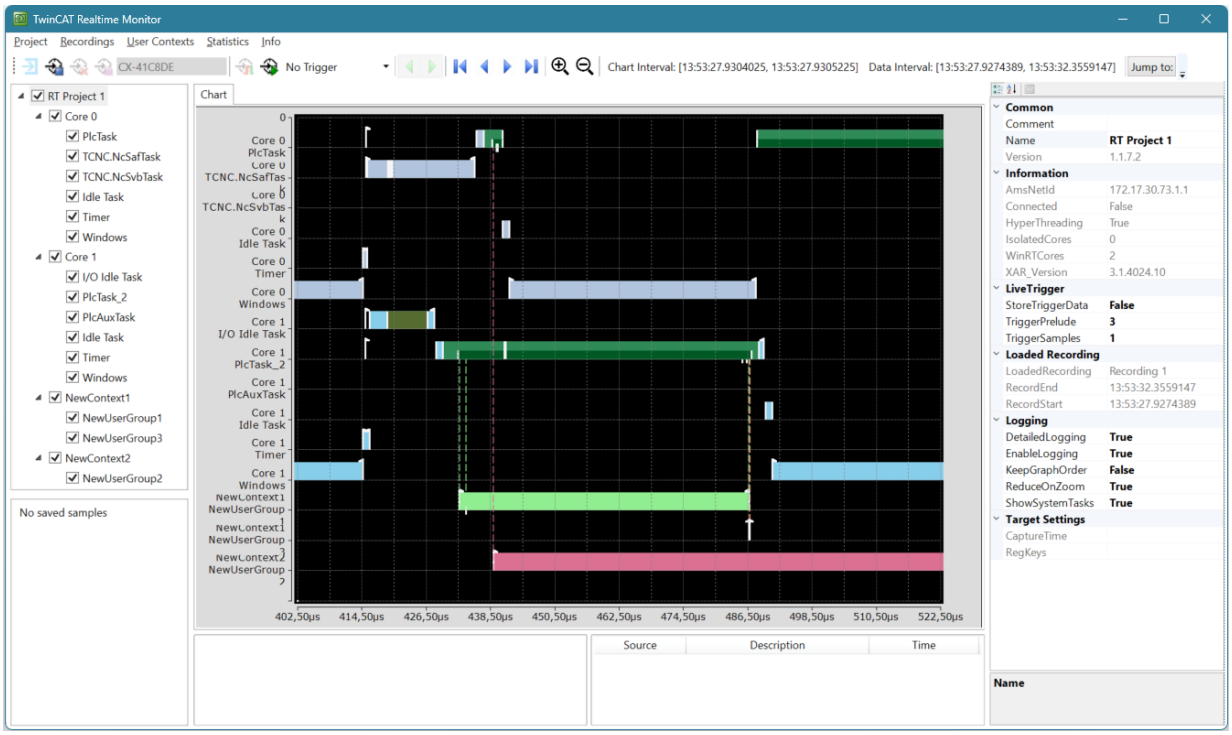
5. Bestätigen Sie mit **Ja**.
6. Wählen Sie im Baum die Kontexte/ Tasks aus, die Sie im Realtime Monitor dargestellt haben möchten, sodass diese mit einem Haken markiert sind.



7. Markieren Sie in der Baumansicht das Projekt und setzen Sie im Eigenschaftenfenster die Option **Detailed Logging** auf „True“. (Siehe rote Markierung im Bild des vorherigen Schrittes bzw. der Beschreibung der Markengruppen-Element [▶ 54])

8. Betätigen Sie den Button **Start Log data**  in der Toolbar des Realtime Monitors.

⇨ Die Aufnahme des Echtzeitverhaltens beginnt.



Stoppen der Aufzeichnung:

Um eine Laufende Aufzeichnung zu stoppen und die Daten zu sichern, gehen Sie wie folgt vor:

✓ Es läuft eine Realtime-Monitor-Aufzeichnung.

1. Betätigen Sie den Button **Stop data collection**  in der Toolbar des Realtime Monitors.
2. Wählen Sie den Menüeintrag Recordings > **Save data as Recording**.

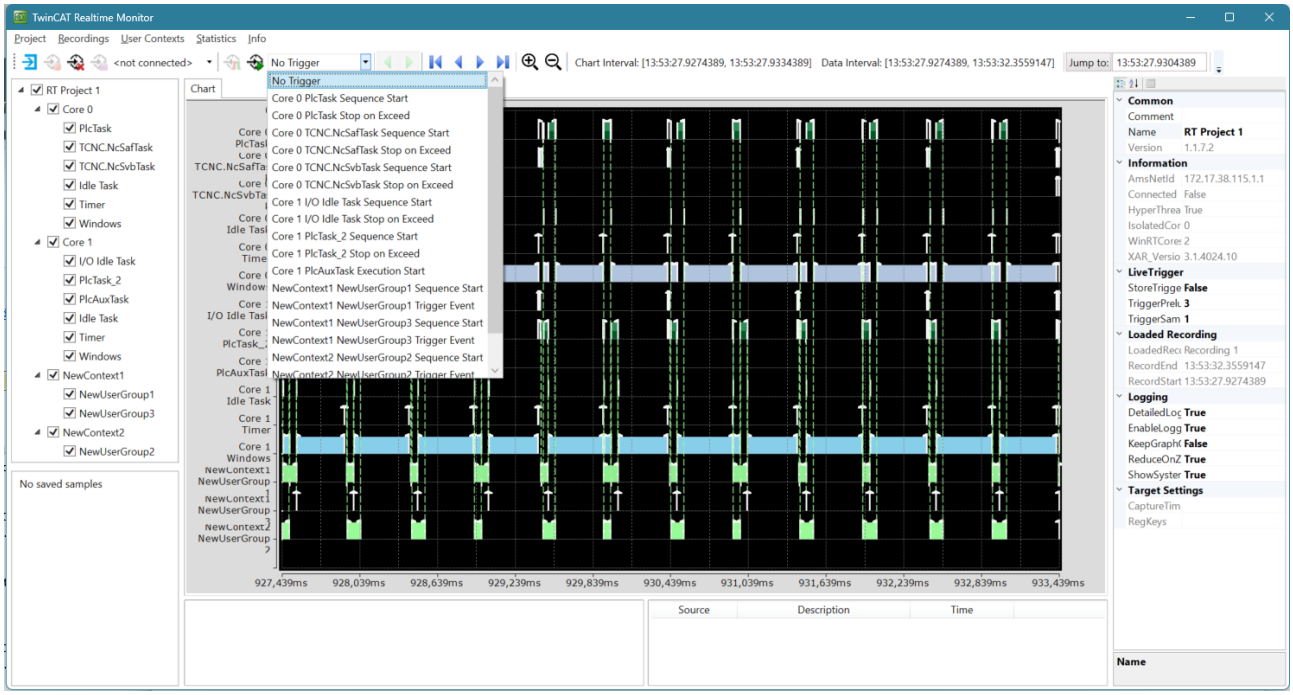
⇨ Die Aufzeichnung des Realtime Monitors ist nun gestoppt und die Daten als Aufnahme (Recording) im Projekt hinterlegt.

6 Konfiguration von Triggern

Um im Datenstrom einer Realtime-Monitor-Aufzeichnung signifikante Ereignisse zu finden, können Trigger verwendet werden.

Für jeden Kontext (Task, User-Kontext) werden in einer Auswahlliste automatisch die folgenden Trigger-Ereignisse eingetragen:

- Sequence Start (für Tasks und User-Kontexte)
- Stop on Exceed (für eine Task)
- Trigger Event (für eine nutzerdefinierte Marke in einem User-Kontext).



Je nachdem, ob Sie mit aufgenommenen Daten oder Live-Daten arbeiten, stehen Ihnen die folgenden Möglichkeiten zur Verfügung:

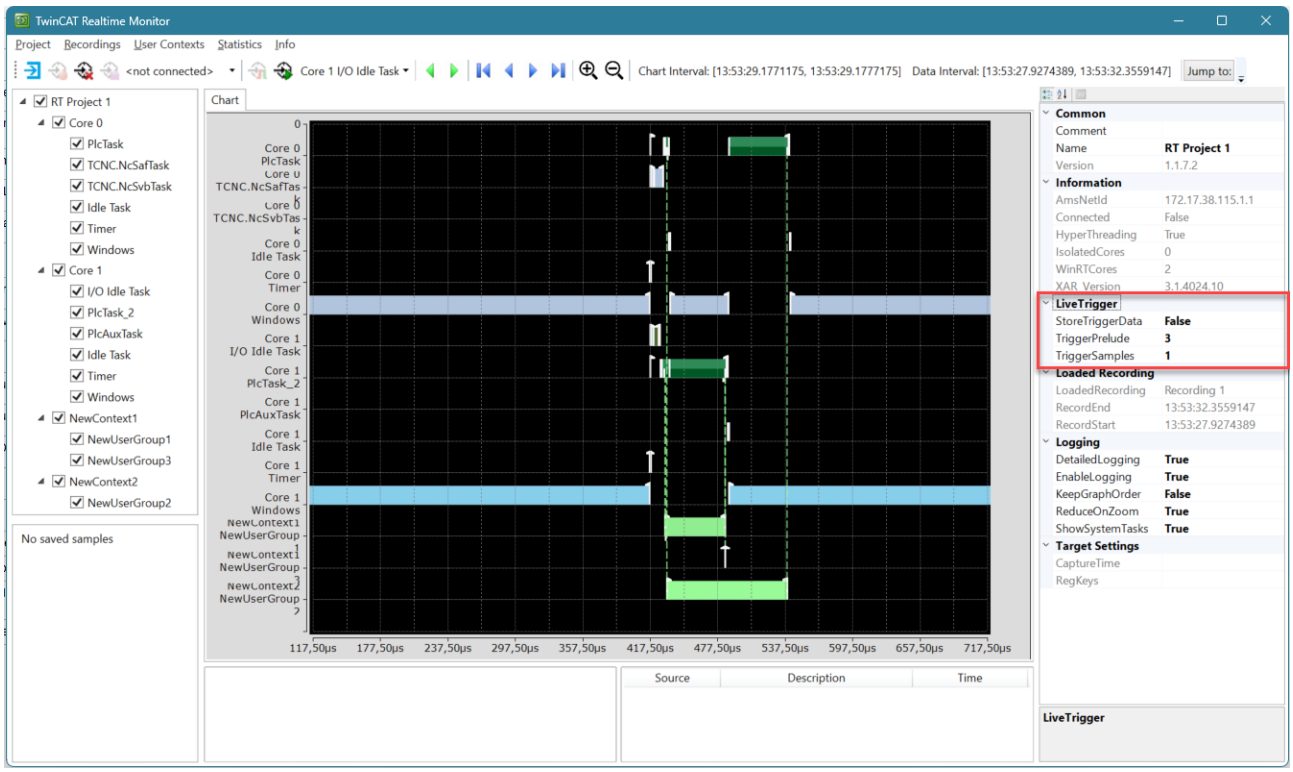
Triggern auf Live-Daten:

Wenn die Steuerung im Run-Mode ist und das Realtime-Monitor-Projekt der aktuellen Konfiguration

entspricht, kann mit Verwendung des Buttons  das Triggern anhand von Live-Daten gestartet werden. Tritt das in der Auswahlbox ausgesuchte Event in den Live-Daten auf, stoppt der Realtime Monitor die Aufnahme automatisch.

In den Einstellungen des Projektknotens können im Bereich „Live Trigger“ die folgenden Einstellungen für das Triggern von Live-Daten hinterlegt werden:

StoreTriggerData:	Definiert, ob die Trigger-Daten als Recording gespeichert werden.
TriggerPrelude:	Definiert, wie viele Sekunden vor dem Ereignis gespeichert werden.
TriggerSamples:	Anzahl der Aufnahmen, die für ein Ereignis gespeichert werden.



Auswerten von aufgenommenen Daten:

Nach der Auswahl des Trigger-Events in der Auswahlbox können Sie mit Hilfe der grünen Pfeiltasten zwischen den einzelnen Zeitpunkten, an denen das Event aufgetreten ist, navigieren. (Siehe Kapitel Symbolleiste - Realtime Monitor Toolbar [▶ 48])

Abspielen der ausgesuchten Trigger-Ereignisse:

Durch Auswahl des Buttons  können Sie die einzelnen Vorkommnisse des ausgewählten Trigger-Events als „Schnelldurchlauf“ abspielen. Damit ist es möglich, schnell einen Eindruck zu erhalten, wann dieses Ereignis auftritt und wie stark der Zeitpunkt des Auftretens unter Umständen variiert.

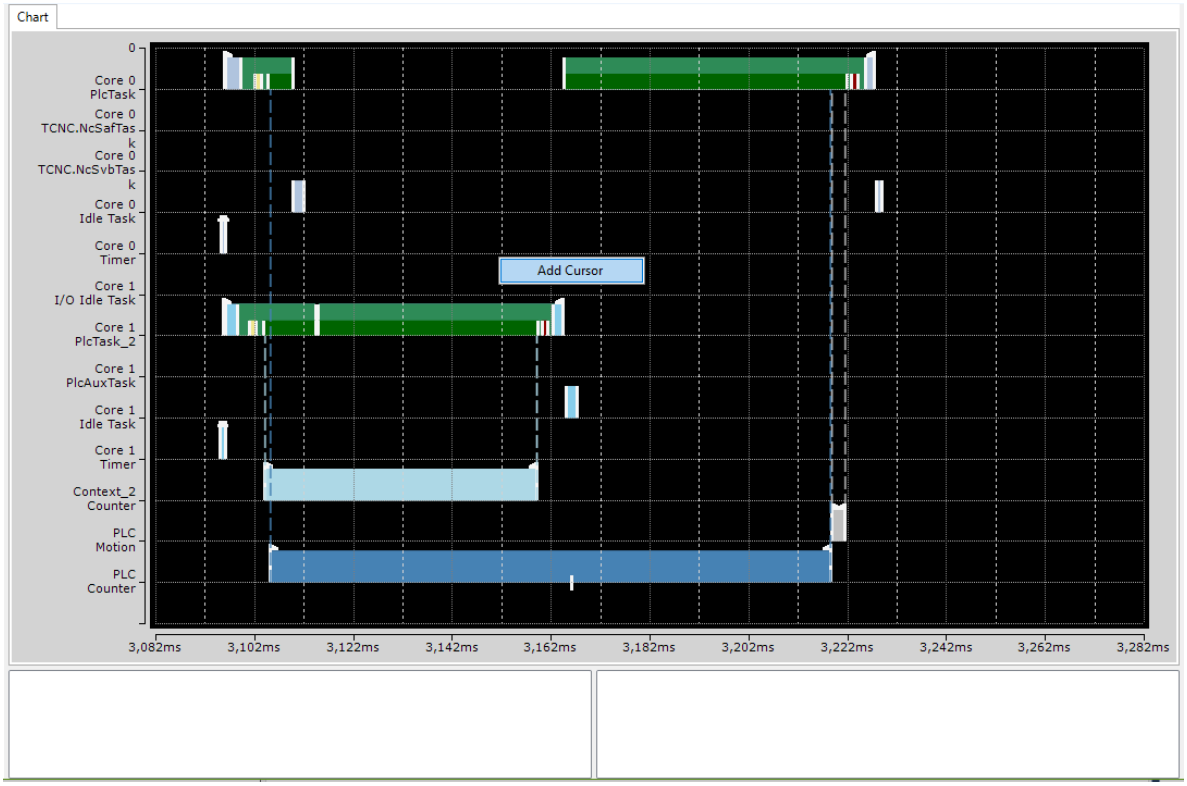
7 Verwendung von Cursors

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:

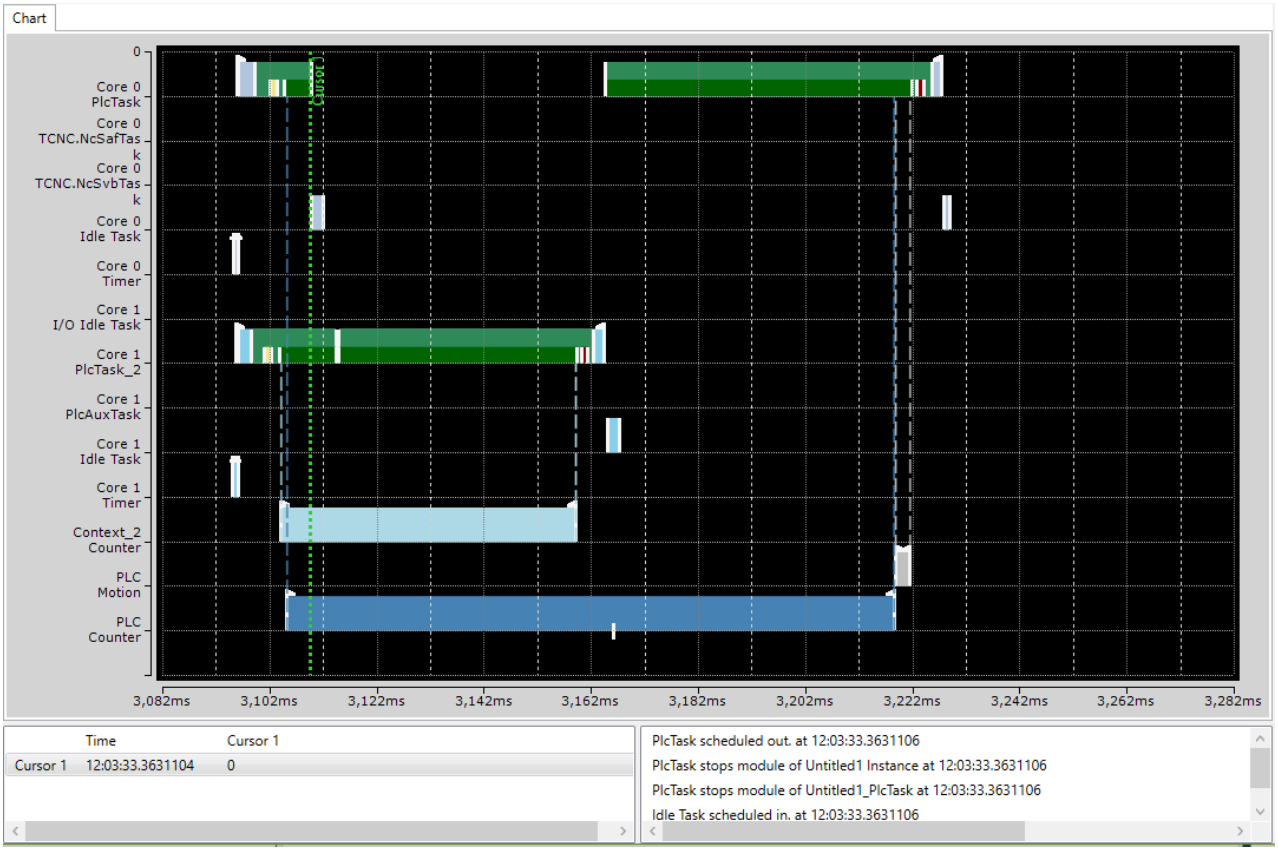
1. Klicken Sie per Rechtsklick innerhalb des Darstellungsbereichs des Charts.
   ⇨ Es öffnet sich ein Kontext-Menü, welches den Befehl **Add Cursor** sowie für alle bereits existierenden Cursors einen Befehl enthält, um diese zu löschen.



⇨

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

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

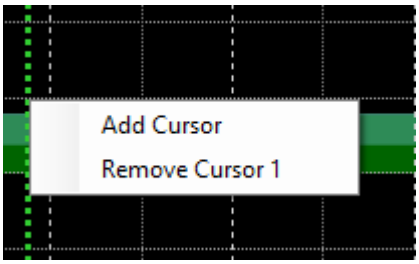


Löschen von Cursors

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.

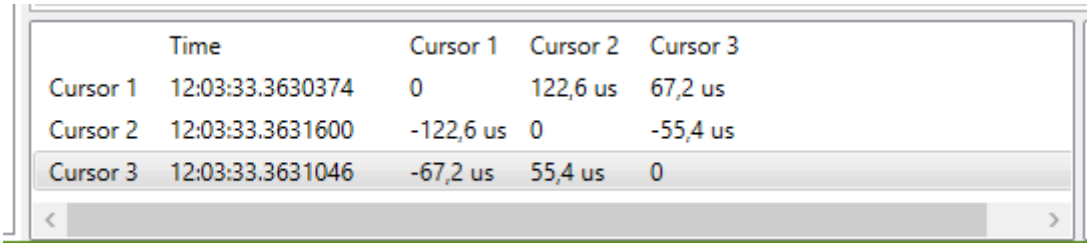
Innerhalb des Cursor-Fensters

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.

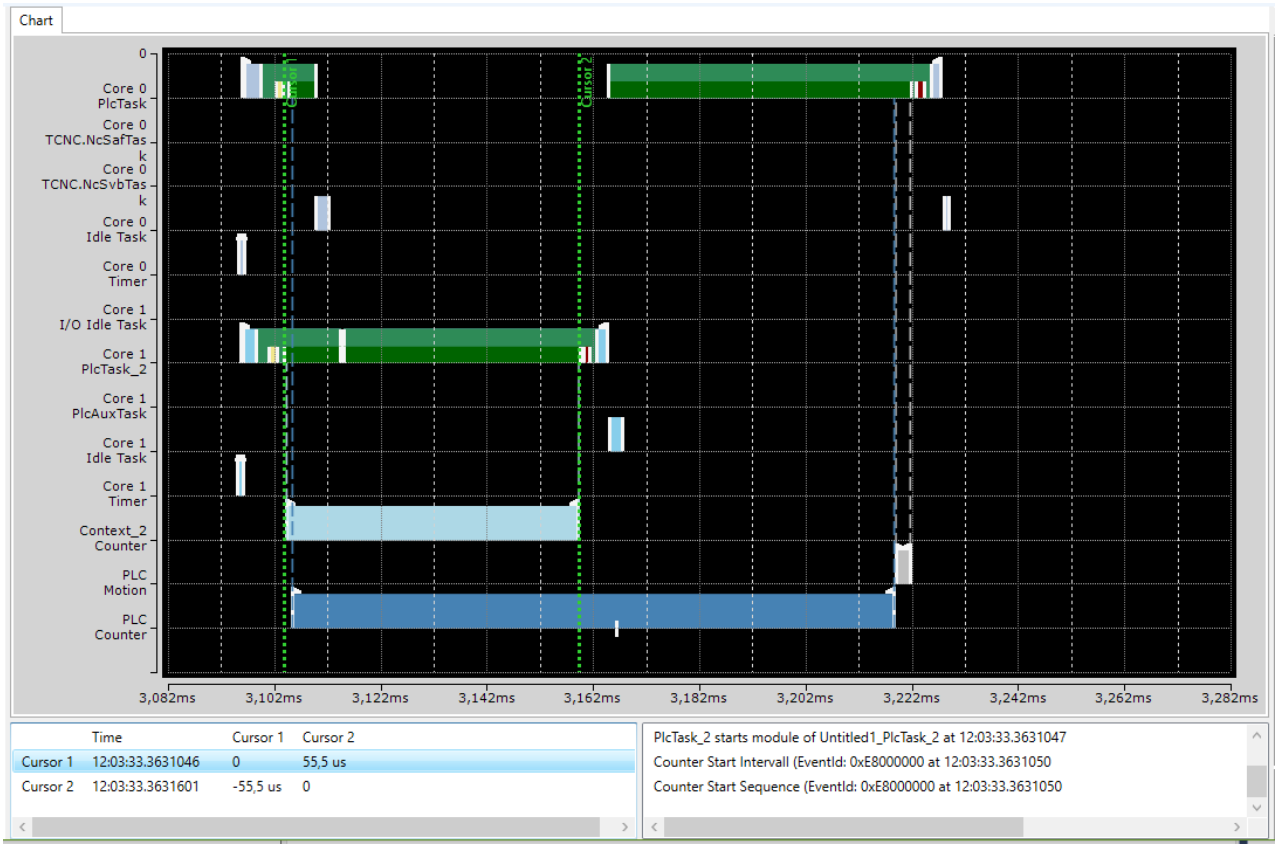


	Time	Cursor 1	Cursor 2	Cursor 3
Cursor 1	12:03:33.3630374	0	122,6 us	67,2 us
Cursor 2	12:03:33.3631600	-122,6 us	0	-55,4 us
Cursor 3	12:03:33.3631046	-67,2 us	55,4 us	0

Ein Doppelklick auf einen Cursor sorgt dafür, dass die Darstellung innerhalb des Charts genau an die Stelle springt, an der 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

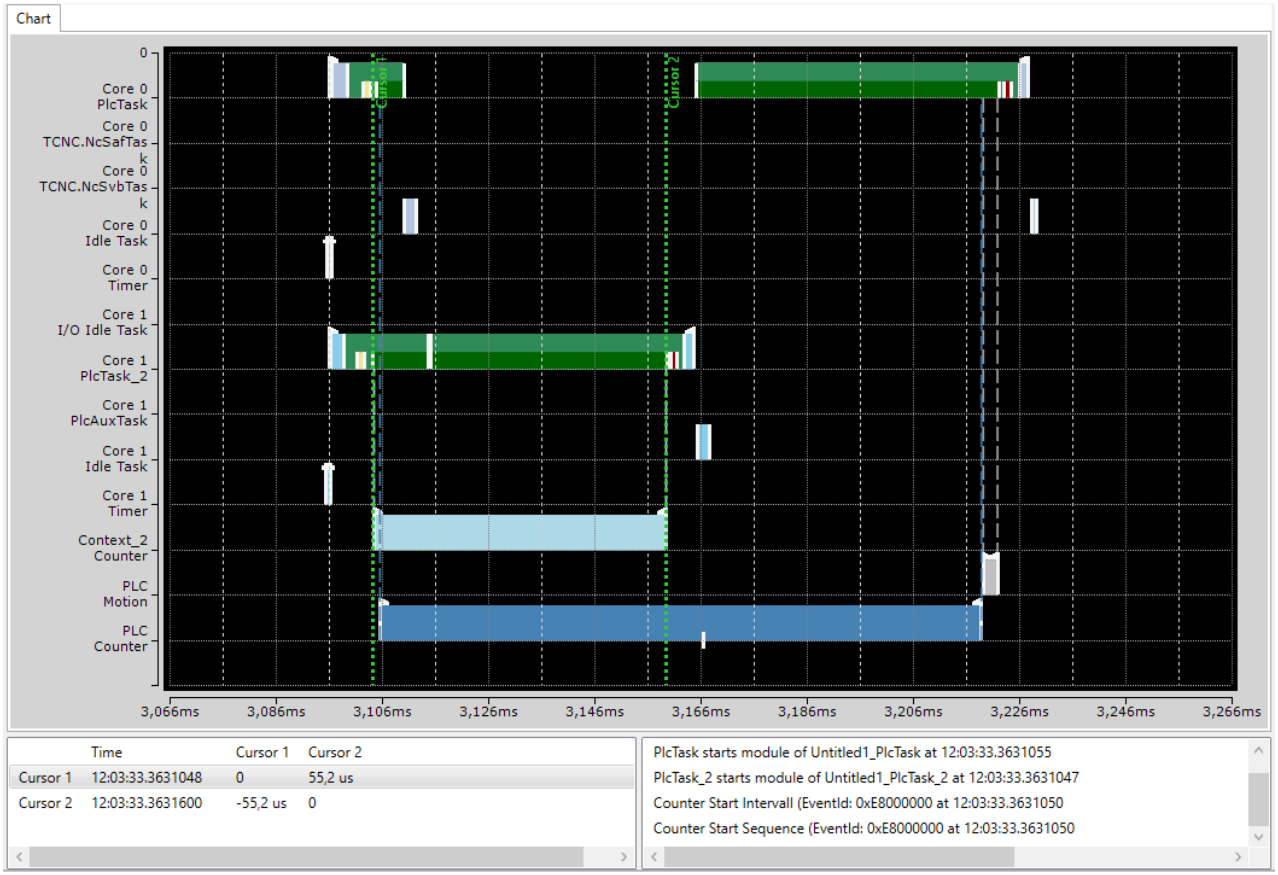
Die folgenden Eigenschaften sind für Cursors verfügbar.

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:

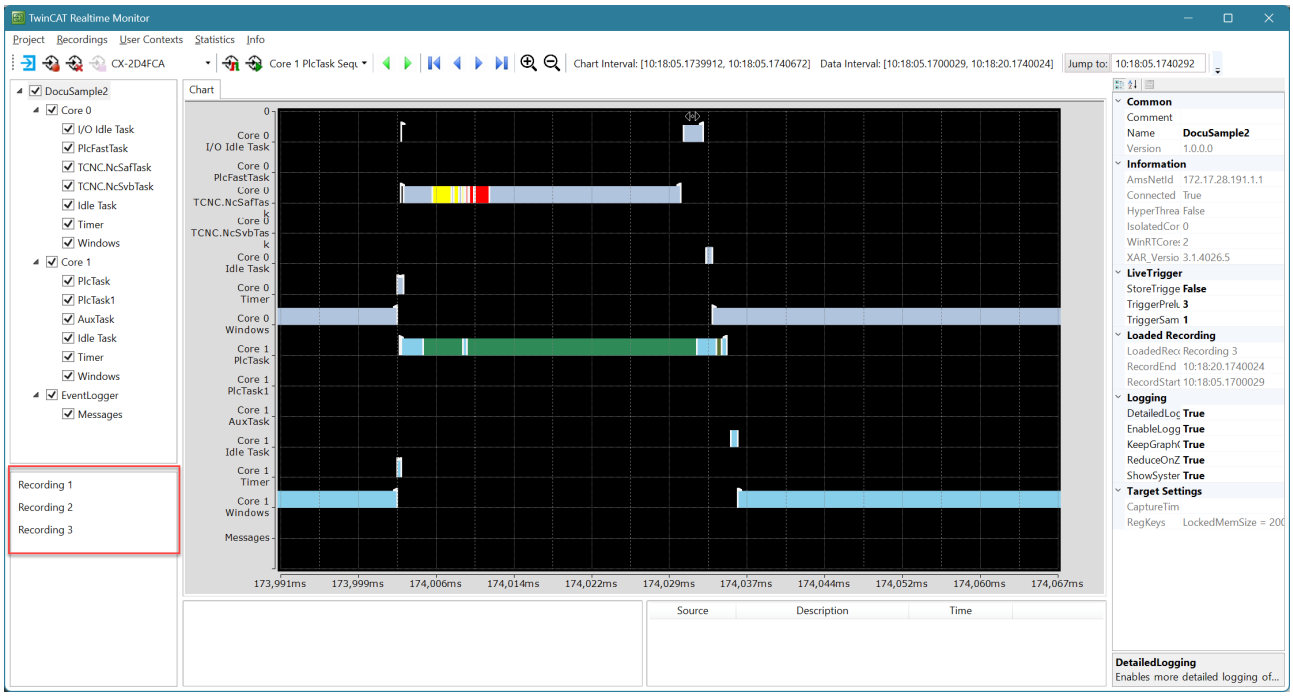
- Die Task PlcTask beginnt mit der Abarbeitung des Laufzeitmoduls Untitled1.
- Die Task PlcTask_2 beginnt ebenfalls mit der Abarbeitung des Laufzeitmoduls Untitled2.
- Der Anwendungsprozess Counter startet sowohl die Sequenz als auch das Intervall.



8 Aufzeichnungen/ Recordings

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

Innerhalb eines Realtime-Monitor-Projekts können mit derselben Konfiguration mehrere Aufzeichnungen (Recordings) durchgeführt und im Projekt hinterlegt werden. Diese werden in der Aufzeichnungsliste angezeigt. (Siehe roter Rahmen im folgenden Bild)



Diese Aufzeichnungen (Recordings) können Sie sowohl über das Kontextmenü als auch über das Menü **Recordings** verwalten.

Speichern einer Aufzeichnung

Nachdem Sie eine Aufzeichnung mit dem Realtime Monitor abgeschlossen haben, können Sie die Daten mit dem Button  (**Save collected data within project**) in der Symbolleiste [▶ 48] oder mit dem Eintrag **„Save data as Recording“** im Menü **Recordings** als Aufzeichnung (Recording) innerhalb des Projekts speichern.

Hinzufügen einer Aufzeichnung

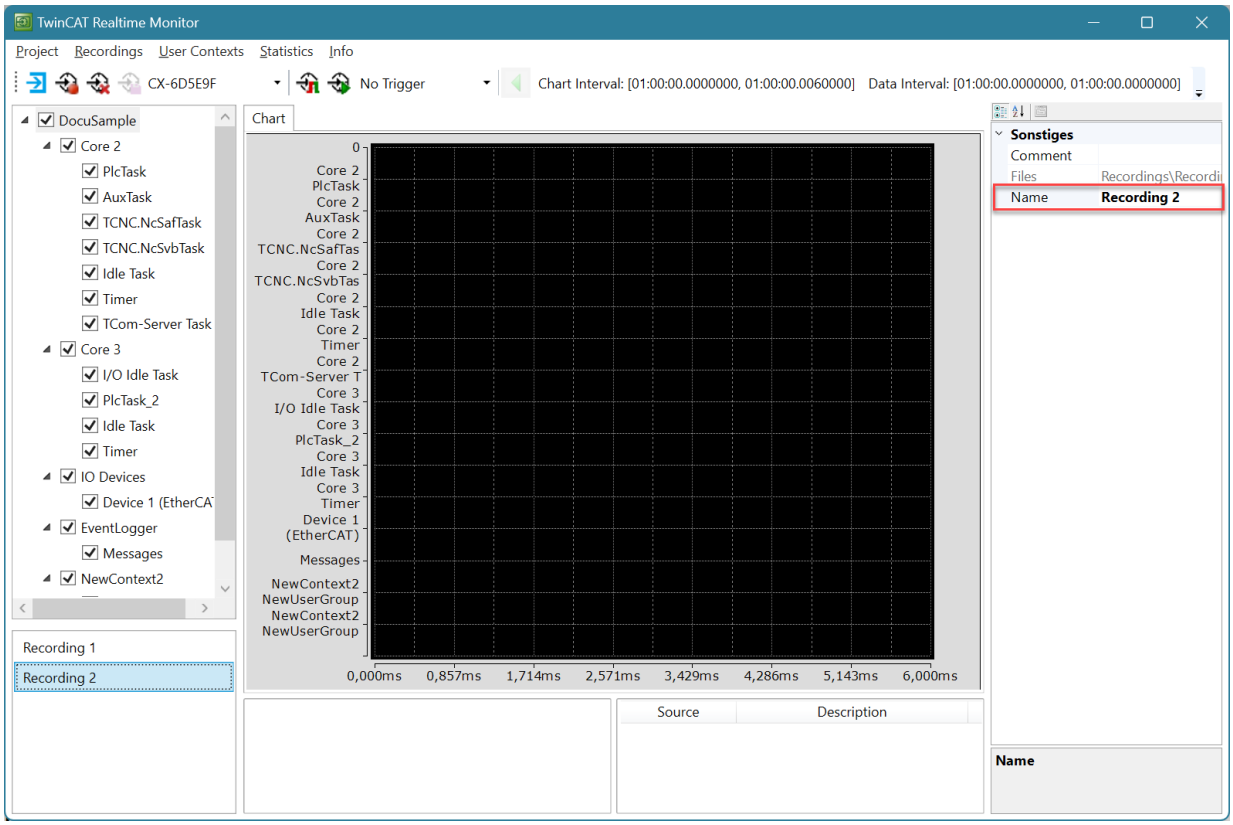
Um bereits aufgenommene Daten zu einem Realtime-Monitor-Projekt hinzuzufügen, verwenden Sie den Menüeintrag **Import Recording File** im Menü **Recordings**. Die Aufzeichnung wird dann am Ende der Liste hinzugefügt und entsprechend aufsteigend nummeriert.

Die Datendateien des Realtime Monitors enthalten nur den Namen des Projekts. 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 Projekts angezeigt. (Siehe Projektknoten [▶ 52])

Umbenennen einer Aufzeichnung:

1. Um eine Aufzeichnung umzubenennen, wählen Sie diese aus und wechseln in das Eigenschaftenfenster.
2. Klicken Sie in das Feld mit dem Namen der Aufzeichnung und ändern diesen.

3. Bestätigen Sie die Änderung durch Drücken der Return-Taste.



Alle Datendateien zu einer Aufzeichnung werden in einem Ordner gespeichert. Das Umbenennen der Aufzeichnung ändert den Namen des Ablageordners, nicht den Namen der Dateien. Wird diese Aufzeichnung in ein anderes Realtime-Monitor-Projekt geladen, geht die Änderung des Namens damit verloren.

Auswählen einer Aufzeichnung

Um eine Aufzeichnung im Anzeigefenster darzustellen, wählen Sie diese aus und verwenden im Kontextmenü den Befehl **Load Recording to Chart** oder doppelklicken Sie auf die darzustellende Aufzeichnung.

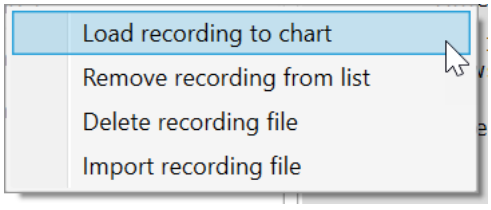
Entfernen einer Aufzeichnung

Soll eine Aufzeichnung aus dem Projekt entfernt aber nicht von der Festplatte gelöscht werden, wählen Sie diese Aufzeichnung in der Aufzeichnungsliste aus, sodass sie markiert ist. Anschließend verwenden Sie den Kontextmenüeintrag **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 sie markiert ist. Anschließend verwenden Sie den **Kontextmenüeintrag Delete Recording File**.

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.
Import recording file	Importiert eine Aufzeichnung in das Realtime-Monitor-Projekt.

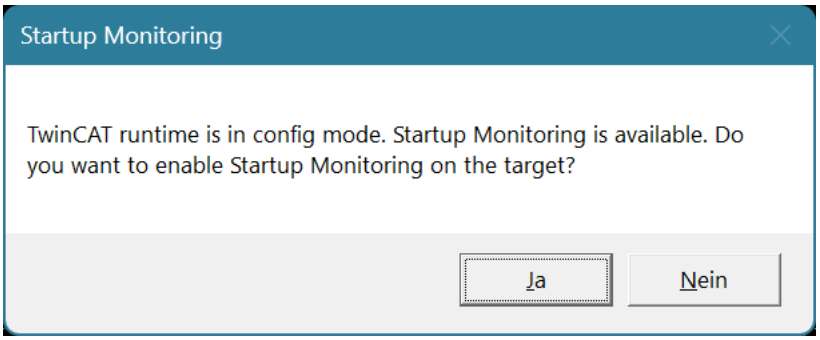
9 Aufzeichnung des Startup-Verhaltens

Ab der TwinCAT 3.1 Version 3.1.4026 besteht die Möglichkeit das Startverhalten eines TwinCAT 3.1 Echtzeitsystems aufzunehmen und mit dem TwinCAT 3 Realtime Monitor darzustellen.

Vorgehen zum Aufnehmen des Startup-Verhaltens:

✓ Führen Sie die Schritte 1-7 des Kapitels Quickstart [▶ 25] aus, an deren Ende Sie ein fertig konfiguriertes Realtime-Monitor-Projekt erhalten.

1. Schalten Sie die TwinCAT-3.1-Runtime in den Config Mode.
2. Betätigen Sie den Aufnahme-Button im Realtime Monitor.
   ⇨ Es erscheint die folgende Meldung:



3. Bestätigen Sie den Dialog mit **Ja**.
4. Schalten Sie die TwinCAT-3.1-Laufzeit in den Run Mode.

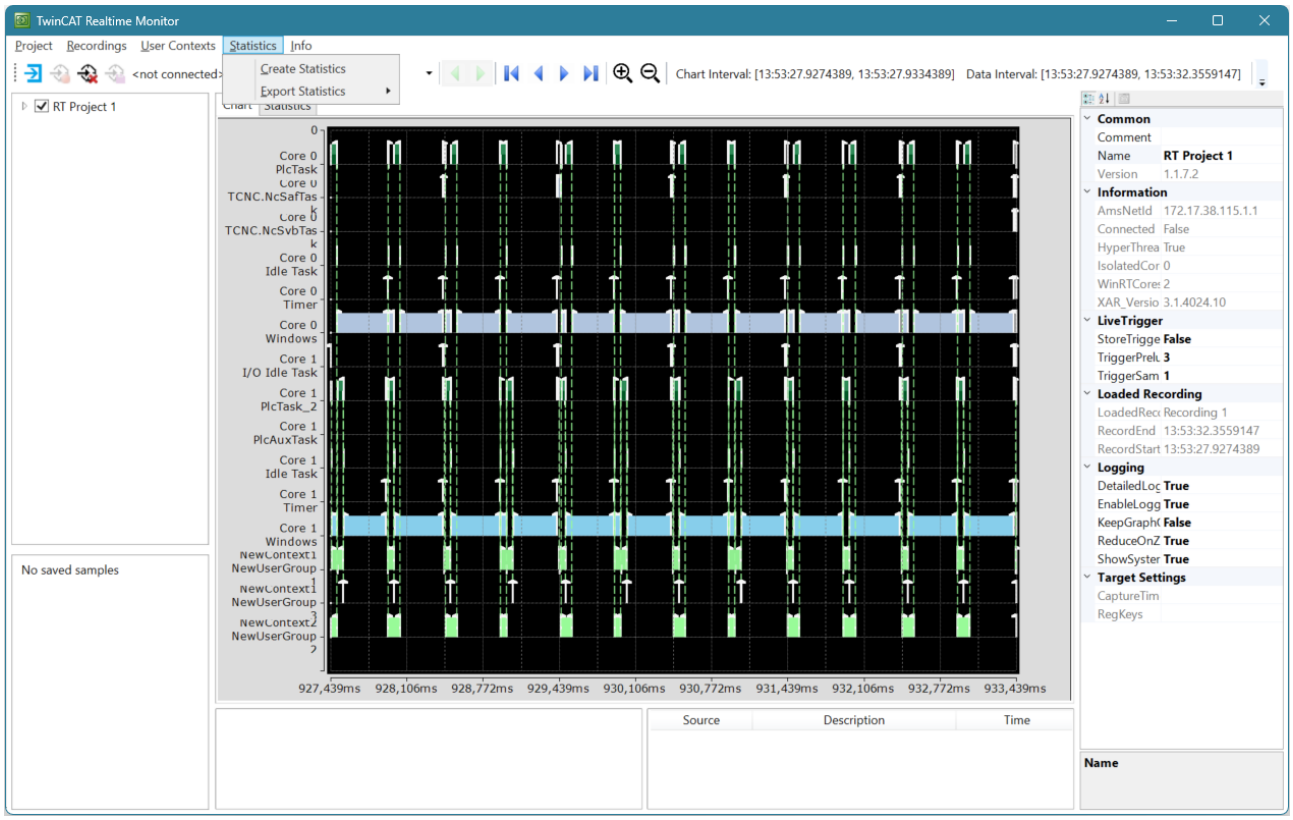
⇨ Der Realtime Monitor startet selbstständig die Aufzeichnung.

10 Statistiken

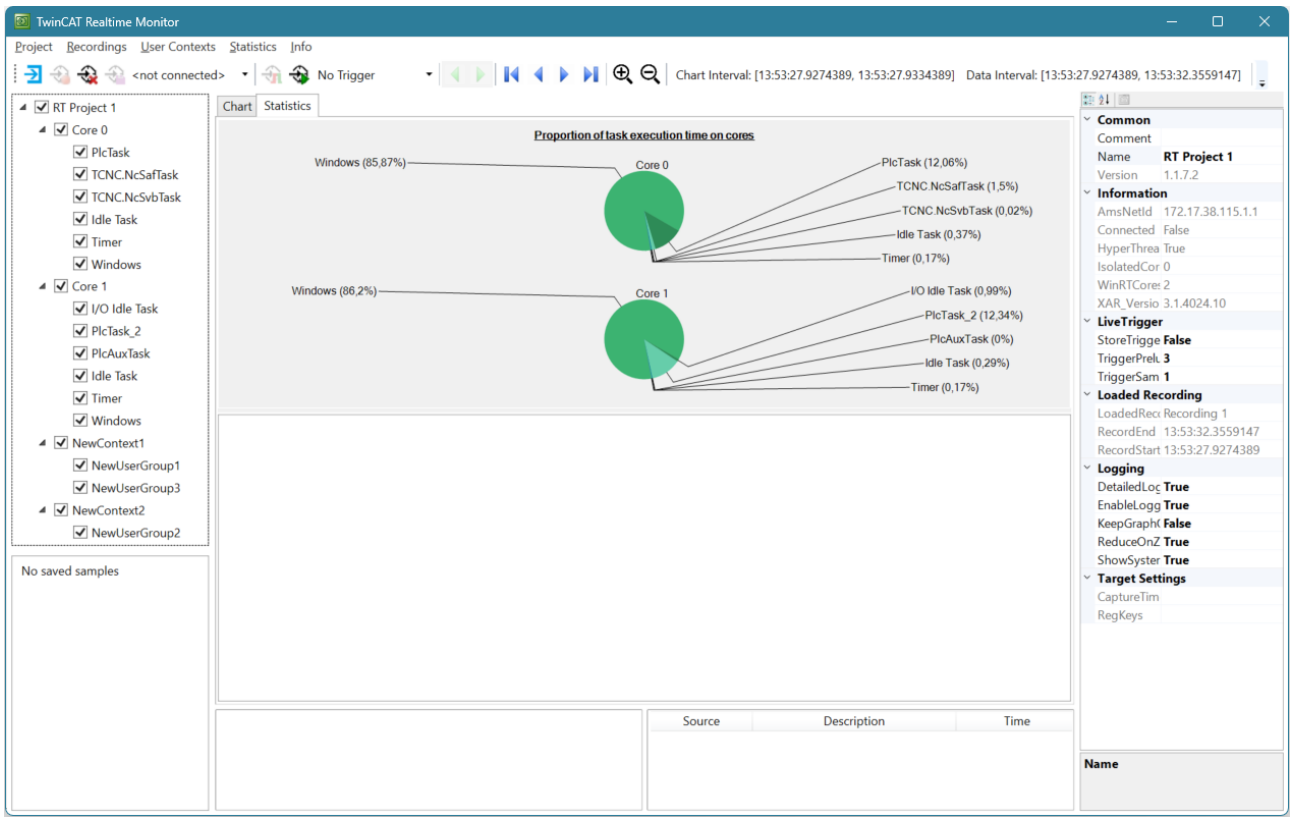
Um Aussagen treffen zu können, wie gut das Echtzeitsystem eingestellt ist, ist es möglich, mit Hilfe des TwinCAT 3 Realtime Monitors eine Statistik zu erzeugen (siehe Kapitel Referenz > Benutzeroberfläche > Statistics [▶ 46]).

Statistik erstellen und anzeigen

1. Um eine Statistik des geladenen Projektes zu erzeugen, wählen Sie im Menü **Statistics** die Option **Create Statitic** aus.



⇨ Die Statistik wird erstellt und es öffnet sich die folgende Ansicht:



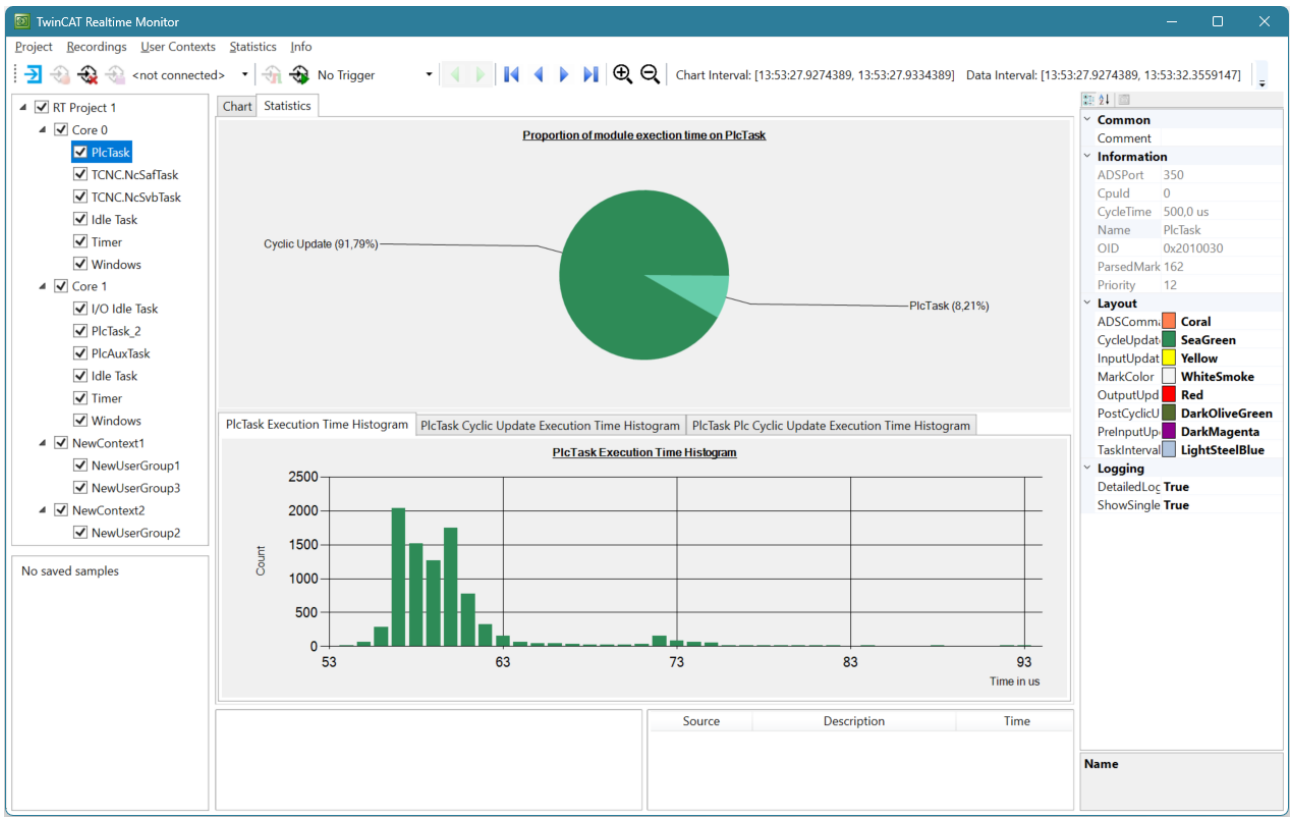
Auf dieser Übersichtsansicht wird die Verteilung der Ausführungszeiten pro Echtzeitkern als Kreisdiagramm dargestellt. Die Summe aller auf einem Echtzeitkern ausgeführten Tasks, inklusive der Idle-Task und unter Umständen des Betriebssystemkontextes (außer auf isolierten Kernen), beträgt immer 100%.

Sie erhalten eine ähnliche Darstellung für einen einzelnen Echtzeitkern, wenn Sie im Projektbaum einen der verwendeten Echtzeitkerne auswählen.

Im Bereich unter dem Kreisdiagramm erhalten Sie die Änderungen der Ausführungszeiten über die Zeit jeweils als Diagramm auf mehrere Reiter (einem pro TwinCAT-Task) aufgeteilt.

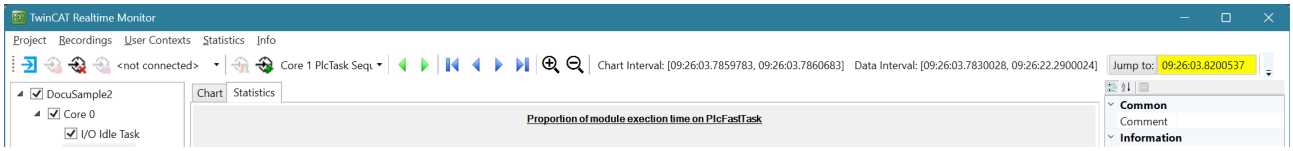
Eine genauere Analyse des zeitlichen Verhaltens ist möglich, wenn Sie eine Echtzeit-Task direkt im Projektbaum auswählen. Für eine SPS-Task erhalten Sie die Ansicht, in welcher das Cycle Update, also der Bereich, in dem der Applikationscode ausgeführt wird, in einem separaten Reiter dargestellt wird.

Im Balkendiagramm sehen Sie die Verteilung der Ausführungszeiten über den aufgenommenen Zeitraum. Um alle Ausführungszeiten zu sehen, scrollen Sie mit dem Mausrad auf dem Balkendiagramm. So zoomen Sie in die Darstellung hinein oder hinaus.



Möchten Sie sehen, in welchem Kontext eine entsprechende Zykluszeit das erste Mal aufgetreten ist, gehen Sie wie folgt vor:

1. Klicken Sie doppelt mit der linken Maustaste auf den entsprechenden Balken in der Verteilung.
   ⇨ Der Zeitstempel wird hinter dem Feld **Jump to** eingetragen und gelb unterlegt.

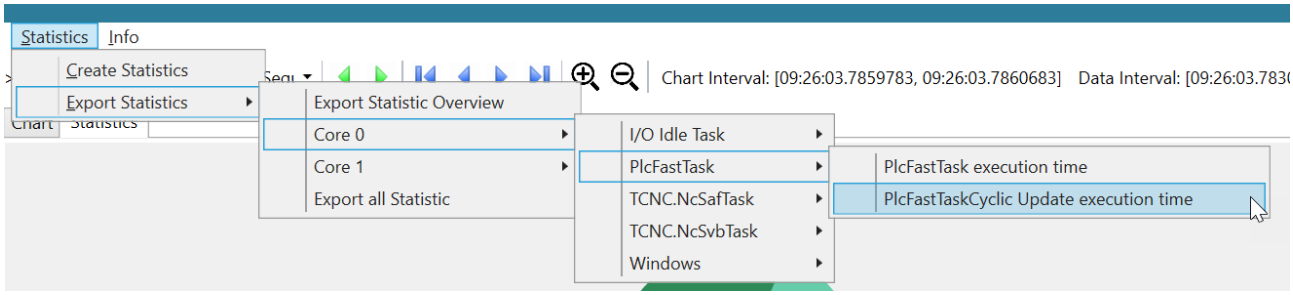


2. Klicken Sie auf den Button **Jump to**, um in die Chart-Ansicht zu wechseln.
   ⇨ Der entsprechende Zeitstempel wird mittig im Chart dargestellt.
3. Um die Markierung im Fenster **Jump to** rückgängig zu machen, klicken Sie in der Chartansicht auf einen beliebigen Bereich.

Statistik exportieren:

Falls in eigenen Tools weitere Analysen durchgeführt werden sollen, können die einzelnen Zeitstempel der Tasks exportiert werden. Der Export erfolgt im Dateiformat *.csv.

4. Wählen Sie im Menü **Statistics -> Export Statistics** den Zeitkontext aus, welchen Sie exportieren möchten.



⇨ Sie erhalten eine *.csv -Datei mit den entsprechenden Zeitstempeln.

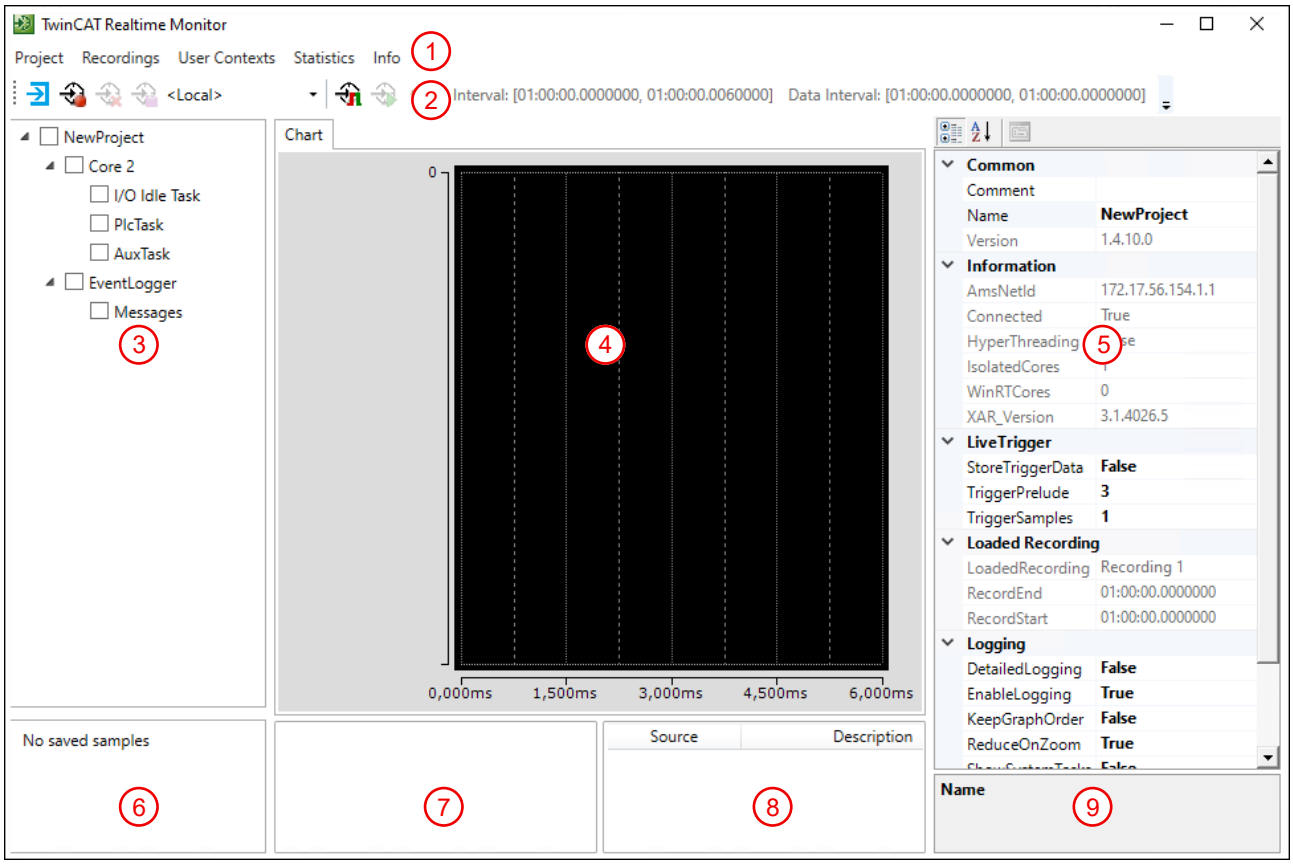
Es ist ebenso möglich die Statistik komplett zu exportieren.

1. Wählen Sie im Menü **Statistics** die Option **Export all Statistic** aus.

⇨ Sie erhalten ein Zip-Archiv, welches für jeden Zeitkontext im Projekt eine eigene *.csv-Datei enthält.

11 Referenz, Benutzeroberfläche

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

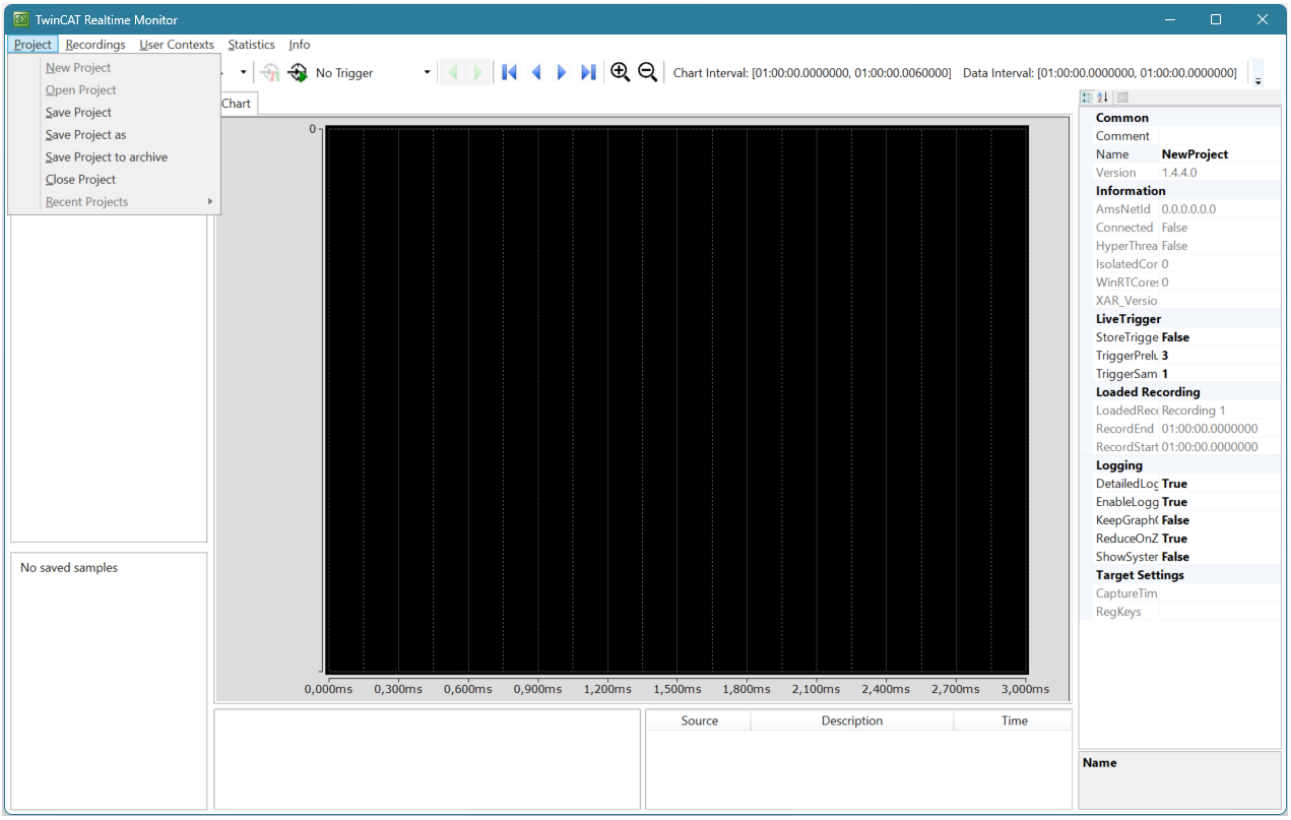


1	Menüleiste
2	Symbolleiste
3	Projektbaum
4	Anzeigefenster
5	Eigenschaftenfenster
6	Aufzeichnungsliste (Recordings)
7	Cursor-Fenster
8	Event-Fenster
9	Beschreibungsanzeige der ausgewählten Eigenschaft

11.1 Menüleiste

Die Menüleiste des TwinCAT 3 Realtime Monitor stellt die im Folgenden beschriebenen Befehle zur Verfügung.

11.1.1 Project



New Project

Funktion: Der Befehl erstellt ein neues Realtime-Monitor-Projekt.

Aufruf: Project > **New Project**

Open Project

Funktion: Der Befehl öffnet ein bestehendes Realtime-Monitor-Projekt.

Aufruf: Project > **Open Project**

Save Project

Funktion: Der Befehl speichert ein bestehendes Realtime-Monitor-Projekt.

Aufruf: Project > **Save Project**

Save Project as

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

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

Save Project to archive

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

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

Close Project

Funktion: Der Befehl schließt ein bestehendes Realtime-Monitor-Projekt.

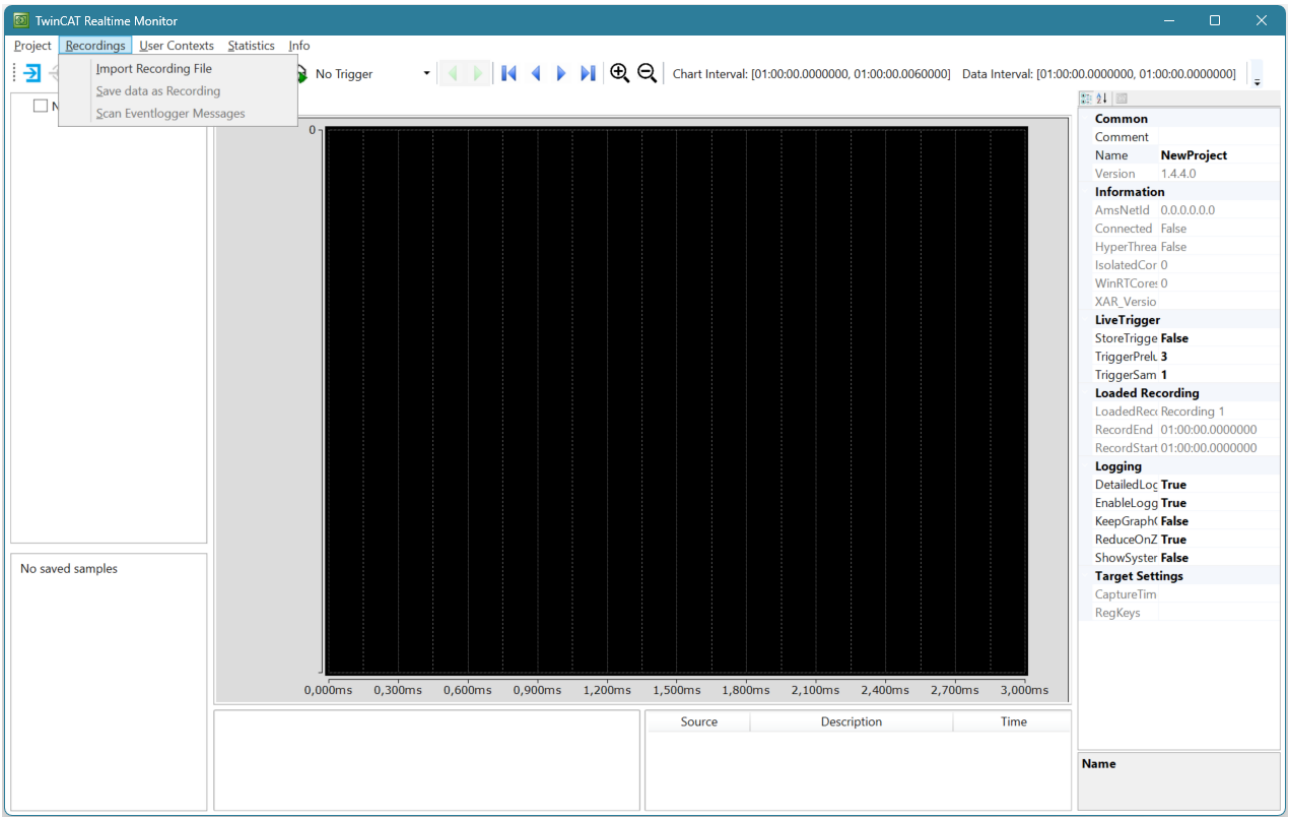
Aufruf: **Project** > **Close Project**

Recent Projects

Funktion: Der Befehl zeigt eine Liste der zuletzt geöffneten Realtime-Monitor-Projekte und ermöglicht es, diese erneut zu öffnen.

Aufruf: Project > Recent Projects

11.1.2 Recordings



Import Recording File

Funktion: Der Befehl importiert eine Aufzeichnungsdatei.

Aufruf: Recordings > **Import Recording File**

Save data as Recording

Funktion: Der Befehl speichert die aufgenommenen Daten als Aufzeichnung.

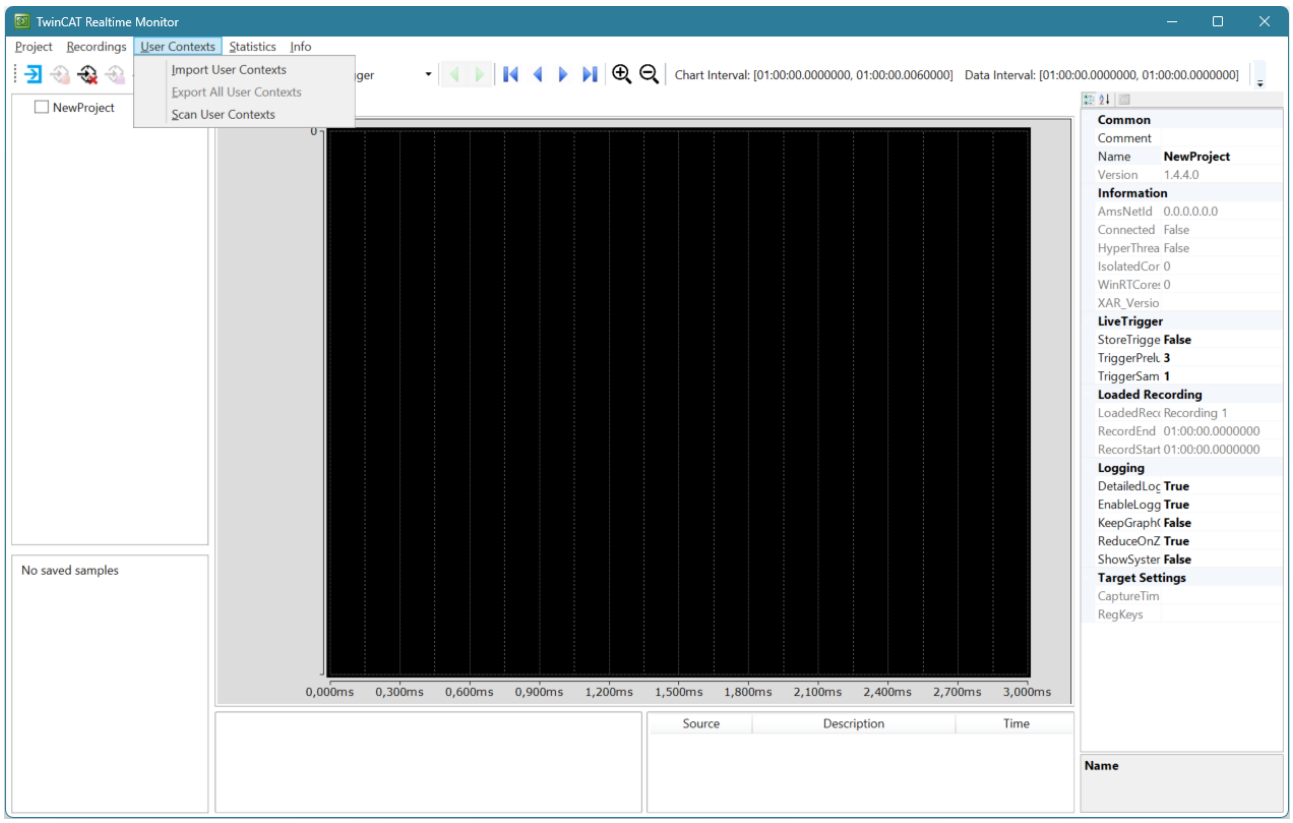
Aufruf: Recordings > **Save data as Recording**

Save Eventlogger Messages

Funktion: Der Befehl scannt die Eventlogger Messages und fügt diese als Events den Realtime-Monitor-Daten hinzu.

Aufruf: Recordings > **Save Eventlogger Messages**

11.1.3 User Contexts



Import User Contexts

Funktion: Der Befehl importiert bestehende Nutzerkontexte in ein 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: **User Contexts** > **Import User Contexts**

Export All User Contexts

Funktion: Der Befehl exportiert bestehende Nutzerkontexte aus einem geöffneten Realtime-Monitor-Projekt.

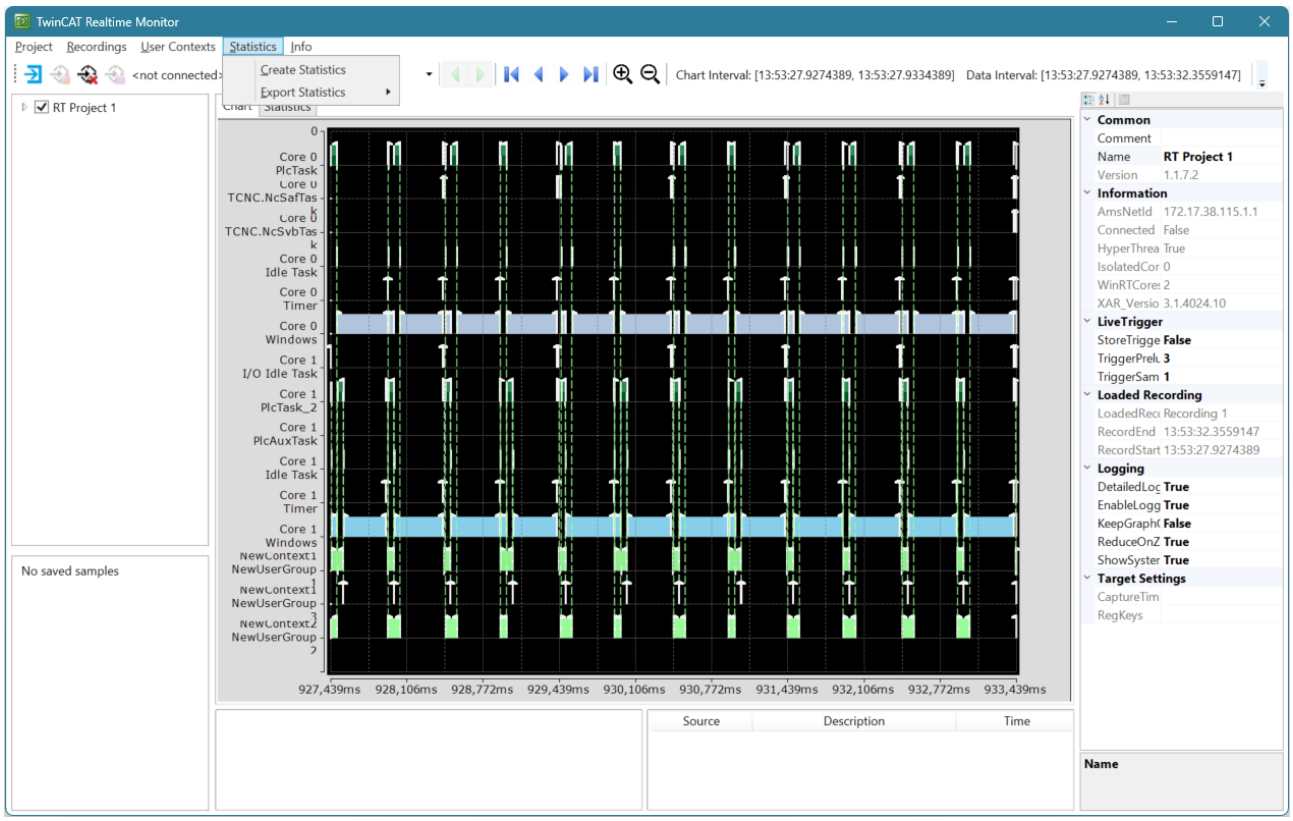
Aufruf: **User Contexts** > **Export All User Contexts**

Scan User Contexts

Funktion: Der Befehl scannt nach bestehenden Nutzerkontexten und fügt diese in ein Realtime-Monitor-Projekt ein.

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

11.1.4 Statistics

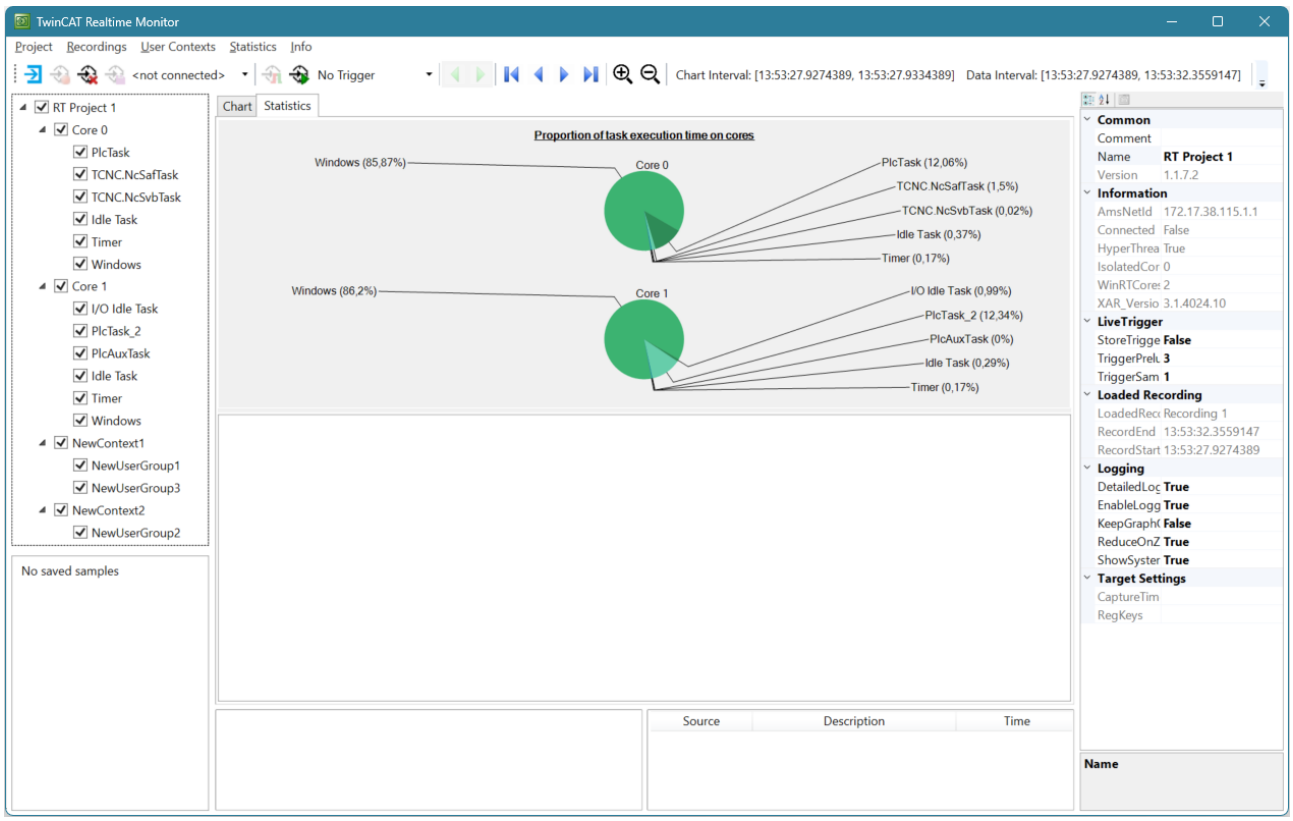


Create Statistics

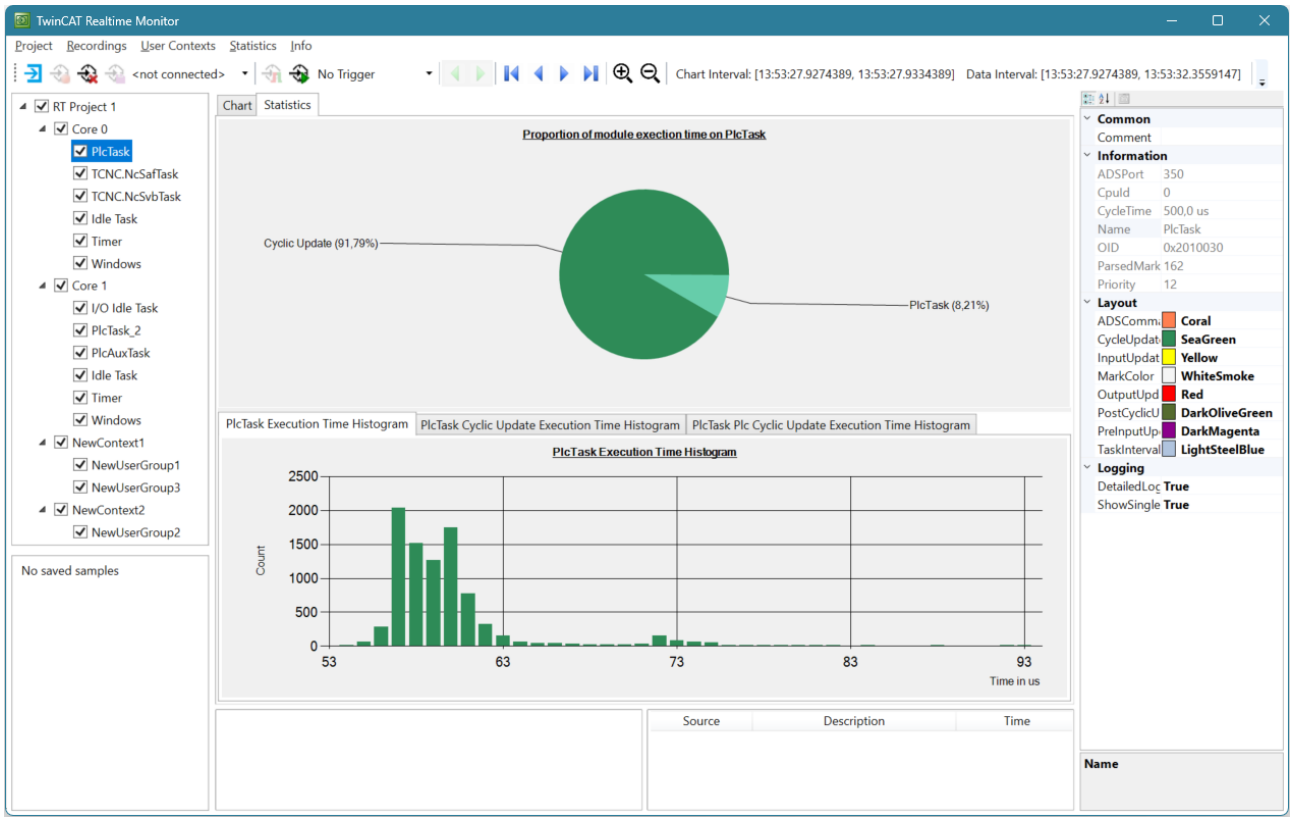
Funktion: Der Befehl wertet die mit dem TwinCAT 3 Realtime Monitor aufgenommenen Marken aus und generiert eine Statistik. Diese wird im Reiter **Statistics** angezeigt.

Aufruf: **Tools** > **Create Statistics**

Beispiele für eine generierte Statistik:



Nach Auswahl einer spezifischen Task:

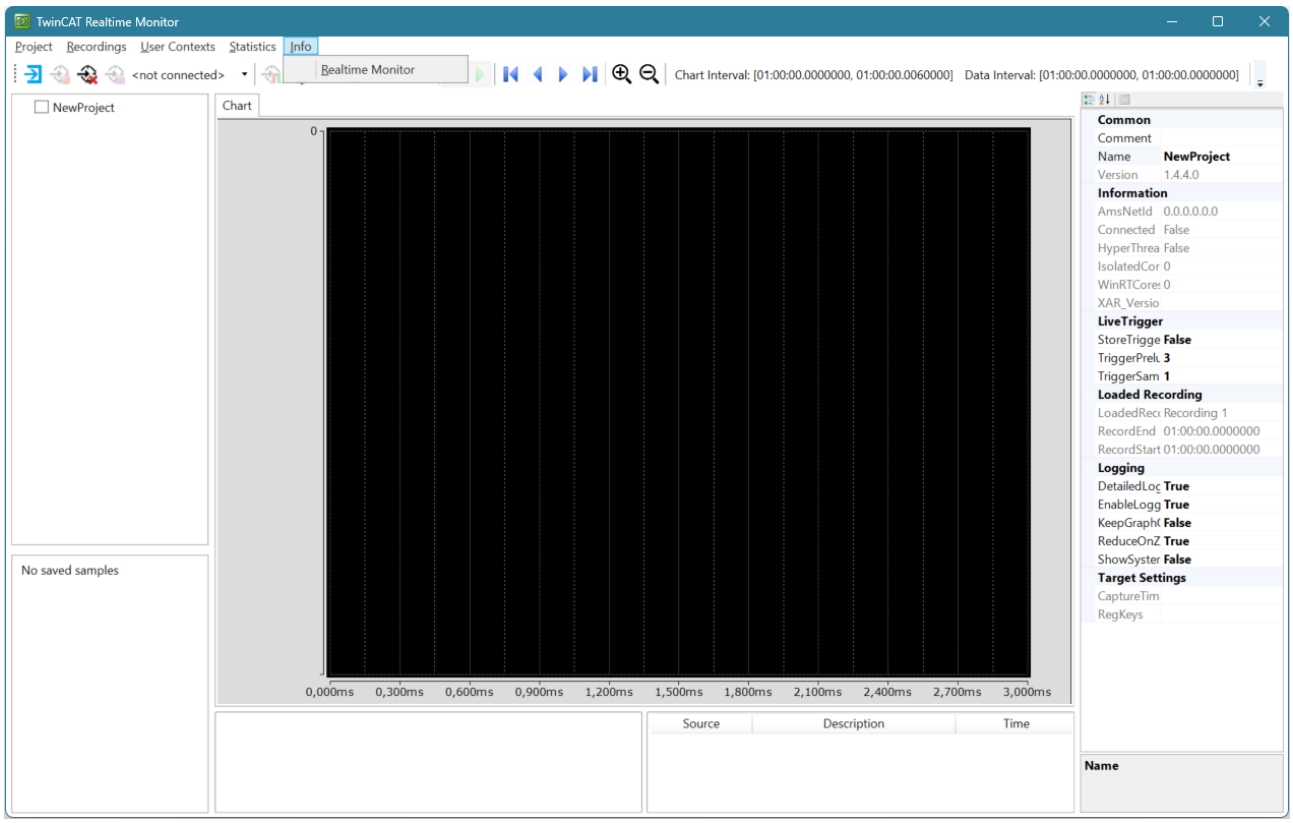


Export Statistics

Funktion: Der Befehl exportiert die ausgewählte Statistik in eine CSV-Datei. Werden alle Statistiken exportiert, werden die einzelnen CSV-Dateien in ein Zip-Archiv gepackt.

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

11.1.5 Info



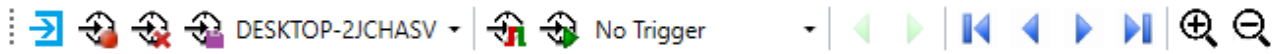
Realtime Monitor

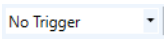
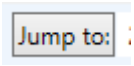
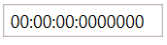
Funktion: Der Befehl öffnet ein Dialog-Fenster, welches die Versionsnummer der installierten TwinCAT 3 Realtime Monitor Version aufzeigt.

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

11.2 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 zu dem Zeitpunkt, der im nächsten Eingabefeld angegebenen wird.
	Eingabefeld zur Eingabe eines Zeitpunktes

11.3 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 [▶ 57] oder FB_RTMon_LogMarkBase [▶ 62]), entweder bezogen auf den ADS-Port des Anwenderprogramms oder anhand einer nutzerdefinierten Kontext-ID.

Die Benennung der nutzerbezogenen Knoten mit entsprechenden Namen erfolgt anhand ihrer Eigenschaften-Seite (siehe Kontextknoten [▶ 53] bzw. Markengruppen-Element [▶ 54]).

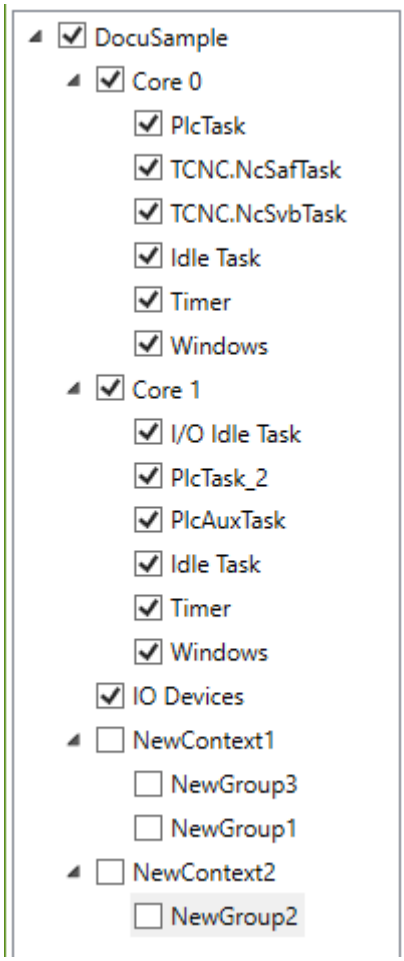
Kontext-Menü-Einträge im Projektbaum

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 Kontextknoten	Fügt eine nutzerbezogene Markengruppe hinzu.
Remove User Context	Nutzerbezogener Kontextknoten	Löscht einen nutzerbezogenen Kontext.
Export User Context	Nutzerbezogener Kontextknoten	Exportiert einen nutzerbezogenen Kontext inklusive aller Unterelemente.
Remove User Group	Nutzerbezogener Markengruppen-Knoten	Löscht eine nutzerbezogene Markengruppe.

Beispiel:

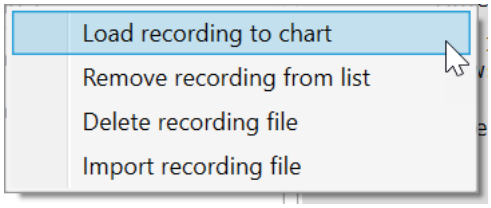
Die folgende Abbildung zeigt die Darstellung eines Projektbaums, wie er nach dem Starten der Aufzeichnung automatisch erzeugt wird. Die Abfrage, ob nach User-Kontexten gesucht werden soll, wurde mit **Ja** bestätigt. Neben den System-Tasks verteilt auf die Rechenkerne Core 0 und Core 1, werden auch 3 nutzerbezogene Markengruppen erzeugt, die hier aber noch nicht benannt wurden.



11.4 Aufzeichnungsliste

In einem Realtime Monitor Projekt können mehrere Aufzeichnungen verwaltet werden. Diese werden in der Aufzeichnungsliste dargestellt. Weitere Informationen zum Thema Aufzeichnungen finden Sie im Kapitel Aufzeichnungen/ Recordings [▶ 34].

Kontextmenü-Einträge in der Aufzeichnungsliste



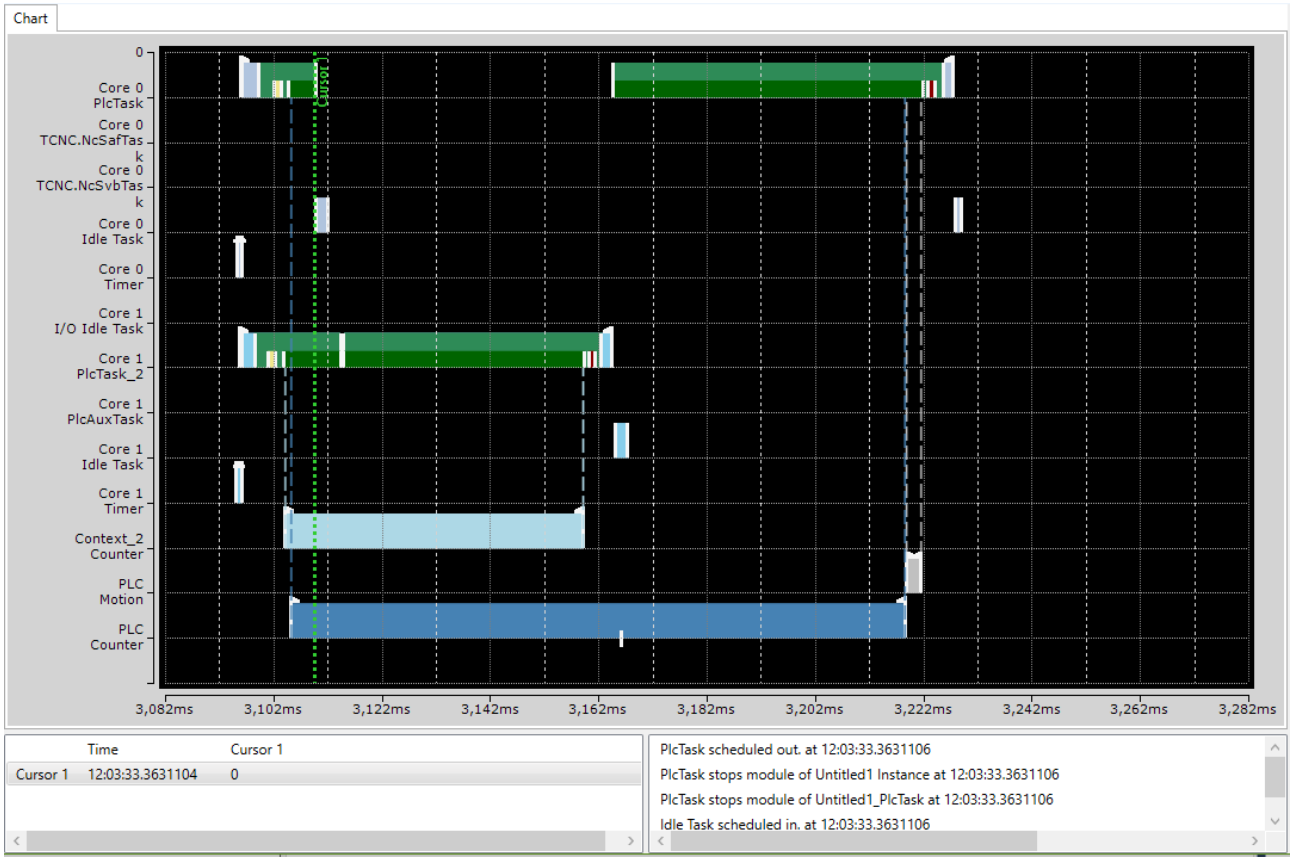
Die folgende Tabelle zeigt alle Kontextmenü-Einträge:

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.
Import recording file	Importiert eine Aufzeichnung in das Realtime-Monitor-Projekt.

11.5 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 [▶ 48]) und analog mit der Maus und ähnlicher Bediengeräte, kann innerhalb der Darstellung des Anzeigefensters navigiert und die Darstellung vergrößert / verkleinert werden.



Um Zeitmessungen oder Analysen vorzunehmen (siehe hierzu Verwendung von Cursors [▶ 30]), setzen, löschen oder verschieben Sie Cursors mit Hilfe des Kontextmenüs.

11.6 Eigenschaftenfenster

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

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

11.6.1 Projektknoten

Folgende Einstellungen stehen auf dem Projektknoten des TwinCAT 3 Realtime Monitors zur Verfügung:

Eigenschaft	Bedeutung
Common	
Comment	Kommentar/Bemerkung zum Projekt
Name	Name des Projekts
Version	Version des Projekts
Information	
AmsNetId	AmsNetId des Zielsystems
Connected	Verbindungsstatus zum Zielsystem
HyperTreading	Zeigt an, ob Hyperthreading aktiv ist.
IsolatedCores	Zeigt die Anzahl der im Projekt verwendeten isolierten Kerne an.
WinRTCores	Zeigt die Anzahl der im Projekt verwendeten Windows-Echtzeit-Kerne an
XARVersion	Zeigt die TwinCAT-Version des Zielsystems an.
Live Trigger	
StoreTriggerData	Speichert Trigger-Daten als Aufzeichnung (Recording).
TriggerPrelude	Definiert die Zeit in Sekunden vor einem Trigger Event (Trigger-Vorlauf).
TriggerSamples	Definiert die Anzahl der Trigger Samples, die gespeichert werden sollen.
Loaded Recording	
LoadedRecording	Name der aktuell geladenen Aufzeichnung
RecordingEnd	Endzeitpunkt der Aufzeichnung
RecordingStart	Startzeitpunkt der Aufzeichnung
Logging	
DetailedLogging	Schaltet das detaillierte Logging ein.
EnableLogging	Aktiviert das Logging.
KeepGraphOrder	Behält die Reihenfolge der Kontexte in der grafischen Darstellung bei, auch nachdem sie unsichtbar und wieder sichtbar gemacht wurden.
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.
TargetSettings/ Capture Time	Zeitstempel, wann die Aufnahme auf dem Zielsystem aufgenommen wurde. Hierfür wird die Uhrzeit des Zielsystems verwendet.
RegKeys	Zeigt an, welche echtzeitbeeinflussende Registry-Keys gesetzt sind.

Die im Bereich **Logging** aufgeführten Eigenschaften gelten immer für alle Unterelemente. Diese Eigenschaften gelten also auf Projektebene 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 Werts auf Projektebene werden die Werte für alle Unterelemente gesetzt.

11.6.2 Kontextknoten

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

Echtzeitkontext:

Eigenschaft	Bedeutung
Common	
Comment	Optionaler Kommentar
Information	
BaseTime	Basiszeit des Kerns
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.
ShowSingleMarker	Zeigt auch einzelne Marken.

Eventlogger

Eigenschaft	Bedeutung
Common	
Comment	Kommentar
Information	
Id	
Name	Eventlogger
Logging	
EnableLogging	Schaltet das Mitschreiben von Eventlogger-Events an.

Anwendungskontext:

Eigenschaft	Bedeutung
Common	
Comment	Kommentar
Information	
ContextId	Kontext-ID, die bei den Marken übergeben wurde
Name	Name des Kontexts
Logging	
ShowSingleMarker	Zeigt auch die Einzelmarken an.
ShowTaskReference	Zeigt die Task-Referenzen an.

Bei der Verwendung des Bausteins FB_RTMon_LogMark [▶ 57] wird als Kontext-ID automatisch die Port-Nummer des SPS-Laufzeitmoduls eingestellt.

11.6.3 Markengruppen-Element

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

Echtzeit-Tasks:

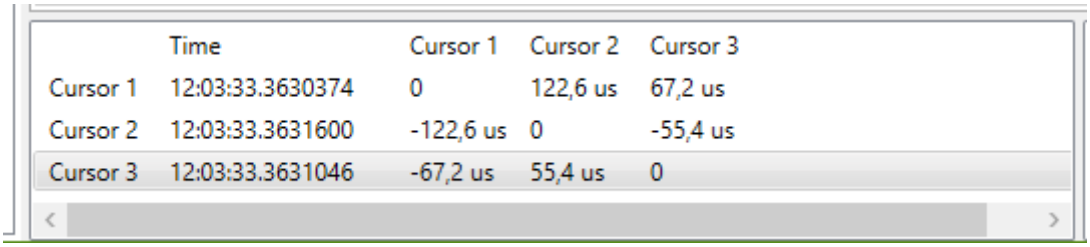
Eigenschaft	Bedeutung
Common	
Comment	Kommentar
Information	
ADSPort	ADS-Port der Task
CpuId	CPU-ID, auf dem die Task ausgeführt wird.
CycleTime	Zykluszeit der Task
EventCount	Anzahl der Ausführungen (innerhalb der Aufnahmezeit)
Name	Name der Task
OID	Objekt-ID der Task
ParsedMasks	Anzahl der gefundenen Marken
Priority	Eingestellte Priorität
Layout	
ADSCommandColor	Farbe des Bereichs zur Abarbeitung der ADS-Kommandos (Standard: Coral)
CycleUdateColor	Farbe des CycleUpdates der Task (Standard: grün)
InputUpdateColor	Farbe des InputUpdates der Task (Standard: gelb)
MarkColor	Markenfarbe (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: gold)
TaskIntervallColor	Farbe der Taskintervall-Markierung (Standard: hellblau)
Logging	
DetailedLogging	Aktivieren des detaillierten Loggings.
ShowSingleMarker	Aktiviert das Anzeigen einzelner Marken.

Anwender-Prozesse:

Eigenschaft	Bedeutung
Common	
Comment	Kommentar
Information	
GroupId	ID der Markengruppe / des Prozesses
Name	Name des darzustellenden Prozesses
ParsedMarks	Anzahl der gefunden Marken
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 Prozessmarken zu Echtzeittasks durch gestrichelte Linien)

11.7 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 Doppelklick auf einen Cursor sorgt dafür, dass die Darstellung innerhalb des Charts genau an die Stelle springt, an der der Cursor steht. Der Cursor wird mittig im Darstellungsbereich angezeigt.

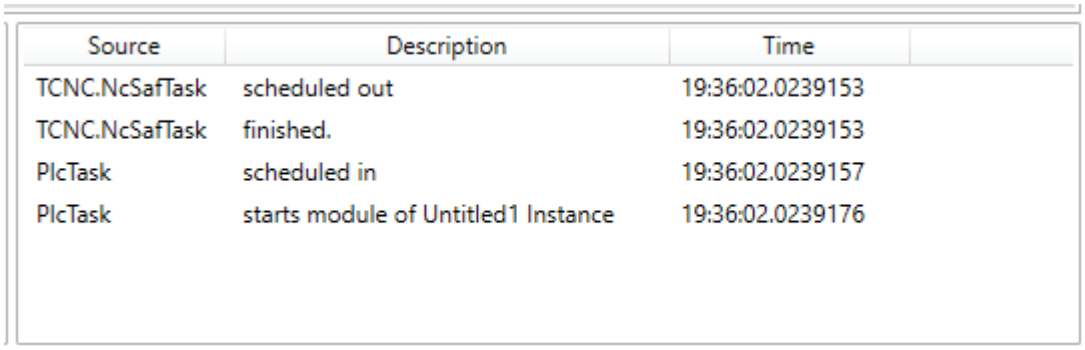
Mithilfe des Kontextmenüeintrags **Remove Cursor** ist es möglich, den jeweils markierten Cursor zu löschen.

Die Verwendung von Cursors ist ausführlich beschrieben unter Verwendung von Cursors [▶ 30].

11.8 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 Scheduler 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

12 SPS API

12.1 Funktionsbausteine

12.1.1 FB_RTMon_LogMark

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

Der FB_RTMon_LogMark erweitert den generischen Funktionsbaustein FB_RTMon_LogMarkBase [▶ 62]. Die Information der Kontext-ID wird dabei automatisch durch den ADS-Port des aufrufenden Tasks initialisiert und muss dadurch beim Aufruf nicht mehr berücksichtigt werden. Die verschiedenen Markenoptionen (siehe TcMarkOption [▶ 64]) wurden zudem in Methodenaufrufe verpackt, so dass der Baustein ohne das vorherige Anlegen einer Marke auskommt. 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 übergeben kann (z. B. Zustand einer Statemachine, Fehlermeldung, etc.).

Eingänge

```
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
```

Name	Typ	Beschreibung
nContextId	UINT	Kontext-ID, für welche diese Marke gesetzt werden soll. Der Initialwert ist *TwinCAT_SystemInfoVarList._AppInfo.AdsPort*. Wird dieser nicht geändert, wird automatisch die ADS-Port-Nummer als Kontext-ID verwendet.
bLogCallingTask	BOOL	Boolscher Parameter, der definiert, ob für die Marke der ausführende Task mit abgespeichert wird. Der Initialwert ist TRUE.

Ausgänge

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

Name	Typ	Beschreibung
bError	BOOL	Boolscher Wert, der anzeigt, ob der Baustein im Fehlerzustand ist.
hrErrorCode	HRESULT	HResult-Error Code. Ist der Ausgang *bError* true, kann an diesem Ausgang der Fehlercode ausgelesen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT 4026.0 oder neuer	PC oder CX (x86, x64, Arm®)	Tc3_RealtimeMonitoring 1.1.3

12.1.1.1 LogIntervalStart

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

Rückgabewert

Name	Typ	Beschreibung
LogIntervalStart	HRESULT	Liefert im Fehlerfall den Fehlercode zurück.

Eingänge

```
VAR_INPUT
    nGroupId    : UINT; // Defines the group
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Name	Typ	Beschreibung
nGroupId	UINT	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
nEventId	UINT	Optionale EventId

12.1.1.2 LogIntervalStop

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

Rückgabewert

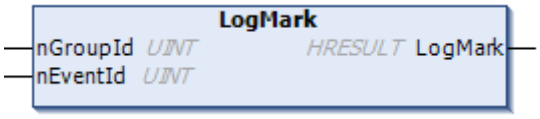
Name	Typ	Beschreibung
LogIntervalStop	HRESULT	Liefert im Fehlerfall den Fehlercode zurück.

Eingänge

```
VAR_INPUT
    nGroupId    : UINT; // Defines the group
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Name	Typ	Beschreibung
nGroupId	UINT	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
nEventId	UINT	Optionale EventId

12.1.1.3 LogMark



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.

Rückgabewert

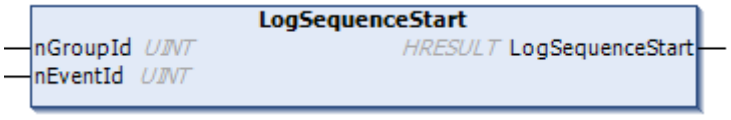
Name	Typ	Beschreibung
LogMark	HRESULT	Liefert im Fehlerfall den Fehlercode zurück.

Eingänge

```
VAR_INPUT
    nGroupId    : UINT; // Defines the group
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Name	Typ	Beschreibung
nGroupId	UINT	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
nEventId	UINT	Optionale EventId

12.1.1.4 LogSequenceStart



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

Rückgabewert

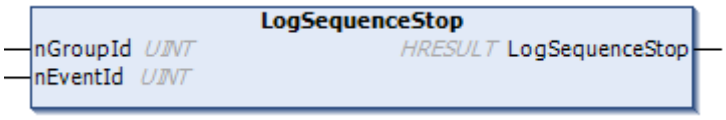
Name	Typ	Beschreibung
LogSequenceStart	HRESULT	Liefert im Fehlerfall den Fehlercode zurück.

Eingänge

```
VAR_INPUT
    nGroupId    : UINT; // Defines the group
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Name	Typ	Beschreibung
nGroupId	UINT	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
nEventId	UINT	Optionale EventId

12.1.1.5 LogSequenceStop



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

Rückgabewert

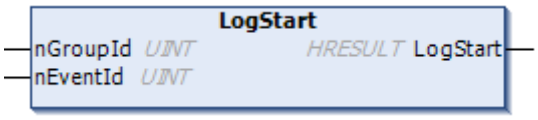
Name	Typ	Beschreibung
LogSequenceStop	HRESULT	Liefert im Fehlerfall den Fehlercode zurück.

Eingänge

```
VAR_INPUT
    nGroupId    : UINT; // Defines the group
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Name	Typ	Beschreibung
nGroupId	UINT	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
nEventId	UINT	Optionale EventId

12.1.1.6 LogStart



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.

Rückgabewert

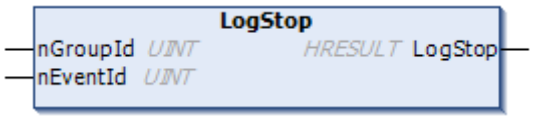
Name	Typ	Beschreibung
LogStart	HRESULT	Liefert im Fehlerfall den Fehlercode zurück.

Eingänge

```
VAR_INPUT
    nGroupId    : UINT; // Defines the group
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Name	Typ	Beschreibung
nGroupId	UINT	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
nEventId	UINT	Optionale EventId

12.1.1.7 LogStop



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.

Rückgabewert

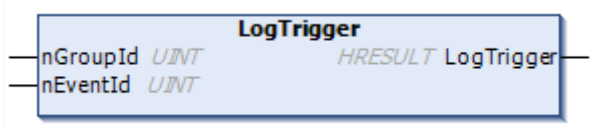
Name	Typ	Beschreibung
LogStop	HRESULT	Liefert im Fehlerfall den Fehlercode zurück.

Eingänge

```
VAR_INPUT
    nGroupId    : UINT; // Defines the group
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Name	Typ	Beschreibung
nGroupId	UINT	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
nEventId	UINT	Optionale EventId

12.1.1.8 LogTrigger



Die Methode erstellt eine Marke mit einem Trigger für die übergebene Marken-ID. Der Realtime Monitor Server kann so konfiguriert werden, dass er anhand dieses Triggers die Aufnahme startet.

Rückgabewert

Name	Typ	Beschreibung
LogTrigger	HRESULT	Liefert im Fehlerfall den Fehlercode zurück.

Eingänge

```
VAR_INPUT
    nGroupId    : UINT; // Defines the group
    nEventId    : UINT; // Set to distinguish different events inside the group
END_VAR
```

Name	Typ	Beschreibung
nGroupId	UINT	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
nEventId	UINT	Optionale EventId

12.1.2 FB_RTMon_LogMarkBase

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

Im Gegensatz zum Funktionsbaustein FB_RTMon_LogMark [▶ 57], 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).

Ausgänge

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

Name	Typ	Beschreibung
bError	BOOL	TRUE, wenn ein Fehler aufgetreten ist.
hrErrorCode	HRESULT	Gibt den aufgetretenen Fehlercode aus.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT 4026.0 oder neuer	PC oder CX (x86, x64, Arm®)	Tc3_RealtimeMonitoring 1.1.3

12.1.2.1 LogMark

Die Methode erstellt eine Marke für die übergebene Markengruppen-ID. Der Markentyp wird anhand des Parameters nMarkCtrl (siehe TcMarkOption [▶ 64]) übergeben.

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

Rückgabewert

Name	Typ	Beschreibung
LogMark	HRESULT	Liefert im Fehlerfall den Fehlercode zurück.

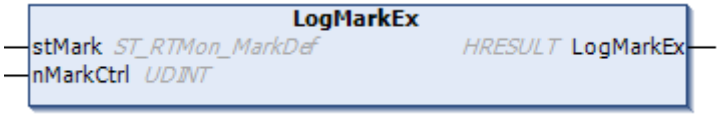
Eingänge

```
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
```

Name	Typ	Beschreibung
nContextId	UINT	Definiert die Kontext-Id, unter der die Marke im TwinCAT 3 Realtime Monitor gruppiert werden soll.
nGroupId	UINT	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
nEventId	UINT	Optionale EventId
nMarkCtrl	UDINT	Definiert den Markentyp.

12.1.2.2 LogMarkEx



Die Methode erstellt eine Marke. Die Markendefinition erfolgt anhand des Datentyps ST_RTMon_MarkDef [▶ 63]. Der Markentyp wird anhand des Parameters nMarkCtrl (siehe TcMarkOption [▶ 64]) übergeben.

Rückgabewert

Name	Typ	Beschreibung
LogMarkEx	HRESULT	Liefert im Fehlerfall den Fehlercode zurück.

Eingänge

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

Name	Typ	Beschreibung
stMark	ST_RTMon_MarkDef [▶ 63]	Übergabeparameter für eine definierte Marke, die geschrieben werden soll.
nMarkCtrl	UDINT	Definiert den Markentyp.

12.2 Datentypen

12.2.1 ST_RTMon_MarkDef

Datentyp, welcher eine Marke repräsentiert. Mithilfe dieses Datentyps ist es möglich, eine generische Marke (noch ohne deren Typ) zu definieren. Diese wird dann in der Methode LogMarkEx [▶ 63] des Funktionsbausteins FB_RTMon_LogMarkBase [▶ 62] zusätzlich zum Markentyp mit übergeben.

Syntax

```
TYPE ST_RTMon_MarkDef:
STRUCT
    nContextId : UINT;
    nGroupId : UINT;
    nEventId : UINT;
END_STRUCT
END_TYPE
```

Parameter

Wert	Beschreibung
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.

Sie können sowohl die ContextId als auch die GroupId im TwinCAT 3 Realtime Monitor mit Namen versehen. Sie können diese über die Funktionen Export User Context [▶ 45] bzw. Import User Context [▶ 45] exportieren bzw. importieren, sodass sie auch für die weitere Aufzeichnung zur Verfügung stehen.

12.3 Globale Konstanten

12.3.1 TcMarkOption

Die Konstanten in dieser globalen Variablenliste definieren die möglichen Markentypen (siehe Markentypen [▶ 20]). 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.

```
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
```

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, müssen Sie die Option **Show Task Reference** (siehe Markengruppen-Element [▶ 54]) einschalten.

13 C++ API

13.1 Datentypen

13.1.1 TcMark16

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

Syntax

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

Parameter

Wert	Beschreibung
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 Export User Context [▶ 45] bzw. Import User Context [▶ 45] exportiert bzw. importiert werden, so dass sie auch für die weitere Aufzeichnung zur Verfügung stehen.

13.2 Klassen

13.2.1 CTcLogMark

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

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

13.2.1.1 LogIntervalStart

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

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

Parameter

Wert	Beschreibung
GroupId	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
EventId	Optionale Event-ID.

13.2.1.2 LogIntervalStop

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

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

Parameter

Wert	Beschreibung
GroupId	Marken-ID (Markengruppe) für die die Marke geschrieben werden soll.
EventId	Optionale Event-ID.

13.2.1.3 LogMark

Die Methode erstellt eine Marke für die übergebene Markengruppen-ID. Der Markentyp wird anhand der Konstanten aus der TcLogMark.h (siehe Konstanten [▶ 67]) 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.

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

13.2.1.4 LogMarkEx

Die Methode erstellt eine Marke. Die Markendefinition erfolgt anhand des Datentyps TcMark16 [▶ 65]. Der Markentyp wird anhand der Konstanten aus der TcLogMark.h (siehe Konstanten [▶ 67]) übergeben.

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

13.2.1.5 LogSequenceStart

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

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

13.2.1.6 LogSequenceStop

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

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

13.2.1.7 LogStart

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.

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

13.2.1.8 LogStop

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.

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

13.2.1.9 SetContextId

Mit dieser Methode wird die verwendete Kontext-ID gesetzt.

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

13.2.1.10 InitLogMark

Initialisiert die Instanz der CTcLog-Mark-Klasse.

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

Parameter

Wert	Beschreibung
ipSrv	Interface Pointer zum TcObjectServer.

13.2.1.11 ReleaseLogMark

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

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

13.3 Konstanten

Diese Konstanten - definiert in der TcLogMark.h - definieren die möglichen Markentypen [▶ 20].

```
#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
```

14 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

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

Arm, Arm9 and Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

Mehr Informationen:
www.beckhoff.com/te1010

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