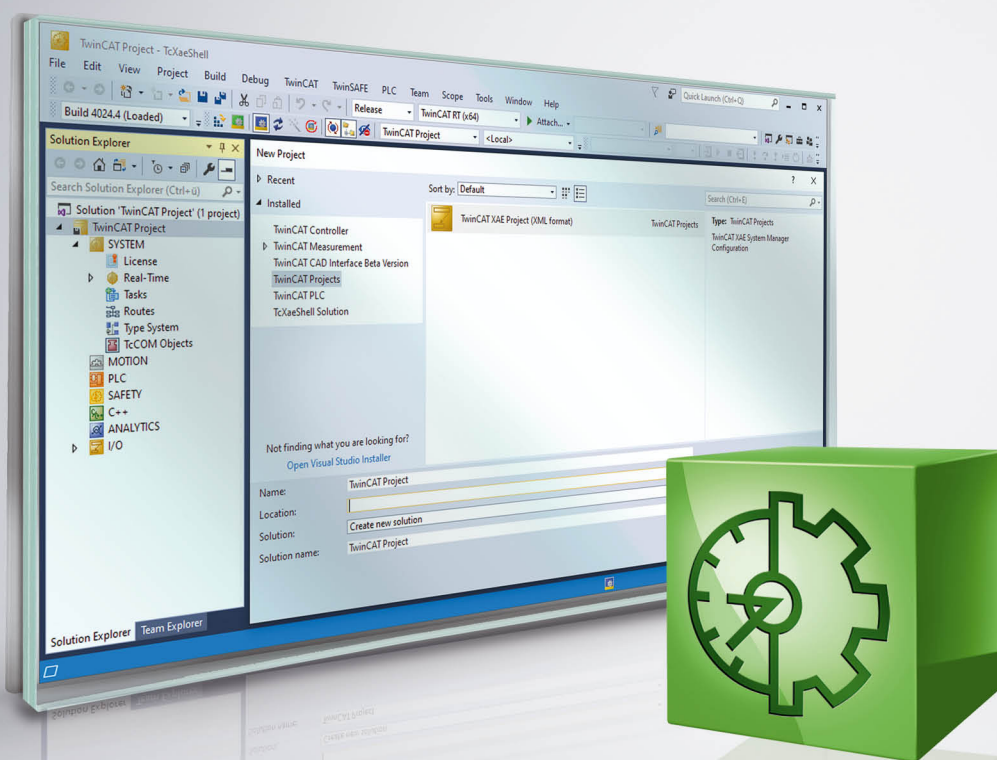


Handbuch | DE

TwinCAT 3

Korrigierte Zeitstempel



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht	8
3	Systemvoraussetzungen	9
4	Limitierungen	10
5	Technische Einführung	11
5.1	Consumer	11
5.1.1	TwinCAT Komponenten als Offset Consumer	11
5.1.2	Anwendungs-Implementierung	11
5.2	Provider	12
5.2.1	NTP Provider	12
5.2.2	DC Provider	16
5.2.3	Anwendungs-Implementierung	18
6	Realtime API	20
6.1	Strukturen	20
6.1.1	Enum TimeType	20
6.2	Schnittstellen	20
6.2.1	Schnittstelle ITcSetExternalTime	20
6.2.2	Schnittstelle ITcExternalTime	22
7	ADS API	25
8	Beispiele	27
8.1	ADS Consumer	27
8.2	PLC Consumer	27
8.3	C++ Consumer	28
8.4	C++ Provider	28
9	Anhang	30
9.1	FAQ	30
9.1.1	Windows als NTP Client	30
9.1.2	Windows als NTP Server	30

1 Vorwort

1.1 Hinweise zur Dokumentation

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

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

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

Disclaimer

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

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

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

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

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

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zu widerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

Erklärung der Symbole

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

GEFAHR

Akute Verletzungsgefahr!

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

WARNUNG

Verletzungsgefahr!

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

VORSICHT

Schädigung von Personen!

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

HINWEIS

Schädigung von Umwelt oder Geräten

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

Tipp oder Fingerzeig

i Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Hinweise zur Informationssicherheit

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

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

2 Übersicht

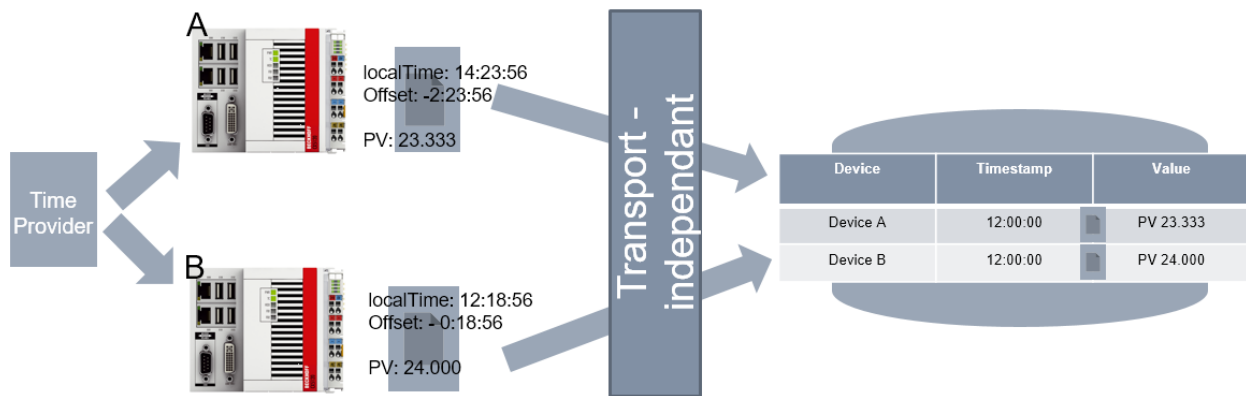
Steuerungen erzeugen Daten, welche in modernen, verteilten Anlagen gesammelt und in Verbindung gebracht werden sollen.

Da Steuerungen erstmal alleinstehende Geräte sind, besitzen diese auch unabhängige Zeitbasen. In einer gemeinsamen Datenbank würden Daten damit nicht in zeitliche Korrelation gebracht werden können.

Um diesem Problem zu begegnen ist es seit langer Zeit möglich Steuerungen untereinander zu synchronisieren, dazu dient beispielsweise das Netzwerkprotokoll IEEE 1588, bzw. PTP.

In vielen Szenarien reicht es jedoch aus, die Daten mit einem einheitlichen Zeitstempel zu versehen, die Steuerungen jedoch unabhängig voneinander zu betreiben, sodass zum einen die mit den oben genannten Protokollen verbundenen Hardwarekosten reduziert werden, zum anderen aber auch keine technische Abhängigkeit zwischen den Steuerungen existiert.

In diesem Kapitel werden deswegen die TwinCAT Komponenten zur Adaption von Zeitstempeln für das Speichern von zeitsynchronen Daten beschrieben.



Die Abbildung verdeutlicht die grundlegende Idee: Unabhängige Steuerungen beziehen den lokalen Zeitstempel und passen diesen mittels eines Offset an, welcher dann für die Speicherung der gemeinsamen Daten verwendet wird.

Hierfür steht in der TwinCAT Echtzeit eine zentrale Komponente bereit, das „External Time Interface“. Diese Komponente

- erhält von einer konfigurierten Quelle („External Time Provider“) den Offset zur korrigierten Uhrzeit.
- stellt in Abhängigkeit von der aktuellen, lokalen Zeit eine korrigierte Uhrzeit dem „External Time Consumer“ bereit.

Diese korrigierte Uhrzeit kann nun von unterschiedlichen Komponenten innerhalb und außerhalb der Echtzeit verwendet werden.

Als Quelle dient typischerweise entweder ein NTP Server oder ein DC-Zeitsignal basierend auf EtherCAT und z. B. synchronisiert mittels EL6688 durch PTP (IEEE1588). Eine Quelle kann aber auch vom Kunden implementiert werden, sodass andere Zeitsignale als Quelle realisiert werden können.

Neben der oben beschriebenen zentralen Komponente in der TwinCAT Echtzeit umfasst das Konzept also zwei Arten von Komponenten:

1. External Time Provider: Stellen einen Offset für die Anpassung von Zeitstempeln der zentralen Komponente bereit.
Ein Provider bezieht z. B. via NTP (Network Time Protocol, vgl. RFC 4330) einen Zeitstempel, woraus er im Vergleich zu der lokalen Systemzeit einen Offset berechnet und diesen bereitstellt.
2. External Time Consumer: Verwenden einen Offset, den sie aus der zentralen Komponente beziehen. Somit kann in den Komponenten ein Zeitstempel verwendet werden, der auf entfernten Geräten zu vergleichbaren Daten führt.
Als Consumer kommen dabei alle TwinCAT Komponenten, welche Zeitstempel verwenden in Frage, aber auch Kundenanwendungen.

3 Systemvoraussetzungen

Technische Daten	Voraussetzung
Betriebssystem	Windows 7/10, Windows Embedded Standard 7
Zielplattform	PC-Architektur (x86, x64)
Minimale TwinCAT-Version	TwinCAT 3.1 Build 4024.0 und höher
Erforderliches TwinCAT-Setup-Level	TwinCAT 3 XAE, XAR
Erforderliche TwinCAT-Lizenz	Beliebige Laufzeitlizenz (PLC, C++)

4 Limitierungen

Einige wichtige Einschränkungen sind zu berücksichtigen:

- Die TwinCAT System Zeit wird nicht verändert durch die hier beschriebene External Time Schnittstelle
- Die bereitgestellten External Time Offsets werden wie vom Provider bereitgestellt den Consumern bereitgestellt. Hieraus ergibt sich, dass
 - der Offset vom Provider korrekt berechnet werden muss.
 - keine Monotonie in den Zeitstempeln garantiert werden kann.
- Die External Time Offsets werden nicht gespeichert und nachträglich zum Abruf bereitgestellt. Es werden im TwinCAT System also nur die aktuellen Offsets verwaltet.

5 Technische Einführung

TwinCAT stellt sowohl auf Seite des External Time Provider wie auch des External Time Consumer unterschiedliche Schnittstellen bereit, um das Konzept der korrigierten Zeitstempel nutzbar zu machen.

Auf External Time Consumer-Seite sind unterschiedliche TwinCAT Komponenten in der Lage den externen Zeitstempel zu verwenden. Zusätzlich gibt es auch für Anwendungen unterschiedliche Zugriffsmöglichkeiten.

Auf External Time Provider-Seite werden Module bereitgestellt, die per NTP einen Offset berechnen und bereitstellen können. Zusätzlich gibt es ein Modul, welches per DC den Offset verwenden kann. Die entsprechende Schnittstelle zur Bereitstellung des Offsets wird für TwinCAT C++ ebenfalls angeboten, sodass eigene External Time Provider erstellt werden können.

Zeitstempel für unterschiedliche Anwendungsfälle

Wichtig ist hierbei, dass TwinCAT vier Typen von Zeitstempeln in diesem Konzept unterscheidet:

1. None: Lokale Systemzeit und keine Korrektur
2. Soft: Empfohlene Verwendung z. B. für NTP
3. Medium: Empfohlene Verwendung z. B. für IEEE1588
4. Hard: Empfohlene Verwendung z. B. für Hardware-Synchronisierung, bei der kein Drift entstehen sollte

Ein External Time Provider stellt einen der möglichen Offsets bereit, es lässt sich nur ein Provider pro Typ definieren.

Ein External Time Consumer kann dann beliebigen Offset verwenden, es lassen sich alle vier Typen von Offset individuell verwenden. So ist es in unterschiedlichen Kompletten oder Betriebsarten möglich, unterschiedliche Zeitstempel zu verwenden. Beispielsweise kann eine lokale Diagnose auch mit der lokalen Systemzeit erfolgen, während gleichzeitig etwa aggregierte Daten von unterschiedlichen Systemen mit dem Offset vom Typ „Soft“ korrigiert und in einer gemeinsamen Datenbank abgelegt werden.

Die Schnittstellen der korrigierten Zeitstempel verwenden Datentypen mit der Länge 8 Byte und werden vom 1.1.1601 in 100ns gezählt.

5.1 Consumer

External Time Consumer sind Komponenten, die die lokal gültige Systemzeit durch einen Offset korrigieren können. Hierfür müssen die Komponenten einen Offset vom Typ Soft, Medium oder Hard auswählen bzw. konfigurieren und entsprechend abfragen.

5.1.1 TwinCAT Komponenten als Offset Consumer

Die folgenden Komponenten von TwinCAT unterstützen den Ansatz der korrigierten Zeitstempel – die jeweilige Dokumentation beschreibt, wie diese Funktionalität aktiviert werden kann:

- TwinCAT 3 Eventlogger

Diese Liste wird erweitert.

5.1.2 Anwendungs-Implementierung

Anwendungen können die External Time Offsets in unterschiedlichen Komponenten verwenden:

- Echtzeit - PLC: Die PLC kann einen Offset abfragen bzw. einen lokalen Zeitstempel entsprechend korrigieren lassen.
- Echtzeit - C++: C++ TcCOM Module sind in der Lage den Offset abzufragen und entsprechend zu handeln.

- User-Mode – ADS Device Notifications: Die mit den ADS Device Notifications versendeten Zeitstempel können korrigiert sein.
- User-Mode – ADS Read: Der korrigierte Zeitstempel kann durch ein ADS Read abgerufen werden. Dieses kann in ADS Summenkommandos verwendet werden um einen Zeitstempel zusammen mit Daten abzurufen.

Unter den entsprechenden API Kapiteln sind die Schnittstellen dokumentiert.

5.2 Provider

External Time Provider sind Komponenten, die einen External Time Offset in Bezug zur lokalen Systemzeit durch eine externe Informationsquelle bestimmen und in TwinCAT bereitstellen. Hierdurch können External Time Consumer unabhängig vom Provider eine korrigierte Uhrzeit erhalten.

TwinCAT liefert Provider mit:

- NTP Provider: Eine Implementierung, die per (S)NTP ein Zeitsignal von einem NTP-Server abfragt und bereitstellt.
- DC Provider: Eine Implementierung, die die DC Zeit aus dem EtherCAT Master (z.B. per IEEE1588 oder PTP) als Offset weiterreicht an TwinCAT
- Zusätzlich ist der Kunde in der Lage eigene Provider bereitzustellen.

5.2.1 NTP Provider

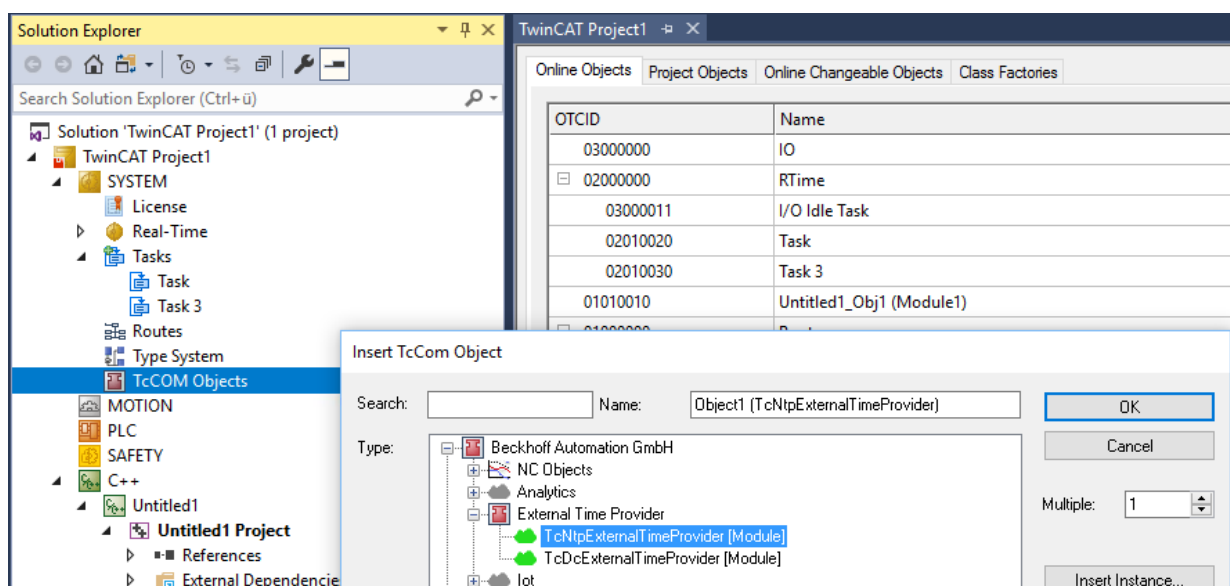
Der NTP Provider ist ein (S)NTP Client, der ein Zeitsignal von einem NTP Server zyklisch bezieht. Hierdurch kann er einen Offset der Systemzeit von dem Zeitsignal des NTP Servers berechnen und entsprechend bereitstellen.

Konfiguration

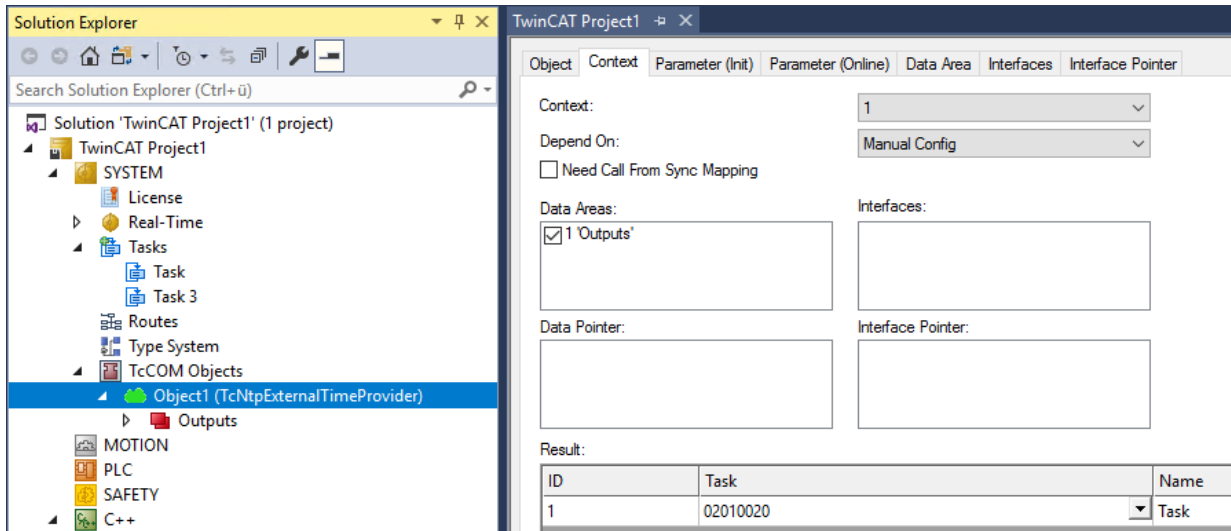
Der NTP Provider ist als TcCOM Modul „TcNtpExternalTimeProvider“ implementiert. Dieses Modul wird als TcCOM Modul folgendermaßen in Betrieb genommen:

✓ TwinCAT Projekt

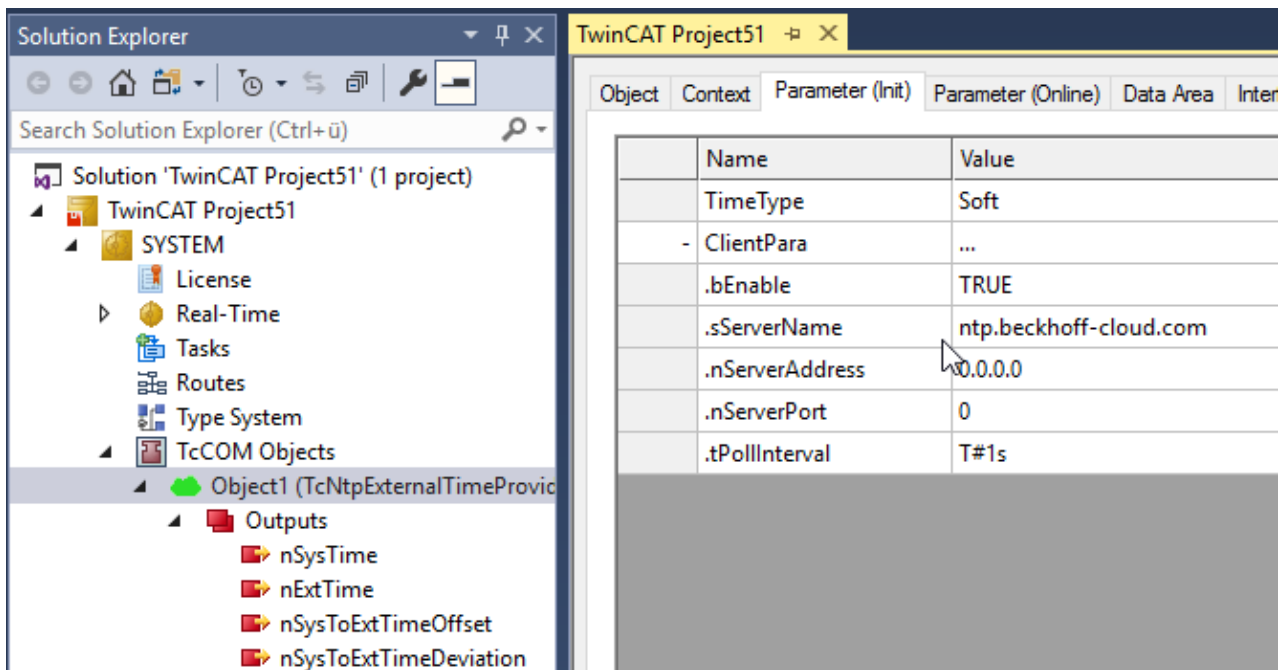
1. Einfügen eines TcCOM Moduls unter System->TcCOM Objects unter Auswahl des Typen TcNtpExternalTimeProvider in der Kategorie External Time Provider.



2. Das Modul benötigt eine Task, von dem es aufgerufen wird. Diese wird über den **Context**-Tab des Moduls parametrisiert:



⇒ Das TcCOM Modul kann parametrisiert werden:



Die Konfiguration erfolgt im Tab **Parameter (Init)**. Die Parameter bedeuten:

- **TimeTime**: Der Type des Offsets für den dieses Modul einen Offset ermitteln soll.

Client Para:

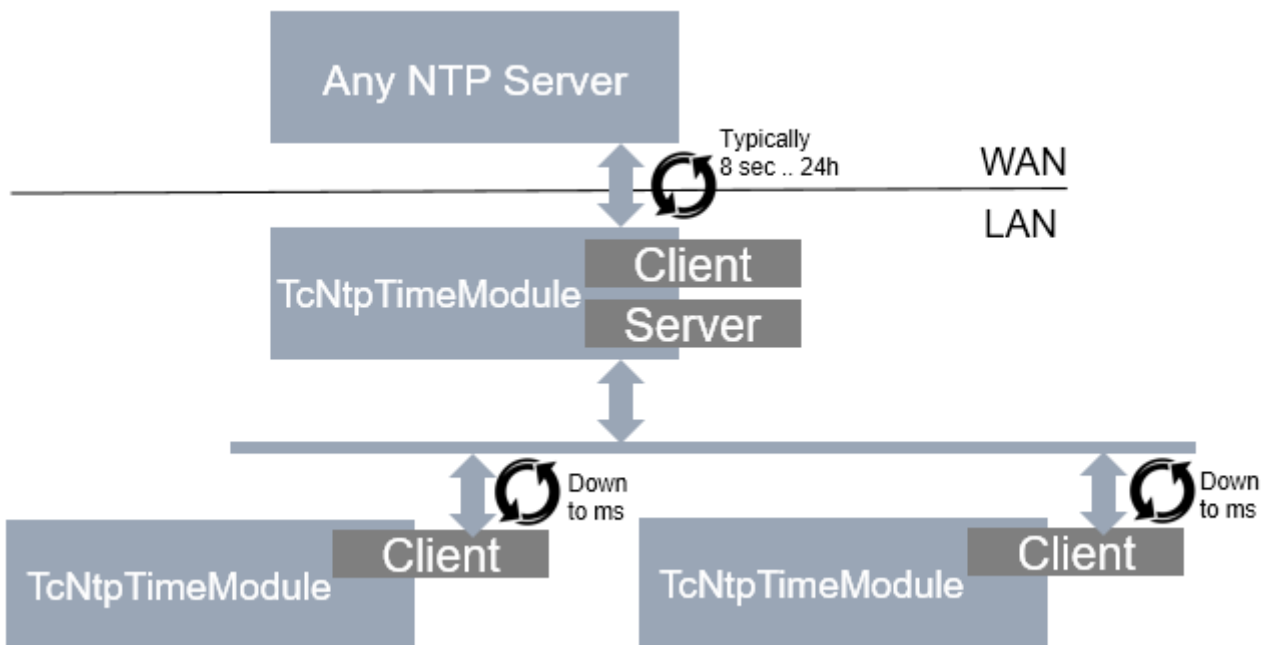
- **bEnable**: Zum Unterbinden der NTP Kommunikation kann das Modul deaktiviert werden.
- **sServerName**: Der Name des NTP Servers, der als Quelle genutzt werden soll.
- **nServerAddress**: Die IP Adresse eines NTP Servers (wird genutzt, falls sServerName leer ist).
- **nServerPort**: Der zu nutzende UDP Port des NTP Servers (default: 123).
- **tPollIntervall**: Das Intervall, in dem die NTP Abfragen gestartet werden sollen. Es wird dabei das vom Server angegebene Maximum berücksichtigt, sodass ggf. langsamere Anfragen entstehen.

Dieses Modul reicht einen ermittelten Offset zum einen über das `ITcSetExternalTime [► 20]` Interface zu TwinCAT. Zum anderen Stehen auch Ausgänge für das Mapping bereit.

NTP Provider als NTP Server

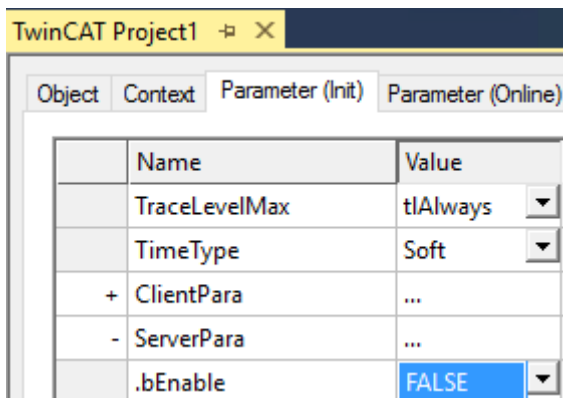
Optional kann das gleiche Modul auch als NTP Server fungieren. Somit kann ein Zeitsignal von einem externen NTP Server (als Client) bezogen werden und gleichzeitig an unterlagerte Systeme bereitgestellt werden.

Für den externen Server gilt nach NTP Protokoll typischerweise eine minimale Abfragezeit von 8 Sekunden oder mehr. Der NTP Provider als NTP Server lässt hingegen häufigere Abfrageintervalle zu.



Server Funktion

Die Server-Funktionalität ist normalerweise ausgeblendet. Sie lässt sich über **Show Hidden Parameter** einblenden und konfigurieren:



- **bEnable:** Einschalten der NTP-Server Funktionalität für dieses Modul. Dafür muss der Port udp/123 in der Windows Firewall entsprechend geöffnet werden.
- **nPort:** Der UDP Port, der genutzt wird, um den Server anzubieten (default: 123).

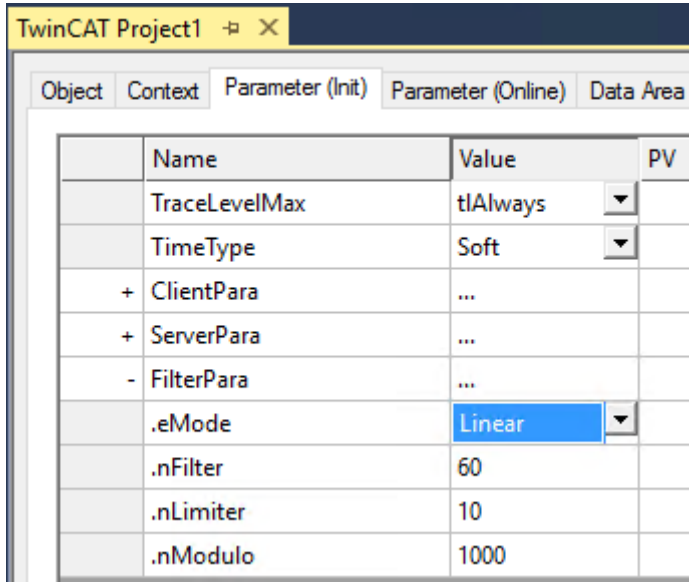
Die folgenden Parameter dienen dazu, die bereitgestellten NTP Informationen anzupassen. Per Default werden sie gesetzt wie im Protokoll vorgesehen, können hier jedoch überschrieben werden:

- **nLeap:** Manuelle Konfiguration des Leap Indicators.
- **nStratum:** Manuelle Konfiguration des Stratums.
- **nRoot:** Manuelle Konfiguration der Root-Server Information, wie sie in Abhängigkeit von Stratum definiert wird.

Filter-Funktion

Werden Offsets durch die NTP Server-Abfrage ermittelt, kann das Modul selbstständig einen Übergang des alten Offsets zum neuen Offset durchführen.

Diese Funktionalität ist normalerweise ausgeblendet. Sie lässt sich über **Show Hidden Parameter** einblenden und konfigurieren:



- **eMode:** Eine Auswahl an Modes. Aktuell werden entweder keine oder eine lineare Anpassung (Default) vorgenommen.

Bei der Auswahl von „Linear“ als eMode gelten folgende Parameter:

- **nFilter:** Anzahl der Werte über die gemittelt wird, also Anzahl der NTP Responses. Bei einem PollInterval von 1s würde nFilter = 60 ein Filter über eine Minute bewirken. (Default: 60).
- **nLimiter:** Der Offset wird maximal um diesen Wert pro Zyklus verändert. Wenn die Differenz der lokalen zur externen Uhr 100ms und die Zykluszeit 1ms wäre, würde beim nLimiter = 10 es somit 100.000 Zyklen oder 1,6 Minuten dauern bis der Offset eingeregelt wurde. (Default 1µs).
- **nModulo:** Rundung des Offsets. Üblicherweise sollte dieses in Abhängigkeit von der Zykluszeit gewählt werden. Der Offset wird über diesen Modulo angepasst, damit keine „unrunden“ Zeiten entstehen. Die DC Time wird Modulo der Zykluszeit zurückgegeben, korrigiert mit dem Offset bleibt der Zeitstempel somit „rund“. Dadurch verändert sich der Offset/Zeitstempel aber auch mit kleinen Sprüngen, wenn eine Anpassung stattfindet. Wie unter nLimiter beschrieben und mit nModulo = 1000 würde sich der Offset und somit der relative Zeitstempel jeden 100ten Zyklus um 0,1ms inkrementieren.

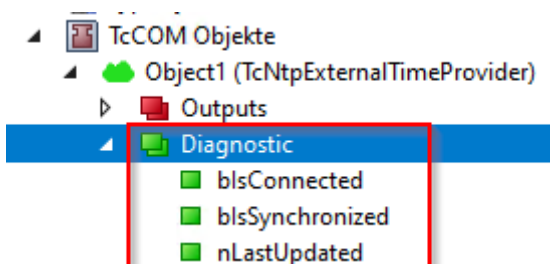
Diagnose

Unter dem Tab **Parameter (Online)** können entsprechende Diagnose-Informationen eingesehen werden.

Object	Context	Parameter (Init)	Parameter (Online)	Data Area	Interfaces	Interface Pointer
		Name	Online	CS	Unit	
-	ServerInfo		...	<input type="checkbox"/>		
		.nLeap	0			
		.nVersion	4			
		.nMode	4			
		.nStratum	2			
		.tPollIntv	T#1m4s		[ms]	
		.fPollPrec	1e-07		[s]	
		.fRootDelay	0.00047302968		[s]	
		.fRootDisp	0.0029449912		[s]	
		.sRefId	129.70.130.70			
		.nRefTime	2019-06-11T07:24:03.9653238			
		.nOrgTime	2019-06-11T07:24:05.1789999			
		.nRecTime	2019-06-11T07:24:05.4467752			
		.nTmtTime	2019-06-11T07:24:05.4468292			
		.nDstTime	2019-06-11T07:24:05.199			

Zu jeder Zeile befindet sich in der Spalte **Kommentar** eine entsprechende Beschreibung.

Zusätzlich sind zur programmatischen Auswertung entsprechende Symbole bereit:



5.2.2 DC Provider

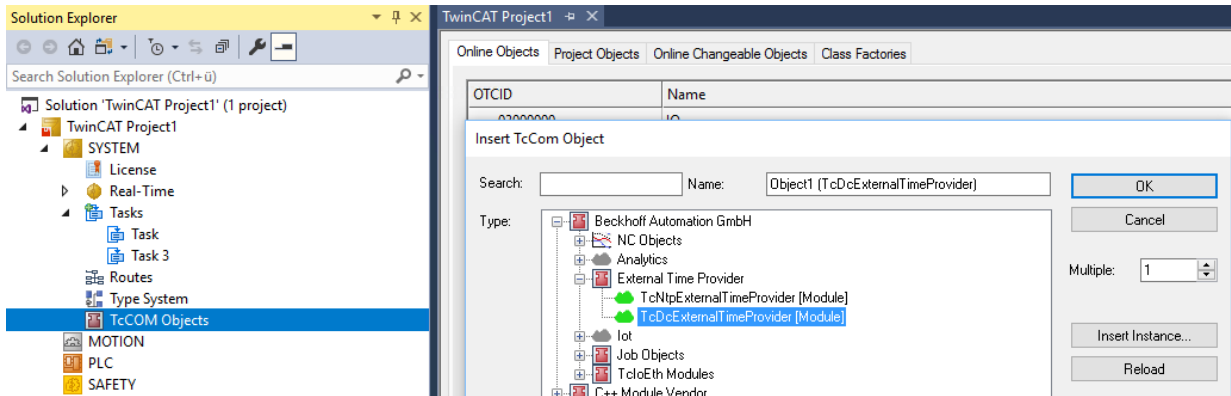
Der DC Provider bezieht ein Offset durch das Mapping von einem EtherCAT Master. Es kann verwendet werden um aus dem I/O Bereich Zeitwerte als Offset zu verwenden, wie sie beispielsweise durch den EtherCAT Master (DC Zeit) oder eine EL6695 bereitgestellt werden.

Konfiguration

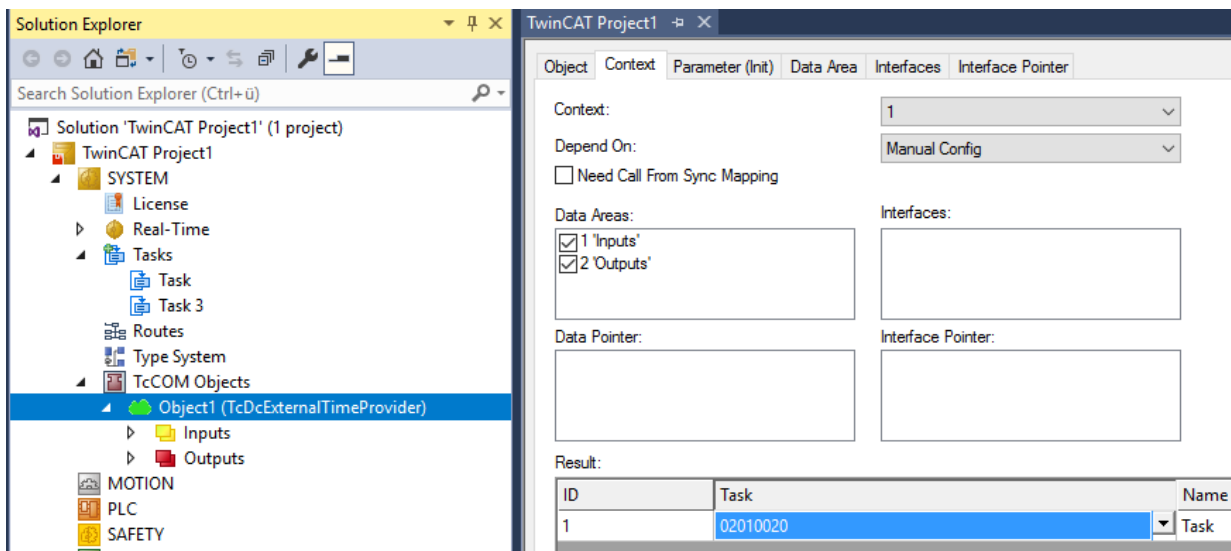
Der DC Provider ist als TcCOM Modul „TcDcExternalTimeProvider“ implementiert. Dieses Modul wird als TcCOM Modul folgendermaßen in Betrieb genommen:

- ✓ TwinCAT Projekt

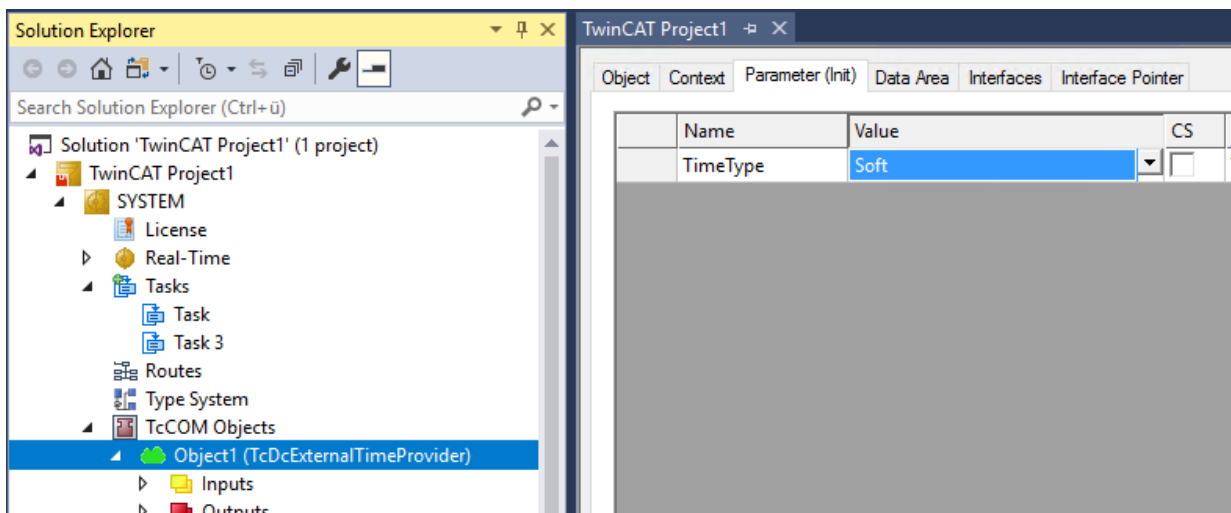
1. Einfügen eines TcCOM Moduls unter System->TcCOM Objects unter Auswahl des Typen TcDcExternalTimeProvider in der Kategorie External Time Provider.



2. Das Modul benötigt einen Task, von dem es aufgerufen wird. Dieser wird über den Context-Tab des Moduls parametrier:



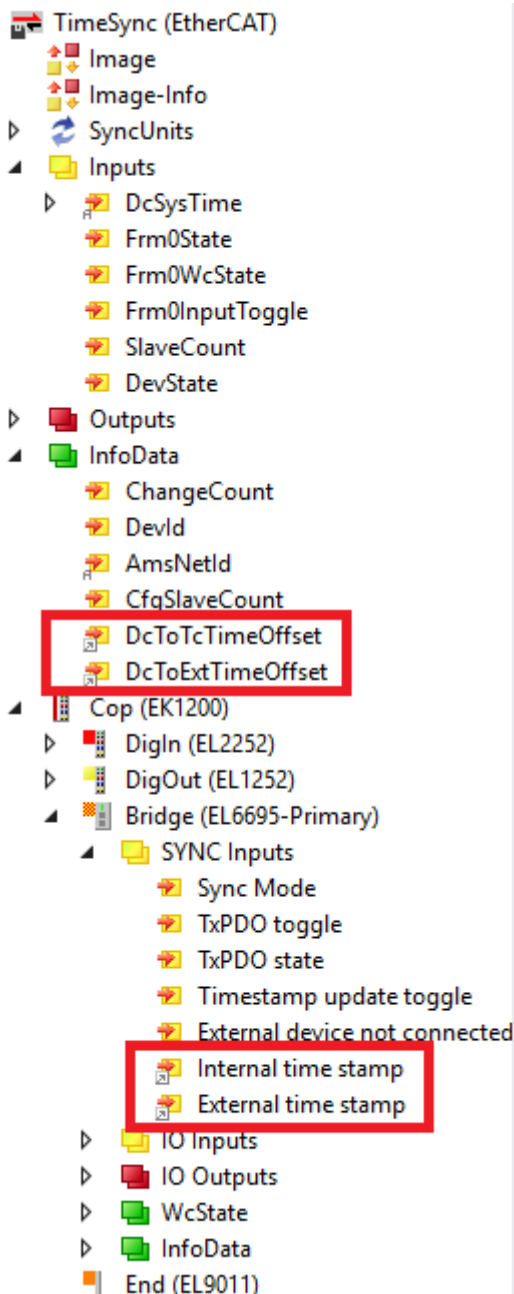
⇒ Das Modul kann parametrier werden:



Die Konfiguration erfolgt im Tab „Parameter (Init)“. Die Parameter bedeuten:

- TimeTime: Der Type des Offsets für den dieses Modul einen Offset ermitteln soll.

- Den Offset selber bezieht dieses Modul per Mapping:



Dieses Modul reicht einen ermittelten Offset zum einen über das `ITcSetExternalTime` [► 20] Interface zu TwinCAT. Zum anderen stehen auch Ausgänge für das Mapping bereit.

5.2.3 Anwendungs-Implementierung

Eine Anwendung kann eigene TimeOffset Provider bereitstellen, indem in TwinCAT C++ das Interface `ITcSetExternalTime` verwendet wird.

Dieses Modul stellt für die jeweiligen Offsets ggf. zyklisch einen Wert bereit.

Ablauf

Ein Modul realisiert den folgenden Ablauf

- ✓ Ein TcCOM Modul wurde instanziiert
- 1. Mittels `RegisterExternalTimeProvider` registriert sich das Modul als Provider eines bestimmten Typs von Offset (Soft/Medium/Hard)

2. Mittels SetExternalTimeOffset kann ggf. zyklisch ein Offset bereitgestellt werden
3. Mittels UnregisterExternalTimeProvider meldet sich das Modul entsprechen ab

Mittels der Registrierung wird sichergestellt, dass zu einem Zeitpunkt ein Offset von nur genau einem Modul genutzt werden kann.

Die genauere Beschreibung der Schnittstelle ITcSetExternalTime finden Sie im Kapitel [Schnittstelle ITcSetExternalTime](#) [▶ 20].

6 Realtime API

An dieser Stelle sind Schnittstellen und Strukturen dokumentiert um mit den korrigierten Zeitstempeln aus der Echtzeit umzugehen.

6.1 Strukturen

6.1.1 Enum TimeType

TwinCAT stellt vier unterschiedliche Zeitstempel bereit. Um diese zu unterscheiden wird das Enum TimeType verwendet.

Syntax

```
enum TimeType {
SystemTime = 0,
ExternalTimeHard = 1,
ExternalTimeMedium = 2,
ExternalTimeSoft = 3, // e.g. NTP
};
```

Werte

Die Verwendung der 3 externen Zeitstempel-Typen unterliegt der Anwendung; die unten beschriebene Verwendung ist lediglich ein Vorschlag.

Name	Beschreibung
ExternalTimeHard	Vorgeschlagene Verwendung bei harten Offsets, die keinen Drift haben
ExternalTimeMedium	Vorgeschlagene Verwendung bei genauen Offsets wie z.B. IEE1588
ExternalTimeSoft	Vorgeschlagene Verwendung bei allgemeinen Offsets, wie z.B. NTP

6.2 Schnittstellen

An dieser Stelle werden die Schnittstellen beschrieben, welche für die korrigierten Zeitstempel verwendet werden.

Zu den unterschiedlichen Zeitformaten und Darstellungen befindet sich im C++ SDK eine entsprechende Liste.

Siehe: Infosys [C/C++](#)

6.2.1 Schnittstelle ITcSetExternalTime

Die Schnittstelle ITcSetExternalTime wird vom TcCOM Object Server implementiert. Sie kann genutzt werden um einen extern ermittelten Offset bereitzustellen.

Syntax

```
TCOM_DECL_INTERFACE("00000067-0000-0000-e000-000000000064", ITcSetExternalTime)
struct __declspec(novtable) ITcSetExternalTime : public ITcExternalTime
```

Methoden

Name	Beschreibung
RegisterExternalTimeProvider [▶ 21]	Einen Provider für einen Offset in Bezug auf TimeType anmelden
UnregisterExternalTimeProvider [▶ 21]	Einen Provider für einen Offset in Bezug auf TimeType abmelden
SetExternalTimeOffset [▶ 22]	Für den angemeldeten TimeType einen neuen Offset bereitstellen

Anmerkungen

Dieses Interface steht nicht für die SPS zur Verfügung.

6.2.1.1 Methode RegisterExternalTimeProvider

Einen Provider für einen Offset in Bezug auf TimeType anmelden

Syntax

```
HRESULT TCOMAPI RegisterExternalTimeProvider(OTCID oidProvider, TimeType type) = 0;
```

Parameter

oidProvider: (Typ: OTCID) Die ObjektID des Providers; normalerweise die des Aufrufenden

type: (Typ: [TimeType](#) [[▶ 20](#)]) Der zu registrierende TimeOffset Typ.

Rückgabewert

Typ: HRESULT

Informiert über den Erfolg der Registrierung

Beschreibung

6.2.1.2 Methode UnregisterExternalTimeProvider

Einen Provider für einen Offset in Bezug auf TimeType abmelden

Syntax

```
HRESULT TCOMAPI UnregisterExternalTimeProvider(OTCID oidProvider, TimeType type) = 0;
```

Parameter

oidProvider: (Typ: OTCID) Die ObjektID des Providers; normalerweise die des Aufrufenden

type: (Typ: [TimeType](#) [[▶ 20](#)]) Der abzumeldende TimeOffset Typ.

Rückgabewert

Typ: HRESULT

Informiert über den Erfolg der Abmeldung

Beschreibung

6.2.1.3 Methode SetExternalTimeOffset

Für den angemeldeten TimeType einen neuen Offset bereitstellen

Syntax

```
HRESULT TCOMAPI SetExternalTimeOffset(OTCID oidProvider, TimeType type, __int64 offset) = 0;
```

Parameter

oidProvider: (Typ: OTCID) Die ObjektID des Providers; normalerweise die des Aufrufenden

type: (Typ: TimeType [▶ 20]) Der TimeOffset Typ

offset: (Typ: __int64) Der neu Offset-Wert.

Rückgabewert

Typ: HRESULT

Informiert über den Erfolg.

Beschreibung

Es gilt für den Offset ExternalTime = Internal Time + Offset. D.h. wenn die Zeit in TwinCAT in der Vergangenheit liegt, muss der Offset grösser 0 sein.

6.2.2 Schnittstelle ITcExternalTime

Die Schnittstelle ITcExternalTime wird vom TcCOM Object Server implementiert. Sie kann genutzt werden um einen extern ermittelten Offset abzurufen und zu verwenden.

Syntax

```
TCOM_DECL_INTERFACE("00000066-0000-0000-e000-000000000064", ITcExternalTime)
struct __declspec(novtable) ITcExternalTime : public ITcUnknown
```

 **Methoden**

Name	Beschreibung
SystemTimeToExternalTime [▶ 22]	Berechnung eines korrigierten Zeitstempels in Bezug auf die Systemzeit
ExternalTimeToSystemTime [▶ 23]	Berechnung der Systemzeit in Bezug auf einen korrigierten Zeitstempel
GetExternalTimeOffset [▶ 23]	Abruf eines Offsets in Bezug auf den TimeType
GetExternalTimeProvider [▶ 24]	Abfrage der ObjectID des aktuellen Providers

6.2.2.1 Methode SystemTimeToExternalTime

Berechnung eines korrigierten Zeitstempels in Bezug auf die Systemzeit

Syntax

```
HRESULT TCOMAPI SystemTimeToExternalTime(TimeType type, __int64& time) = 0;
```

Parameter

type: (Typ: [TimeType](#) [[▶ 20](#)]) Der zur Berechnung zu verwendende TimeOffset Typ

time: (Typ: `__int64&`) Der um den Offset zu korrigierende Zeitstempel

Rückgabewert

Typ: HRESULT

Informiert über den Erfolg.

Beschreibung**6.2.2.2 Methode ExternalTimeToSystemTime**

Berechnung der Systemzeit in Bezug auf einen korrigierten Zeitstempel

Syntax

```
HRESULT TCOMAPI ExternalTimeToSystemTime(TimeType type, __int64& time) = 0;
```

Parameter

Type: (Typ: [TimeType](#) [[▶ 20](#)]) Der zur Berechnung zu verwendende TimeOffset Typ

time: (Typ: `__int64&`) Der korrigierte Zeitstempel, welcher um den Offset bereinigt wird.

Rückgabewert

Typ: HRESULT

Informiert über den Erfolg.

Beschreibung

Es wird der zum Zeitpunkt des Aufrufes gültige Offset verwendet um die lokale Systemzeit zu ermitteln.

6.2.2.3 Methode GetExternalTimeOffset

Abruf eines Offsets in Bezug auf den TimeType

Syntax

```
HRESULT TCOMAPI GetExternalTimeOffset(TimeType type, __int64& offset) = 0;
```

Parameter

type: (Typ: [TimeType](#) [[▶ 20](#)]) Der abzurufende TimeOffset Typ

offset: (Typ: `__int64&`) Der Wert, welcher auf den Offset gesetzt wird.

Rückgabewert

Typ: HRESULT

Informiert über den Erfolg.

Beschreibung

6.2.2.4 Methode GetExternalTimeProvider

Abfrage der ObjectID des aktuellen Providers

Syntax

```
HRESULT TCOMAPI GetExternalTimeProvider(TimeType type, OTCID& oidProvider) = 0;
```

Parameter

type: (Typ: [TimeType](#) [[▶ 20](#)]) Der TimeOffset Typ, dessen Provider abgefragt werden soll.

oidProvider: (Typ: OTCID&) Die ObjectID, welche auf die ObjectID des Provider gesetzt wird.

Rückgabewert

Typ: HRESULT

Informiert über den Erfolg.

Beschreibung

7 ADS API

Die TimeOffsets können auch über ADS abgefragt werden. Hierfür gibt es zwei Wege

1. ADS Notification: ADS Notifications beinhalten einen Zeitstempel, der den Änderungszeitpunkt der Daten beinhaltet.
Ein ADS Client sendet dafür vor dem AddDeviceNotification ein ADS Kommando, wodurch das Zielsystem registriert, welcher Typ von korrigiertem Zeitstempel von diesem ADS Client gewünscht ist.
2. ADS Read: Per ADS Read kann ein korrigierter Zeitstempel ausgelesen werden. Dieses kann verwendet werden um in einem ADS Summen Kommando einen korrigierten Zeitstempel zu dem Zeitpunkt zu erhalten, an dem die ADS Kommandos ausgeführt wurden.

Index Group	Index Offset	Zugriff	Datentyp	Beschreibung	Anmerkung
ADSIGRP_EXT ERNALTIME 0xF088					
	ADSIOFFS_EX TERNALTIME_ SET 0x0000	R	LONG	Lesen des aktuell konfigurierten Offset-Typen für den jeweiligen ADS Client (AmsNetAddr inkl. Client-Port).	Rückgabewert ist Type 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ SET 0x00__	W		Setzen des Offset-Typen für die ADSDevice Notifications des jeweiligen ADS Clients (AmsNetAddr inkl. Client-Port).	__ ist Type 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ OFFSET 0x01__	R	LONGLONG	Lesen des aktuellen Offsets zu einem Typ.	__ ist Type: 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ OFFSET 0x01__	W	LONGLONG	Setzen des aktuellen Offsets zu einem Typ.	__ ist Type: 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ ABSOLUTE 0x02__	R	LONGLONG	Lesen des korrigierten Zeitstempels.	__ ist Type: 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ PROVIDER 0x03__	R	ULONG	Lesen der ObjektID von dem TimeOffset Provider.	__ ist Type: 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ SETALL 0x0400	R	LONG	Lesen des Typen, welcher verwendet wird, wenn kein anderer Typ gesetzt wurde.	Rückgabewert ist Type 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ SETALL 0x04__	W		Setzen des Typen, welcher verwendet wird, wenn kein anderer Typ gesetzt wurde.	__ ist Type 0 = None, 1 = Hard, 2 = Medium, 3 = Soft

Die Defines finden sich in der „Ads.h“ Datei.

Das [Beispiel ADS Consumer](#) [▶ 27] verdeutlicht die Verwendung.

8 Beispiele

Es werden verschiedene Beispiele zur Nutzung der Korrigierten Zeitstempel für den Anwender bereitgestellt:

- [PLC Consumer \[▶ 27\]](#): Ein PLC Programm greift auf korrigierte Zeitstempel zu.
- [C++ Consumer \[▶ 28\]](#): Ein C++ TcCOM Modul greift auf korrigierte Zeitstempel zu.
- [ADS Consumer \[▶ 27\]](#): Ein ADS Client im Usermode greift auf die korrigierten Zeitstempel zu.
- [C++ Provider \[▶ 28\]](#): Ein C++ TcCOM Modul ermittelt einen Offset und stellt diesen bereit.

Davon unabhängig werden die Korrigierten Zeitstempel von weiteren Komponenten des TwinCAT Systems genutzt. Eine nötige Konfiguration ist bei den jeweiligen Komponenten zu finden.

8.1 ADS Consumer

Das Beispiel ADS Consumer ruft korrigierte Zeitstempel ab, wie es in der [ADS API \[▶ 25\]](#) beschrieben ist.

Download

Hier erhalten Sie den https://infosys.beckhoff.com/content/1031/tc3_Korrigierte-Zeitstempel/Resources/zip/7705550603.zip für dieses Beispiel.

- ✓ Starten Sie das TwinCAT Zielsystem, mit dem das ADS Consumer Beispiel kommunizieren soll. Es kann das [PLC Consumer \[▶ 27\]](#) Beispiel verwendet werden.

1. Entpacken Sie die heruntergeladene ZIP-Datei.
2. Öffnen Sie die enthaltene vcxproj-Datei im Visual Studio.
3. Passen Sie die AmsNetID des Ziels an. (TcExternalTimeAdsClient.cpp, Zeile 119)

⇒ Das Beispiel ist einsatzbereit.

Beschreibung

Der Code des Beispiels befindet sich in der CPP Datei TcExternalTimeAdsClient.cpp

In der Main() Methode werden unterschiedliche UseCases für den Empfang von korrigierten Zeitstempel gezeigt:

- Lesen des Providers, des Offsets sowie des korrigierten Zeitstempel vom System Service für die unterschiedlichen Offsets unkorrigiert(0), soft(1), medium(2), hard(3) aber auch einen ungültigen Wert (4) um das Fehlverhalten darzustellen.
- Lesen der korrigierten Zeitstempel von einem PLC Programm wiederum für die unterschiedlichen Offsets.
- Lesen des verwendeten Providers und lesen aller Provider.
- Subscriben auf eine Variable in dem PLC, wobei die per Notification bereitgestellte Zeit einen korrigierten Zeitstempel aufweist. Die Ausgabe hiervon erfolgt in der Methode AdsNotificationCallback().


8.2 PLC Consumer

Das Beispiel PLC Consumer ruft einen korrigierten Zeitstempel aus dem TwinCAT-System ab und verwendet diesen.

Download

Hier erhalten Sie den https://infosys.beckhoff.com/content/1031/tc3_Korrigierte-Zeitstempel/Resources/zip/7705583115.zip für dieses Beispiel.

1. Öffnen Sie die enthaltene tszip-Datei in TwinCAT 3 mit einem Klick auf **Open Project**

2. Wählen Sie Ihr Zielsystem aus.
3. Bauen Sie das Beispiel auf Ihrer lokalen Maschine (z. B. **Build->Build Solution**).
4. Aktivieren Sie die Konfiguration mit einem Klick auf  .
⇒ Das Beispiel ist einsatzbereit.

Beschreibung

Unterhalb von **System >TcCOM Objects** ist der TcNtpExternalTimeProvider konfiguriert. Hier kann unter **Parameter (Init)** ein eigener NTP-Server parametrierbar werden, wenn der voreingestellte pool.ntp.org nicht erreichbar ist.

Das PLC Programm besteht im Wesentlichen aus dem Funktionsbaustein FB_TcExternalTime. Dieser stellt Funktionen bereit, um einen korrigierten Zeitstempel aus dem TwinCAT System auszulesen. Die Variable `_eTimeType` stellt dabei den Typen (Soft, Medium, Hard) dar und kann parametrierbar werden.


In der MAIN wird dieser Funktionsbaustein für den eTimeType „Soft“ verwendet, um die per NTP gesetzte korrigierte Zeit zu verwenden.

8.3 C++ Consumer

Das Beispiel C++ Consumer ruft einen korrigierten Zeitstempel aus dem TwinCAT-System ab und verwendet diesen.

Download

Hier erhalten Sie den https://infosys.beckhoff.com/content/1031/tc3_Korrigierte-Zeitstempel/Resources/zip/7705552907.zip für dieses Beispiel.

1. Öffnen Sie die enthaltene zip-Datei in TwinCAT 3 mit einem Klick auf **Open Project**
2. Wählen Sie Ihr Zielsystem aus.
3. Bauen Sie das Beispiel auf Ihrer lokalen Maschine (z. B. **Build->Build Solution**).
4. Aktivieren Sie die Konfiguration mit einem Klick auf  .
⇒ Das Beispiel ist einsatzbereit.

Beschreibung

Unterhalb von **System >TcCOM Objects** ist der TcNtpExternalTimeProvider konfiguriert.

Hier kann unter **Parameter (Init)** ein eigener NTP-Server parametrierbar werden, wenn der voreingestellte pool.ntp.org nicht erreichbar ist.

Das C++ Modul ermittelt in der CycleUpdate() Methode zyklisch einen lokalen Zeitstempel und lässt diesen korrigieren, was mittels Debugger in den jeweiligen Schritten verfolgt werden kann. Der korrigierte Zeitstempel wird als Parameter (Online) bereitgestellt.


Der dafür notwendige Typ wird als Parameter „TimeType“ am TcCOM Objekt konfigurierbar.

8.4 C++ Provider

Das Beispiel C++ Provider ermittelt einen Offset und legt diesen im TwinCAT-System ab, sodass er von den Consumern verwendet werden kann.

Download

Hier erhalten Sie den https://infosys.beckhoff.com/content/1031/tc3_Korrigierte-Zeitstempel/Resources/zip/770555211.zip für dieses Beispiel.

1. Entpacken Sie die heruntergeladene .zip-Datei.
2. Öffnen Sie die enthaltene .zip-Datei in TwinCAT 3 mit einem Klick auf **Open Project**
3. Wählen Sie Ihr Zielsystem aus.
4. Bauen Sie das Beispiel auf Ihrer lokalen Maschine (z. B. **Build->Build Solution**).
5. Aktivieren Sie die Konfiguration mit einem Klick auf  .
⇒ Das Beispiel ist einsatzbereit.

Beschreibung

Dier Offset Provider bekommt den bereitzustellenden Offset als DataArea „ExternalTime.nOffset“. Dieser wird dem TwinCAT-System als TimeType Medium übergeben was unter **Parameter (Init)** auch zur Laufzeit konfiguriert werden kann.

In der CycleUpdate() Methode wird hierfür die Methode SetExternalTimeOffset verwendet, nachdem ein entsprechendes Register mittels RegisterExternalTimeProvider für einen TimeType erfolgt ist.

9 Anhang

9.1 FAQ

9.1.1 Windows als NTP Client

Windows selber bietet einen NTP Client für die Systemzeit. Zusätzlich kann eine NTP Zeit auch mittels folgendem Skript abgeholt werden, welches für Debugging-Zwecke sinnvoll ist:

```
@echo off
set /p Server=Server:
w32tm /stripchart /computer:%Server% /packetinfo /samples:10
pause
```

9.1.2 Windows als NTP Server

Windows selber bietet einen NTP Server an um Zeitstempel bereitzustellen.

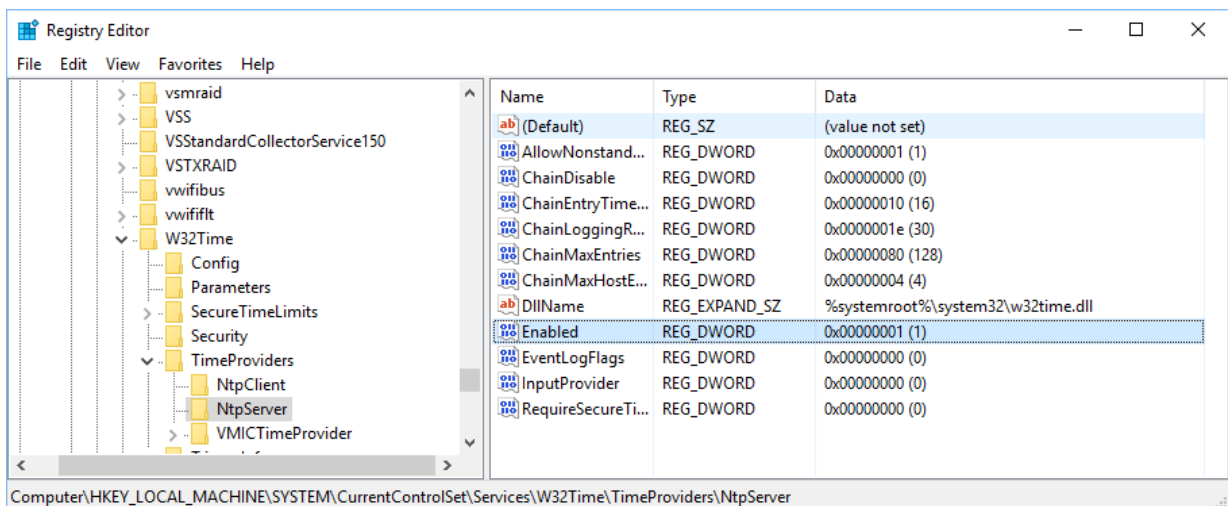
Bitte beachten Sie hierbei, dass nur eine Komponente den Port für NTP (udp/123) nutzen kann. Es kann also entweder die [TwinCAT NTP Server Funktionalität \[► 12\]](#) verwendet werden oder der Windows NTP Server.

Der Windows NTP Server ist standardmäßig ausgeschaltet und kann nachträglich aktiviert werden:

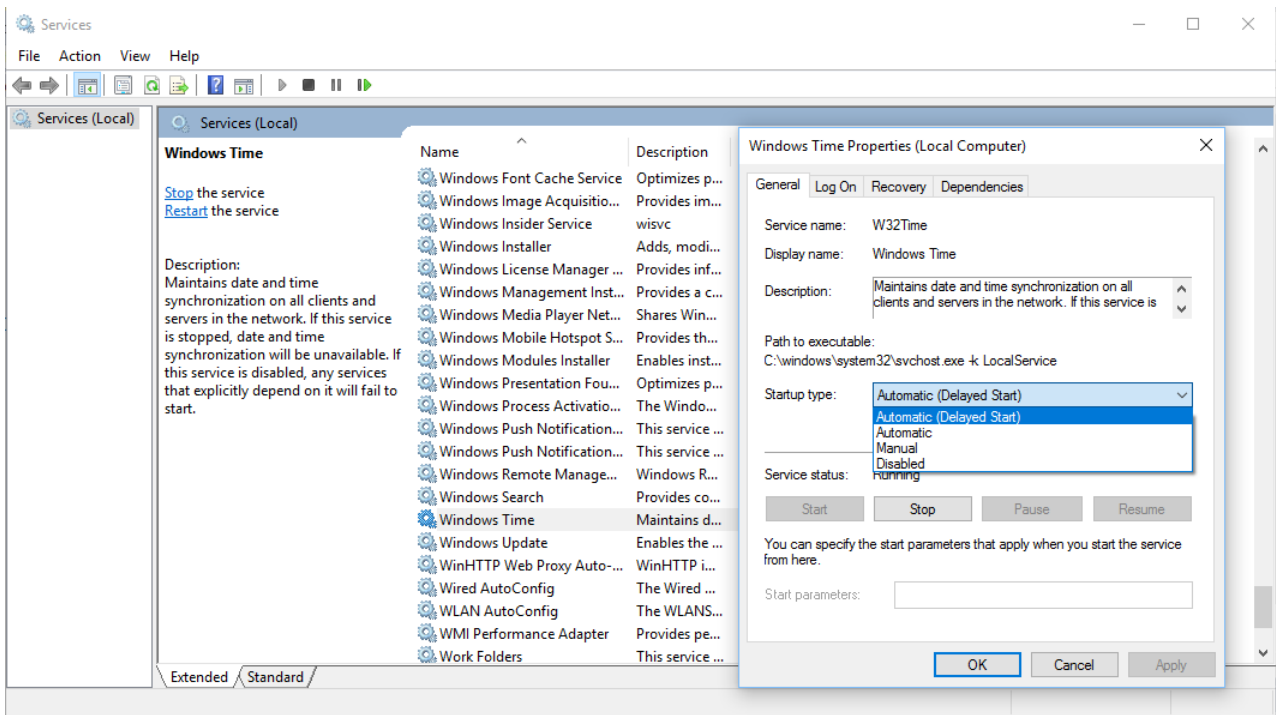
✓ Windows 7 / 10

1. Der Registry Key wird gesetzt:

```
HKLM\System\CurrentControlSet\Services\W32Time\TimeProviders\NtpServer
Enabled = 1
```



2. Und der System Dienst "Windows Time" wird gestartet, ggf auf Autostart gestellt.



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