

TwinCAT 3.1 Build 4024: noch mehr Effizienz und Durchgängigkeit im Engineering

Neue TwinCAT-Eigenschaften reduzieren die Engineeringkosten und steigern die Kommunikationssicherheit

Alle zwei Jahre bringt Beckhoff ein neues Build von TwinCAT auf den Markt. Diese Builds zeichnen sich dadurch aus, dass viele neue Eigenschaften enthalten sind, die das Arbeiten mit TwinCAT noch effizienter und schneller gestalten. So ergibt auch das aktuelle Build 4024 erneut eine Steigerung der Effizienz und Durchgängigkeit im Engineering, was zur Kostenreduzierung und durch Secure ADS auch zu einer höheren Kommunikationssicherheit führt. Weiterhin werden die neuesten Visual-Studio®-Versionen integriert.

Im neuen TwinCAT 3.1 Build 4024 wurde die Visual Studio® 2013 Shell gegen die modernere und verbesserte Visual Studio® 2017 Shell getauscht. Die Startseite von Visual Studio® enthält jetzt für die TwinCAT-Nutzer zielgerichtete aktuelle Informationen. Diese werden über einen – auch separat abonnierbaren – RSS-Feed in Visual Studio® eingespielt.

Multi-User-Zugriff auf die SPS

Inbetriebnahmen werden immer teurer. Zudem lässt sich häufig das entsprechende Personal nur schwer für eine aufwendige Inbetriebnahme gewinnen. Entsprechend naheliegend ist der Wunsch nach einer möglichst schnellen und einfachen Inbetriebnahme auch von komplexen Maschinen oder Anlagen. Dies ist nur dann möglich, wenn parallel mit mehreren Programmierern gearbeitet werden kann. Mit der neuen Multi-User-Fähigkeit bietet TwinCAT jetzt die genau passende Lösung hierfür. Beckhoff hat sich für diese neue Funktion eines Prinzips bedient, welches auch schon für die Offline-Entwicklung in Teams verwendet wird. Offline ist klar, dass man eine Softwareentwicklung in einem Team nur mithilfe eines Source-Code-Kontrollsystems realisieren kann. Diese Möglichkeit ist nun in TwinCAT ebenfalls vorhanden und hier auch für den Online-Betrieb so umgesetzt, dass kein zusätzliches Know-how vorhanden sein muss.

Und wie gestaltet sich das Arbeiten im Team online auf einer Steuerung? Angenommen, dass Programmierer 1 sich auf der Steuerung einloggt und eine Änderung am Code vornimmt. Parallel sind noch Programmierer 2 und 3 eingeloggt. Da Programmierer 2 und 3 den Quellcode der Änderung nicht auf ihren

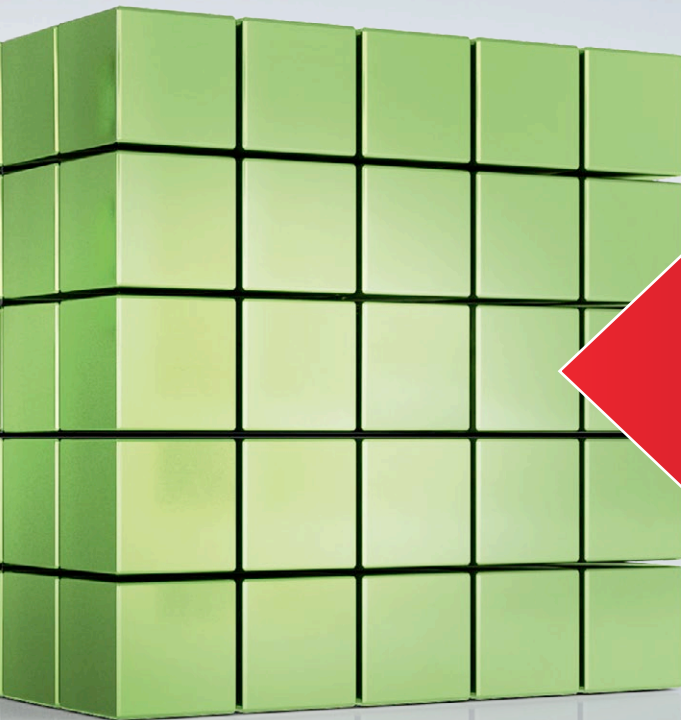
Programmiergeräten haben, muss das Entwicklungssystem den Quellcode von Programmierer 1 so ablegen, dass auch Programmierer 2 und 3 darauf Zugriff haben. TwinCAT nutzt dazu Git – die häufig verwendete Open-Source-Lösung für Sourcecodecontrol – mit einem Repository, das z. B. direkt auf der Steuerung liegen kann. Sobald Programmierer 2 sich einloggt, wird ein Fenster angezeigt, das die Änderungen gegenüber dem Source-Code-Stand auf dem Entwicklungsrechner von Programmierer 2 darstellt. Programmierer 2 kann dann die Änderungen von Programmierer 1 übernehmen oder auch beide Stände zusammenführen. Dazu dient das bekannte TwinCAT Comparetool. Programmierer 2 erkennt genau die Unterschiede zwischen den Quellcodeständen und muss dann entscheiden, welcher Stand übernommen werden soll. Ein weiterer Vorteil der Verwendung von Git auf der Steuerung liegt darin, dass sich zu jeder Zeit ablesen lässt, wer welche Änderungen durchgeführt hat. Damit ist eine lückenlose Nachverfolgung während der Inbetriebnahme oder auch während des gesamten Lebenszyklus einer Maschine gegeben.

Variantenmanagement vermeidet Fehler

Sollen mehrere Varianten einer Maschine in Software abgebildet werden, wurden bislang entweder unterschiedliche Source-Code-Stände für verschiedene Varianten verwendet oder es wurde eine Meta-Software entwickelt, die alle Varianten abdeckt. Über eine Parametrierung wurden dann die Spezifika einer Variante erzeugt. Beides ist umständlich und fehlerbehaftet:

- Unterschiedliche Software lässt sich nur schwer warten. Außerdem müssen Fehlerbehebungen in allen Softwarevarianten durchgeführt werden.





- Multi-User-Zugriff auf die SPS
- Zeitsynchronisierte Datensätze in verteilten Anlagen
- Sichere Kommunikation mit Secure ADS

TwinCAT® 3

TwinCAT® 3 **BECKHOFF**

Recent
Today

- TwinCAT Projekt7.sln

This week

- TwinCAT Projekt6.sln
- TwinCAT Projekt5.sln
- TwinCAT Projekt5.tsproj
- TwinCAT Projekt4.sln
- Tc3_CM_Test.tsproj
- TwinCAT Projekt3.sln

This month

- TwinCAT Projekt2.sln
- TwinCAT Projekt1.sln
- TwinCAT Project5.sln
- TwinCAT Project6.sln

Open
Open Project/ Solution

New project

- New TwinCAT Project...
- New Measurement Project...

Search project templates

- TwinCAT XAE Projekt (XML Format) TwinCAT Projekte
- TwinCAT PLC Project TwinCAT PLC

Create new project...

News

- New Version of TF6720 TC3 TwinCAT IoT Data Agent**
TF6720 has been updated to version 1.1.38.0. The new version is available for download. An optimization has been made to the core application for XAR and ADS installations. Please consider updating. Download 17 Jun 2019 12:55:00
- New Version of TF6720 TC3 TwinCAT IoT Data Agent**
TF6720 has been updated to version 1.1.37.0. The new version is available for download. Several optimizations have been made to the core, configurator and systray applications. Please consider updating. Download 13 Jun 2019 15:59:00
- New version of TF1810: TC3 PLC HMI Web**
TC3 PLC HMI Web has been updated to 1.7.3.0. The new version is available for download. Please consider updating, especially if you are using Build 4020. Download 11 Jun 2019 09:52:00
- New Version of TE1300/TF3300: TC3 Scope**
TC3 Scope has been updated to version 3.3.3140.0. The new version is available for download. Download 07 Jun 2019 09:19:00
- New Version of TF6421 TC3 XML Server**
TF6421 has been updated to version 3.1.23.0. The new version is available for download. Download 06 Jun 2019 12:07:00
- New Version of TF3650 TC3 Power Monitoring**
TF3650 has been updated to version 3.1.29.1. The new version is available for download. Download 05 Jun 2019 10:38:00
- New Version of TF6420: TC3 Database Server**
TC3 Database Server was updated to 3.2.32.1. The new version is available for download. Download 04 Jun 2019 10:23:00
- New Version of TF3600 TC3 Condition Monitoring Level 1**
TF3600 has been updated to version 3.1.18.0. The new version is available for download. Please consider updating. Download 03 Jun 2019 09:11:00

TwinCAT-Anwender erhalten nun über die Visual-Studio®-Startseite zielgerichtete aktuelle Informationen.

- Die Meta-Software hat den Nachteil, dass unnötig viel Code zu sehen ist und die Konfiguration natürlich den maximalen Ausbau zeigt. Das meiste davon wird in einer bestimmten Variante aber gar nicht genutzt. Auch hier ist die Wartbarkeit der Software schwierig.

Die optimale Lösung stellt hingegen eine Software dar, die über eine einfache Einstellung in verschiedenen Varianten generiert wird. In der klassischen Hochsprachenprogrammierung nutzt man dazu die bedingte Kompilierung. In einem TwinCAT-Projekt gibt es aber neben der SPS-Programmierung auch Varianten mit unterschiedlicher I/O-Ausstattung. Dementsprechend ist ein Teilnehmer nicht immer vorhanden oder er hat jeweils unterschiedliche Parameter. Gleiches gilt im Bereich Motion Control für eine Bewegungsachse. Aus diesem Grund stehen im neuen TwinCAT Build die sogenannten Projektvarianten sowie Gruppen davon für das gesamte TwinCAT-Projekt zur Verfügung. Auf diese Weise können zusätzlich variantenspezifische Einstellungen in Bereichen wie z. B. der I/O- oder Motion-Konfiguration vorgenommen werden. Die Pflege der Software ist nun enorm erleichtert. Es gibt nur noch eine Software für alle Maschinenvarianten. Beim Erzeugen einer spezifischen Variante muss lediglich die gewünschte Variante ausgewählt werden und schon wird automatisch die korrekte Software erzeugt. Versionierung und Wartung der Software sind somit sehr einfach.

Verbesserungen im SPS-Bereich

Im Bereich der SPS wurden zahlreiche Verbesserungen eingepflegt, um noch schneller und effektiver SPS-Code entwickeln und testen zu können. Gerade bei der objektorientierten Programmierung ist das Ziel eine bessere und einfachere Wartbarkeit und Wiederverwendung von Code. Die Nutzung von Interfaces ist hierbei eine Grundeigenschaft. Bisher konnte bei der Verwendung von Interface-Pointern zwar die Adresse angezeigt werden, eine Pointer-Dereferenzierung wurde aber nicht durchgeführt. Das ist jetzt mit Build 4024 möglich und führt ebenfalls zu einem verbesserten Engineering.

In den Hochsprachen C++ und C# ist die Programmierung von abstrakten Klassen mit abstrakten Methoden und Eigenschaften weit verbreitet. Diese Möglichkeit steht jetzt auch dem SPS-Programmierer zur Verfügung: Mit abstrakten Klassen lassen sich einfach Muster für Bausteine oder Klassen

erzeugen. Diese Muster können dann beim Ableiten der abstrakten Klasse mit Code gefüllt werden. Damit nähern sich die SPS-Programmierer wieder einen Schritt an die Möglichkeiten von C++ und C# an. Ähnlich verhält es sich mit der Ausnahmebehandlung. Kommt es zu einer Ausnahme (Exception), möchte auch der SPS-Programmierer darauf reagieren. Im Fall einer Division durch Null kann es sinnvoll sein, eine bestimmte Routine auszuführen, um die Maschine in einen sicheren Zustand zu bringen und so Maschinenschäden zu verhindern.

Der Mapping-Dialog wurde dahingehend optimiert, dass nicht alle Knotenebenen aufgeklappt sind. Dies führt zu einem kleineren Scroll-Bereich und damit zu einer besseren Übersicht während des Anlegens einer Verknüpfung, z. B. von der SPS zum I/O. So können die gesuchten Teilnehmer in diesem Dialog einfacher gefunden werden. Eine Option ermöglicht das Umschalten zwischen der alten, der neuen und einer automatisch angepassten Ansicht. Mit dem „Go To Link Variable“ konnte der Nutzer schon immer einfach von einem I/O-Teilnehmer zum Prozessabbild und zurück springen. Bisher gab es aber keine Möglichkeit, vom Prozessabbild in den Code zu wechseln. Das ist jetzt mithilfe des Befehls „Go To Definition“ möglich, was die Fehlersuche erleichtert. Zudem wird das Programm übersichtlicher, da mit dieser Funktionalität auf die noch häufig verwendeten globalen Variablenlisten verzichtet werden kann.

Ein weiterer Vorteil liegt darin, dass der Programmcode von Bausteinen in grafischen Sprachen jetzt auch im Binärformat Base64 gespeichert werden kann. Auf diese Weise lässt sich das Laden und Speichern von Programmen mit einem hohen Anteil an grafischen Sprachen beschleunigen.

Online-Change von C++- oder MATLAB®/Simulink®-Code

Bisher war es nur dem SPS-Programmierer möglich, im laufenden Betrieb der Maschine den Steuerungscode zu ändern. Da im praktischen Einsatz viele Maschinen oder Anlagen sich entweder nicht oder nur selten anhalten lassen, ist das auch ein absolut notwendiges Feature. Wird hingegen C++ oder MATLAB®/Simulink® für die Maschinensteuerung verwendet, konnte man diesen Code bisher nur in Verbindung mit einem Maschinenneustart ändern. Ein entsprechender Neustart ist allerdings in manchen Applikationen nicht zulässig. So dürfen insbesondere im Bereich von Prüfmaschinen Prozesse in der Regel nicht unterbrochen werden. Trotzdem sind immer wieder Programmänderungen erforderlich.

Als Lösung für diese Anforderung hat Beckhoff im neuen TwinCAT Build die Möglichkeit implementiert, auch C++- und MATLAB®/Simulink®-Code im laufenden Betrieb auszutauschen. Hierzu wird der entsprechende Code speziell versioniert. Im laufenden Betrieb kann dann eine Version durch eine neue Version unter Erhalt der Daten ersetzt werden.

Zeitsynchronisierte Datensätze in verteilten Anlagen

In größeren Anlagen mit mehreren Steuerungen sind häufig bestimmte Daten auf einem zentralen Server zu speichern, um diese aggregierten Daten anschließend auswerten zu können. Die Schwierigkeit dabei ist, dass die Zeitstempel auf den unterschiedlichen Geräten sich nicht einheitlich generieren lassen. Werden dann Daten für eine zentrale Anwendung wie z. B. TwinCAT Scope aggregiert, ist auf dem zentralen System kein genauer zeitlicher Bezug möglich. Die Systemzeiten auf den einzelnen Steuerungen können dabei für eine Synchronisierung nicht ohne weiteres nachgestellt werden, da dies die gesamte Anwendung beeinträchtigen würde. Es bleibt nur die Möglichkeit, mithilfe einer Referenzuhr die Offsets zu den lokalen Uhren zu ermitteln, wenn Zeitstempel für zu aggregierende Daten erzeugt werden.

Commit	Date	User	Comment
48925270f9e776a3c059f09b30c102779db53367	1/14/2019 12:57:29 PM	Zweitbenutzer	UpdateRemote 2 modified OnlineChange: - verteil set to 92
4c6d06c74b12738a621577934627689c7d08444e	1/14/2019 1:26:17 AM	Zweitbenutzer	UpdateRemote 1 modified OnlineChange: - verteil set to 5
00000000000000000000000000000000	1/14/2019 12:59:59 PM		Local changes
66646d0e1511c047d09232485495459e59e5508	1/15/2019 3:15:30 PM	Zweitbenutzer	UpdateRemote 1 modified OnlineChange
473f418069a3268114d89844827072a9909b3f8	1/15/2019 3:14:58 PM	Zweitbenutzer	UpdateRemote 1 modified OnlineChange: : 14
6b5029ba3a4e07e7a797933a2c0d849094	1/15/2019 3:14:08 PM	Zweitbenutzer	UpdateRemote 1 modified OnlineChange: : 12
738810c945e46f623a0a3678405d7196b4480a	1/15/2019 3:11:51 AM	Zweitbenutzer	UpdateRemote 1 added 2 modified OnlineChange: : 555
789e765e2a9f9b1d019234e009f8a6671950e	1/15/2019 8:43:37 AM	Zweitbenutzer	UpdateRemote OnlineChange: : merged nit var to 44 instead of 22
6e9721c1844a239e4f63e49422169d1103c	1/15/2019 8:41:28 AM	Zweitbenutzer	UpdateRemote 3 modified Save changes before commit
09e177296b207f680c7e7a12bde095b20624b	1/15/2019 8:39:45 AM	ThomasZ	UpdateRemote 1 modified OnlineChange: : nit var changed to 22 212
34eef9d39fa7e330a55923a1677a6205886c	1/15/2019 8:22:14 AM	Zweitbenutzer	UpdateRemote 1 added 2 modified OnlineChange: : prg1 added
918173b690230e53d642140d478eece0a	1/15/2019 4:59:04 PM	Zweitbenutzer	UpdateRemote OnlineChange: : merge from different states
d06e72050464ad4e83873122579178e36e97	1/15/2019 2:57:50 PM	Zweitbenutzer	UpdateRemote 1 modified OnlineChange: : >> PDU1 added, var nit 102 merged
c579799f5e46b2962008318764eb428f9e2	1/15/2019 11:18:28 AM	ThomasZ	UpdateRemote 1 modified Save changes before commit
be8513087252195690229a2c463e81f50bae2	1/15/2019 11:16:10 AM	ThomasZ	UpdateRemote 1 added 2 modified Save changes before commit
499721c1844a239e4f63e49422169d1103c	1/15/2019 11:14:53 AM	Zweitbenutzer	UpdateRemote 1 modified OnlineChange: : >> var nit changed to 102
64784a153a29620e5d816d045529a64696f	1/15/2019 11:14:53 AM	Zweitbenutzer	UpdateRemote 2 modified OnlineChange: : >> var nit changed to 101
393e6c7a27a3c570227185740c76e5e4e4b	1/15/2019 11:09:21 AM	Zweitbenutzer	UpdateRemote 1 modified OnlineChange: : >> nit changed
e4e81999051430579f396a62558e70a77a2	1/15/2019 11:07:43 AM	Zweitbenutzer	UpdateRemote 1 modified OnlineChange: : >> var added
7af6e4e6d9f85e4c2a30505e6878e99504	1/15/2019 11:04:20 AM	Zweitbenutzer	UpdateRemote 5 added
97e87152028164ee06333e42531ade118cbf	1/15/2019 11:01:17 AM	ThomasZ	UpdateRemote 5 added
6b7090e6534f6262674154a712a7f0e6d639	1/15/2019 11:01:15 AM	Beckhoff	Initial No Data/ Keine Daten

Die Multi-User-Funktionalität von TwinCAT erleichtert deutlich die Inbetriebnahme aus komplexer Maschinen und Anlagen.


```

MAIN -> X
1 PROGRAM MAIN
2 VAR
3   {IF defined (Variant1)}
4     (* The following variables are only declared, if the compiler define 'Variant1' is set *)
5     sVariantUsed      : STRING := 'Variant1';
6     bOutput          RT %Q*  : BOOL;
7   {ELSE}
8     (* The following variables are only declared, if the compiler define 'Variant1' is not set *)
9     sVariantUsed      : STRING := 'NotVariant1';
10    bInput           RT %I*  : BOOL;
11  {END_IF}
12
13  nCounter          : INT;
14 END_VAR

1 {IF defined (Group1)}
2   (* The following code is only executed, if the compiler define 'Group1' is set *)
3   nCounter := nCounter + 1;
4 {ELIF defined (Group2)}
5   (* The following code is only executed, if the compiler define 'Group2' is set *)
6   nCounter := nCounter - 1;
7 {END_IF}

```

Das neue Variantenmanagement in TwinCAT vereinfacht die Versionierung und Wartung der Steuerungssoftware erheblich.

Aber welche Referenzuhr sollte man einsetzen? Um auf eine sehr einfache Weise bereits eine mittlere Genauigkeit zu erreichen, eignet sich der kostenlose und auf allen Beckhoff-Geräten installierte (S)NTP-Service. Dieses Verfahren reicht bei verteilten Anlagen häufig aus, wenn es nur um den Vergleich von Daten geht. Für eine noch höhere Genauigkeit bietet sich das Verfahren nach IEEE 1588 an. Diese auch PTP (Precision Time Protocol) genannte Methode kann sehr genau unterschiedliche SPSen zeitlich miteinander abgleichen, bedingt jedoch eine spezielle Hardware wie die EtherCAT-Klemme EL6688. Eine weitere Steigerung der Genauigkeit ist dann nur noch durch explizite Verkabelung – also einem EtherCAT-Kabel zwischen den Geräten – möglich.

Durch die im neuen TwinCAT Build eingeführte Schnittstelle für eines dieser Verfahren lassen sich nun korrigierte Zeitstempel für unterschiedliche Daten verwenden und in aggregierten Datenbanken somit auch vergleichbare Zeitstempel erreichen.

Sichere Kommunikation mit Secure ADS

Das Beckhoff-Protokoll ADS wurde schon mit der ersten TwinCAT-Version eingeführt und im Kern nie geändert – aber natürlich immer wieder um wichtige Features erweitert. Eine neue Eigenschaft im TwinCAT Build 4024 ist, dass sicher per ADS kommuniziert werden kann. Hierbei wird eine verschlüsselte Verbindung zwischen zwei Teilnehmern aufgebaut, die dann wie gewohnt ADS-Telegramme austauschen können.

Beim Anlegen einer neuen Verbindung zwischen zwei ADS-Geräten kann die Authentifizierung über Zertifikate sichergestellt werden. Dabei werden alle Telegramme automatisch verschlüsselt. Mit TLS hat sich Beckhoff entschieden, das bekannteste Verfahren für Authentifizierung und Verschlüsselung zu nutzen. Um die Zertifikate und deren Verwaltung muss sich der Programmierer nicht kümmern. Das erledigt das TwinCAT-System selbstständig im Hintergrund. Durch die Integration in die zentrale TwinCAT-Kommunikationskomponente (TwinCAT Router) ergeben sich auch für ADS nutzende Bestandsanwendungen die Möglichkeit, durch Rekonfiguration der Verbindung eine verschlüsselte Verbindung zu nutzen, ohne dass die eigentliche Anwendung neu geschrieben oder auch nur neu kompiliert werden muss.



Dr. Josef Papenfort,
Senior Produktmanager TwinCAT,
Beckhoff Automation

Neue Eigenschaften im Safety-Engineering

Neben Security spielt natürlich auch Safety eine wichtige Rolle im TwinCAT-System. Deshalb wurden zahlreiche Eigenschaften ergänzt, die das Arbeiten im Safety-Bereich vereinfachen. Eine der wichtigsten Neuerungen ist die Mehrfachverwendung von Variablen. Variablen müssen somit nicht mehr eindeutig sein, sondern können auf lokaler und auch globaler Ebene – über ein gesamtes Safety-Projekt hinweg – deklariert und genutzt werden. Damit wird das Arbeiten im Bereich Safety einfacher und effektiver.

Weiterhin lassen sich basierend auf bestehenden und zertifizierten Funktionsbausteinen eigene Funktionsblöcke definieren und mehrfach instanzieren. Damit muss beispielsweise eine Funktionalität wie z. B. Sicherheitstür nur einmal erstellt werden. Bei mehreren Sicherheitstüren in einer Applikation kann dieser Baustein dann einfach in mehreren Instanzen und auch mit unterschiedlichen I/O-Belegungen bzw. Parametern wiederverwendet werden.

weitere Infos unter:

www.beckhoff.de/twinCAT3