

Mit TwinCAT PLC schnell und flexibel zum zuverlässigen Steuerungsprogramm

Speicherprogrammierbare Steuerungen sind seit vielen Jahren in der Industrie für verschiedene Einsatzzwecke bekannt und beliebt. Seit 1994 werden Steuerungen nach der Norm IEC 61131-3 programmiert. TwinCAT PLC unterstützt diese Norm ebenfalls, die nicht nur eine Programmiersprache, sondern eine komplette Architektur für Steuerungen beschreibt. Darin sind u. a. Tasks, Bausteine, Datentypen und Programmiersprachen beschrieben.

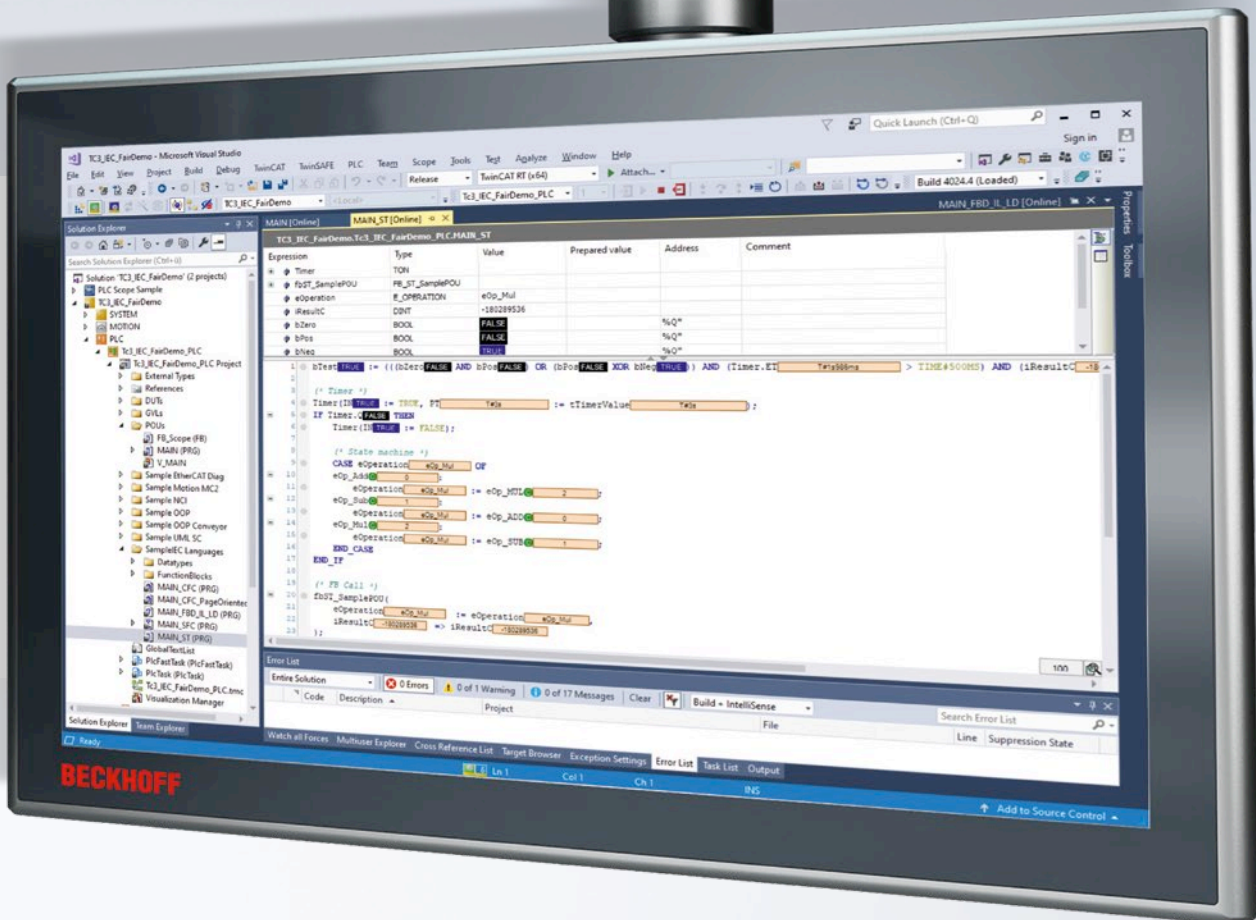
Die IEC 61131-3 stellt derzeit fünf verschiedene Programmiersprachen – zwei textuelle und drei grafische – zur Verfügung. Jeder Anwender kann dabei die Sprache seiner Wahl oder auch eine Kombination aus verschiedenen Sprachen in der TwinCAT PLC verwenden. Die Assembler ähnliche Programmiersprache AWL wird nur noch selten verwendet. Hauptsächlich kommt Structured Text (ST) zum Einsatz, was der Sprache Pascal ähnelt und als Hochsprache gilt. Die grafischen Sprachen Funktionsblockdiagramm (FUP, FBD) und Kontaktplan (KOP, LD) sind eher Programmiersprachen einer klassischen SPS. Hinzu kommt die Ablaufsprache (AS, SFC), mit der sich komplexe Schrittketten einfach grafisch programmieren lassen.

Flexibler Workflow und schnelle Optimierung

In der Engineeringumgebung TwinCAT XAE werden verschiedene Bausteine – Programme, Funktionsbausteine oder Funktionen – in den unterschiedlichen Sprachen erstellt. Startpunkt bei der Ausführung ist immer eine Task. Diese wird zyklisch in einer bestimmten Zeit (Zykluszeit) abgearbeitet. Die Priorität einer Task entscheidet, ob es zu Unterbrechungen im Ablauf kommen kann oder nicht. Eine höherprioritäre Task kann – wenn sie gerade eingeplant ist – eine niederprioritäre Task unterbrechen. Dieses präemptive Scheduling wird vom TwinCAT Realtime Kernel zur Verfügung gestellt. Eine Task ist auch dafür verantwortlich, Eingänge der SPS über den Feldbus von den Sensoren abzuholen und Ausgänge der SPS an die entsprechenden Aktuatoren zu schicken. Dazu verfügt jede Task über ein Prozessabbild, in dem alle Ein- und Ausgänge der Task gelistet und in der Konfiguration mit den Ein- und Ausgängen der Hardware verbunden sind. Mit der Standalone-PLC kann bei Bedarf das SPS-Projekt von der Hardware-Konfiguration getrennt werden. Auf diese Weise lässt sich ein Workflow etablieren, in dem der SPS-Entwickler nicht die Hardware-Konfiguration und der Elektriker nicht den SPS-Code ändern kann.

In der Regel muss ein erstelltes Steuerungsprogramm während der Inbetriebnahmephase „optimiert“, d. h. kleine Anpassungen durchgeführt und Fehler im Ablauf beseitigt werden. Alle aktuellen Werte werden in den entsprechenden Editoren der einzelnen Sprachen angezeigt. Mit der Ablaufkontrolle kann man auch erkennen, welche Zeilen oder Netzwerke gerade durchlaufen werden, was sehr schnell einen Überblick über das aktuelle Geschehen ermöglicht. Durch das Setzen von Variablen kann der Ablauf gezielt überschrieben und für Tests in eine gewünschte Richtung gebracht werden. Komplexere Abhängigkeiten zwischen Variablen können einfach grafisch mit dem Software-Oszilloskop TwinCAT Scope dargestellt werden. Das Besondere einer TwinCAT-SPS liegt darin, dass sich der Ablauf ändern lässt, ohne die Steuerung anzuhalten. Dieses als „Online Change“ bezeichnete Feature bietet große Vorteile, denn je nach Applikation kann ein Stopp der SPS entweder überhaupt nicht erlaubt sein oder zu großen Kosten führen.

Wird mit einem Team von Inbetriebnehmern gearbeitet, kann in TwinCAT die Multi-User-Funktionalität für das SPS-Projekt aktiviert werden. Diese Funktionalität unterstützt den Workflow, indem das System die Versionsstände des SPS-Projekts, die auf die Steuerung gespielt werden, automatisch abspeichert. Dadurch kann beim Einloggen einer geänderten Version ein Vergleich zum aktuell aktiven Stand im Vergleichstool TwinCAT Project Compare angezeigt werden. Auf diese Weise wird vermieden, dass bei einem Online Change bereits eingespielte Änderungen der anderen Inbetriebnehmer überschrieben werden bzw. ein manueller Austausch und Abgleich vorzunehmen ist. Für das Speichern und Zusammenführen der Versionsstände verwendet Multi User im Hintergrund Git, wobei aufgrund der tiefen Integration in den Engineeringworkflow für den Anwender kein Zusatzaufwand entsteht. Das Repository kann dabei wahlweise auf der Steuerung selbst oder einem separaten PC gespeichert werden.



Die Editoren der IEC 61131-3 sind mit TwinCAT nahtlos in die Visual-Studio®-Umgebung integriert.

Gesicherte Softwarequalität

Auch bei einem SPS-Programm muss die Qualität der Software gesichert werden. Das ist analog zur Softwareentwicklung in der IT zu sehen. Was macht die Standard-IT, um „gute“ Software zu schreiben? Zum einen kann in der Konzeptphase versucht werden, Bausteine zu kapseln und möglichst oft – auch mit leichten Modifikationen – wiederzuverwenden. Die Konzepte der objektorientierten Programmierung helfen hierbei. TwinCAT PLC bietet dafür einen UML Editor, mit dessen Hilfe Klassendiagramme für die Planung und Umsetzung wie auch State Charts unterstützt werden. Des Weiteren sollte die Software versioniert werden, d. h. jede Änderung wird geloggt, mit einem Autor und Kommentar versehen und kann jederzeit wiederhergestellt werden. Dafür wird die in Visual Studio® integrierte Schnittstelle zu Source-Code-Verwaltungssystemen verwendet, sodass der Entwickler das gewünschte Verwaltungssystem frei wählen kann. Die integrierte statische Code-Analyse bietet zudem die Möglichkeit, mit zuvor festgelegten Kodierrichtlinien die Einheitlichkeit und Qualität des Codes zu garantieren. Sie kann manuell angestoßen oder automatisch mit der Codeerzeugung durchgeführt werden.

Automatisierte Softwaretests sind ein weiterer Baustein zur Qualitätssicherung, die frei mit den vorhanden Schnittstellen in einer Kombination aus Unit,

Integrations- und Systemtests eingeführt werden können. Und wenn die Software am Ende auf der Maschine ausgeführt wird und schon sicher läuft, kann mithilfe des TwinCAT PLC Profiler die Software optimiert werden. Dabei werden Marker automatisch im SPS-Code hinzugefügt und die Ausführungszeiten gemessen und analysiert. Auf diese Weise können z. B. eine Hot-Path-Analyse für das Identifizieren von Programmteilen mit hohem Zeitverbrauch und weitere Detailanalysen durchgeführt werden. Das TwinCAT Engineering unterstützt den SPS-Entwickler somit entlang des gesamten Software-Lifecycle.

Links: Béla Höfig,
Produktmanager TwinCAT
Rechts: Dr. Josef Papenfort,
Produktmanager TwinCAT



weitere Infos unter:

www.beckhoff.com/tc1200