

Entwicklungsumgebung für IEC-61131-3-Programmierung

Plattformunabhängig programmieren, produktiv produzieren



Mit dem webbasierten Entwicklungstool Codesys Go! will Codesys die Lücke zwischen klassischer OT und moderner IT-Entwicklung schließen. Die plattformunabhängige Lösung ermöglicht erstmals die nahtlose Integration von IEC-Programmierung in CI/CD-Pipelines und DevOps-Workflows. Im Interview erläutern Roland Wagner, Leo Bareth und Martin Decker, warum Go! keine Konkurrenz zum etablierten V3 ist, sondern eine Ergänzung – und wie die für beide Tools identische Runtime volle Kompatibilität garantiert.

Codesys Go! – was steckt dahinter und wie unterscheidet es sich vom bewährten Development System V3?

Roland Wagner: Das Codesys Development System V3 ist und bleibt unser etabliertes Arbeitspferd mit enormem Funktionsumfang. Codesys Go! wird es ergänzen, nicht ersetzen. Aber die V3 ist ein reines Windows-Tool – unter Linux läuft es höchstens in einer VM. Das



► Bei Codesys in Kempten erfuhrt das SPS-MAGAZIN aus erster Hand alles über Codesys Go! (Martin Decker, Kevin Ketterle, Frank Nolte, Markus Lehnert, kniend Leo Bareth, Roland Wagner (v.l.n.r.)).

Leo Bareth: Der zweite Treiber: Wir wollen bewährte Entwicklungsprozesse aus der IT-Welt für die Applikationsentwicklung für Industriesteuerungen zugänglich machen. Immer mehr Auto-

plex: Man braucht eine VM, muss APIs anbinden und so weiter. Mit Go! wird das radikal einfacher.

Ist V3 für die OT und Go! mehr für die IT?

Leo Bareth: Go! ist definitiv ein OT-Tool – am Ende schreibt man IEC-Code, den die SPS ausführt. Aber warum sollten nicht auch SPS-Programmiererinnen und -programmierer von Methoden aus der IT-Entwicklung profitieren? Mit modernen Entwicklungstools, Versionskontrolle und automatisierten Tests, einfach integrierbar in CI/CD-Pipelines, wird die Arbeit viel effizienter. Gleichzeitig kann man agil auf Änderungen reagieren.

Roland Wagner: So haben wir das auch in der Vergangenheit gemacht. Bereits vor 20 Jahren führten wir die Objektori-



Codesys Go! und V3 nutzen ein und dieselbe Runtime.

Leo Bareth, Product Marketing

reicht heute nicht mehr. Go! ist dagegen konsequent plattformunabhängig: Das Backend läuft auf Windows oder Linux, das Frontend im Browser. Man kann das Backend sogar direkt auf einer leistungsfähigen Steuerung installieren!

matisierungsprojekte werden kontinuierlich weiterentwickelt, oft von mittleren oder großen Teams. Dabei stoßen wir mit V3 an Grenzen, besonders bei der sogenannten DevOps-Integration. In der Praxis ist das möglich, aber kom-



“

Codesys Go! ist eine Ergänzung zu V3 – kein Ersatz.

Roland Wagner, Head of Product Marketing

entierung in die SPS-Programmierung ein – nutzbar, aber nicht zwingend. Dabei bleibt ein Kontaktplan-Editor weiterhin wichtig und deshalb ist er in einer der ersten Versionen von Codesys Go! enthalten.

SPS Welche Funktionen bietet Codesys Go! konkret?

Martin Decker: Das erste Release konzentriert sich auf die ST-Bibliotheksentwicklung. Dafür bringt Go! alles mit, was man braucht: einen IEC-Editor für Strukturierten Text, Compiler und die nötigen Schnittstellen. Viel Entwicklungsarbeit ist in die grundlegend neue Architektur geflossen – deshalb wird der Funktionsumfang anfangs noch überschaubar sein. Dafür steht der Unterbau für die Zukunft.

Roland Wagner: Ganz konkret: Man kann in Go! eine ST-Bibliothek entwickeln, über die Kommandozeile kompilieren, automatisiert in CI/CD-Pipelines weiterverarbeiten und dann in einem umfangreichen V3-Projekt einsetzen. Die beiden Welten existieren also nebeneinander. Als ersten Anwendungsfall sehen wir bewusst nicht die klassische Inbetriebnahme an der Maschine.

SPS DevOps und CI/CD ist jetzt schon ein paar Mal gefallen – können Sie das für OT-Leute erklären?

Martin Decker: Stellen Sie sich vor, Sie ändern eine Funktion in Ihrer Bibliothek. Traditionell müssten Sie manuell kompilieren, testen, dokumentieren, an Kollegen verteilen. Bei DevOps passiert das automatisch: Sie geben Ihren Code an das System, und im Hintergrund wird er kompiliert, gegen definierte Testfälle geprüft, auf einer virtuellen Steuerung getestet und bei Erfolg freigegeben. Schlägt ein Test fehl, wissen Sie sofort Bescheid. Das läuft bei jeder Änderung gleich ab – egal ob Sie oder ein Kollege sie macht. Gerade bei Projekten, die über Jahre weiterentwickelt werden, verhindert das, dass sich Fehler einschleichen oder Dokumentation veraltet.

Leo Bareth: Weil das Backend von Go! auf einem Firmenserver laufen kann, spart man sich auch die Versionspflege: Entwickler öffnen Go! im Browser und haben automatisch die richtige Version inklusive Gerätebeschreibungen und passender Bibliotheken.

SPS Welche Einschränkungen gibt es bei Go! gegenüber V3?

Leo Bareth: Go! ist bewusst fokussiert. Statt die V3 in einer neuen Oberfläche nachzubauen, haben wir in einen zukunftsfähigen Unterbau investiert – etwa ein Extension-Modell, mit dem wir und Kunden eigene Add-ons entwickeln können. Aber auch in die Trennung zwischen Backend und Frontend,

die neue dateibasierte Projektablage, und vieles mehr. Dafür fehlen in den ersten Versionen leider noch Funktionen, die man aus der V3 kennt: weitere Spracheditoren, Konfiguratoren, Visualisierung oder Feldbusse.

Martin Decker: Wie bereits gesagt: Das ist in Ordnung so, weil wir mit dem ersten Release nur auf die Biblio-

theksentwicklung zielen, als Ergänzung zur V3. In weiteren Versionen schließen wir Lücken sukzessive. Entscheidend ist: Beide Tools arbeiten zusammen. In Go! entwickelte Bibliotheken lassen sich in V3 mit dem Test Manager testen. Sie nutzen vertraute Workflows weiter, während Sie Go! für spezifische Aufgaben einsetzen. Das war beim Umstieg von V2 auf V3 vor 20 Jahren nicht möglich – diesmal schon.

Leo Bareth: Und ganz wichtig: Go! und V3 nutzen dieselbe Runtime. Kompilierter Code aus Go! läuft auf jedem Gerät, das auch V3-Code ausführt und außerdem lassen sich die Bibliotheken aus der V3 in Go! nutzen. Volle Kompatibilität in dieser Hinsicht.

SPS Kann man denn problemlos zwischen V3 und Go! hin- und herschalten?

Leo Bareth: Nahtlos nicht – obwohl das ursprünglich geplant war.

- Anzeige -

BECKHOFF

Beckhoff Automation GmbH & Co. KG
33415 Verl | Tel.: +49 5246 963-0
info@beckhoff.com
www.beckhoff.com

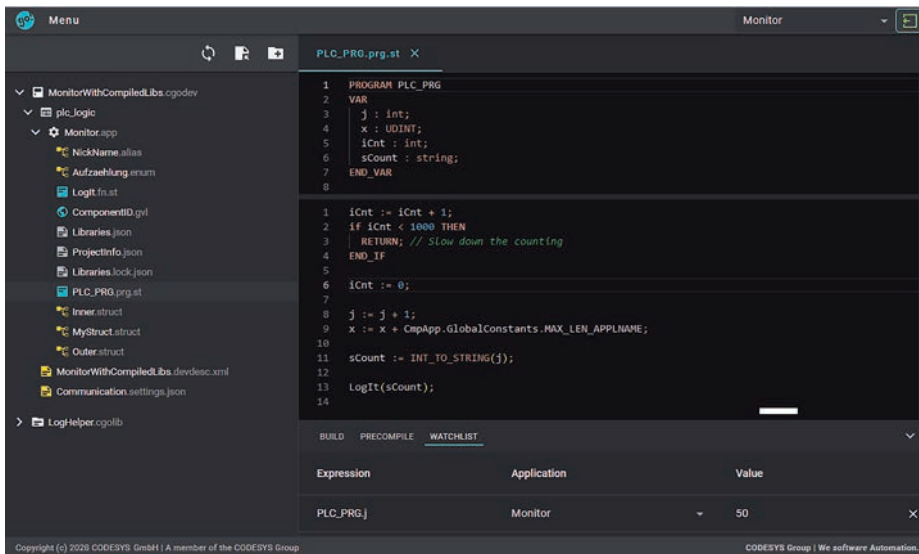
PC-based Control: SPS, Motion Control, Messtechnik und Robotik auf einem PC



- IPC: PCs in verschiedenen Formfaktoren
- I/O: I/Os für alle Signaltypen und Feldbussysteme
- Motion: Hochdynamische Servoverstärker und -Motoren
- TwinCAT: Automationssoftware für Multi-SPS, NC, CNC

Mehr
Informationen:





Die Programmierung mit Codesys Go! erfolgt im Webbrowser – ob auf PC oder Tablet, mit Linux, Windows oder iOS.

Wir wollten die Modernisierung nicht durch strikte V3-Kompatibilität blockieren. Aber es wird ein Konvertierungstool geben, das einen bequemen Wechsel zwischen beiden Plattformen ermöglicht. In den meisten Fällen funktioniert das problemlos. Es kann allerdings vorkommen, dass sich bestimmte Go!-Projekte nicht eins zu eins in der V3 abbilden lassen. Der kompilierte Code selbst ist aber mit beiden Plattformen nutzbar.

SPS Welche Programmiersprachen unterstützt Codesys Go!?

Roland Wagner: Das erste Release enthält Strukturierten Text und KOP. Aber natürlich implementieren wir AS und FUP als noch fehlende IEC-Sprachen, weil Automatisierer sie gerne nutzen. Das unterscheidet uns auch von anderen Lösungen, die oft nur textuelle Sprachen unterstützen.

SPS Was bringt die webbasierte Architektur?

Roland Wagner: Plattformunabhängigkeit ist ein Hauptvorteil. Es gibt den Backend-Teil, der auf Windows oder Linux läuft, sowie den Frontend-Teil mit Editoren, die im Browser ausgeführt werden.

Martin Decker: Dadurch sind verschiedene Architekturen denkbar. Go! kann z.B. auch direkt auf einer Steuerung laufen und ohne PC bedient werden. Ein praktisches Szenario: Die SPS ist im Schaltschrank, ein Problem tritt auf. Weil alles auf der Steuerung ist, steckt man sein Tablet per Kabel

an, öffnet den Browser, gibt die IP-Adresse ein und ist direkt auf der Entwicklungsumgebung. Das vereinfacht die Fehlersuche deutlich. Das wird aber mit den ersten Releases aufgrund fehlender Tools noch nicht so möglich sein, auch wenn die Architektur es schon hergeben würde.

“

Durch die Plattformunabhängigkeit sind verschiedene Architekturen denkbar.

Martin Decker, Product Marketing

Leo Bareth: Durch die Server-Client-Architektur kann es, wie vorhin erwähnt, auch auf einem zentralen Server für Team-Entwicklung laufen, was die Versionsverwaltung vereinfacht.

SPS Wann kommt die erste Version?

Roland Wagner: Die erste Freigabe wird im April sein. Weitere Freigaben folgen im Rhythmus von etwa zwei Monaten. Wir arbeiten schon an den Themen wie Visualisierung, weiteren Feldbussystemen, Motion und OPC UA. Zum Feldbus: Da wir die Protokollstacks in IEC-Bibliotheken programmiert haben, können wir sofort diese einfach übernehmen – ein Riesenvorteil. Die grafischen Konfiguratoren machen wir neu. Hier fangen wir mit Modbus an, der immer noch weit verbreitet ist.

Martin Decker: Das Versioning orientiert sich übrigens an der IT-Entwicklung – also 2026.04 für das April-Release.

SPS Wie sieht das Lizenzmodell aus?

Martin Decker: Der erste Release wird kostenfrei und bereits für den Produktiveinsatz geeignet sein. Zu einem späteren Zeitpunkt, wenn der Funktionsumfang es dann rechtfertigen kann, kommt eventuell ein kostenpflichtiges Lizenzmodell. Aber das wird frühestens 2029 der Fall sein.

SPS Wie sehen Sie die IT-OT-Konvergenz?

Roland Wagner: Wir sehen das als eine ganz natürliche Entwicklung. Automatisierer, die mit Go! arbeiten, bekommen IT-Methoden an die Hand – Versionskontrolle, automatisierte Pipelines –, ohne dass sie es groß merken. Und ITler können mit ihren gewohnten

Tools OT administrieren, etwa unsere virtuellen Steuerungen aber auch eine Go!-Instanz.

Martin Decker: Mit der Vernetzung steigen auch die Anforderungen an die Sicherheit – Stichwort Cyber Resilience Act. Moderne Entwicklungsprozesse helfen dabei: Wenn Code automatisiert getestet und dokumentiert wird, lassen sich Compliance-Anforderungen leichter erfüllen. Auch das ist ein Ziel von Go! ■

Das Interview führte

Dipl.-Ing. (FH) Frank Nolte, Chefredakteur