

Funktionserweiterung der SPS-Programmierung

Hören
statt lesen!



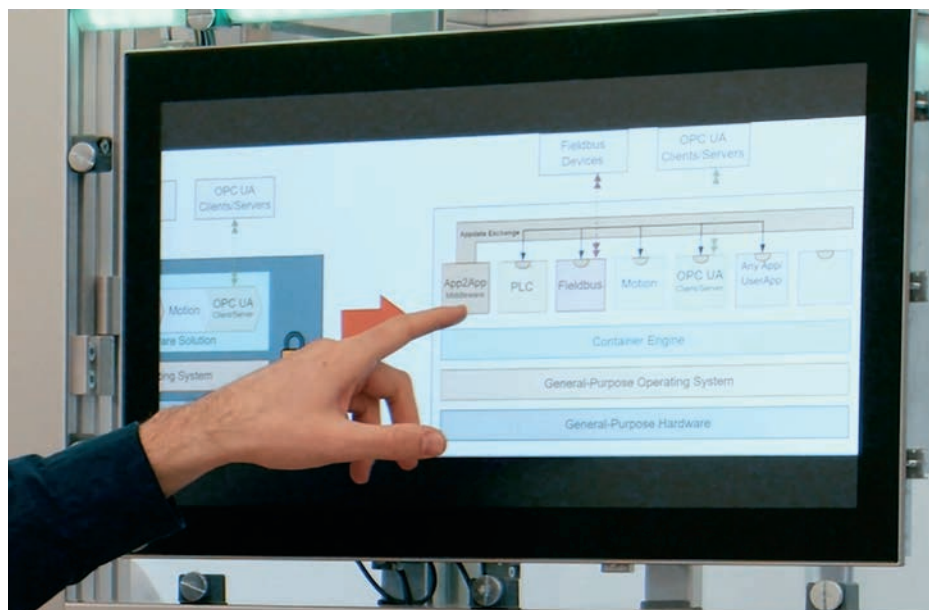
So wird die SPS bereit für die Zukunft

Früher basierten Maschinensteuerungen auf fest verdrahteten Steuerungen. Diese Logik wurde auf die Programmierungsumgebungen übertragen und somit ist auch die SPS-Programmierung entsprechend gestaltet. Bei Funktionalitäten, die über die klassische Ansteuerung einer Maschine hinausgehen, wie Vernetzung oder Machine Learning, ist sie allerdings deutlich limitiert. Warum deshalb eine Verknüpfung mit moderner Software notwendig ist und was die Modernisierung der SPS-Programmierung bringt erfuhrt das SPS-MAGAZIN im Gespräch mit Thomas Barth und Aleksandr Sidorenko, beide Researcher der SmartFactory-KL.

SPS Die SPS-Programmierung gilt zwar als etabliert und zuverlässig, allerdings aufgrund ihrer Struktur auch als antiquiert. Inwieweit lässt sie sich da überhaupt sinnvoll und praktikabel mit moderner Software verknüpfen?

Barth: Aus unserer Erfahrung ist eine sinnvolle und praktikable Verknüpfung von SPS-Programmierung und moderner Software durchaus möglich – vorausgesetzt, die jeweiligen Aufgabenbereiche der beiden Programmierwelten sind voneinander abgegrenzt und die Schnittstellen sind klar definiert. Die Stärken der SPS liegen vor allem in der Nachbildung von Schaltlogiken sowie in der Ansteuerung von Aktorik und der Auslesung von Sensoren - insbesondere bei Aufgaben mit Echtzeitanforderungen. Ein weiterer Vorteil ist die Verfügbarkeit bewährter, standardisierter Bibliotheken, wie die der PLCopen, auf diesen aufgebaut werden kann.

Sidorenko: Ein gutes Beispiel für die Synergie zwischen klassischer SPS-Software und modernen Programmiersprachen ist die skillbasierte Produktion, auf der unsere Demonstratorökosystem für flexible Fertigung basiert. In unseren Demonstratoren nutzen wir die SPS-Programmierung, um die Grundfunktionen einzelner Maschinenkomponenten zu realisieren. Dabei analysieren wir die Produktionsmaschine hinsichtlich ihrer elementaren, voneinander unabhängigen Fähigkeiten welche wir als Atomic Skills bezeichnen. Diese können beispielsweise das Befördern eines Bauteils in-



► Durch eine leistungsfähige Vernetzung von SPS mit Softwareapplikationen aus anderen Domänen lassen sich nicht nur flexiblere sowie kostengünstigere Automatisierungslösungen realisieren, sondern auch neue Ansätze für die Konvergenz von IT und OT erschließen.

nerhalb der Maschine, das Drehen einer Werkzeugkassette oder das Öffnen und Schließen eines Greifers sein.

Barth: Die so beschriebenen Maschinenfähigkeiten, die mittels SPS implementiert werden, lassen sich anschließend flexibel von moderner Software – z.B. in C++, Python oder Java – über OPC UA aufrufen, parametrisieren und ansteuern. Diese Herangehensweise eröffnet uns neue skalierbare Möglichkeiten, Maschinen flexibler anzusteuern, insbesondere im Hinblick auf die Konvergenz von OT- und IT-Welt.

SPS Deckt IEC61131-3 diese Möglichkeiten überhaupt ab?

Sidorenko: Das Standardsoftwaremodell sowie die Sprachen der IEC61131-3 werden dafür kritisiert, dass sie den Anforderungen moderner Steuerungssoftware nicht gerecht werden. Jedoch hat die Aufnahme von objektorientierten (OOP) Funktionen in die dritte Ausgabe der Norm die Modularität und Struktur des SPS-Codes verbessert. Die OOP-Version der strukturierten Textsprache (ST) bietet alle wichtigen Merkmale der modernen OO-Sprache, wie Klassen, Schnittstellen, Methoden und Eigenschaften. Einige Features, die in



Hochsprachen üblich sind, werden aus gutem Grund nicht unterstützt, da sie die Echtzeit- und Sicherheitsanforderungen gefährden sowie die Effizienz behindern, die für die Entwicklung von Software für kritische Systeme unerlässlich sind.

Barth: Obwohl solche IEC61131-Sprachen wie AWL und KOP in der Tat antik erscheinen, bietet die FUP einige Vorteile für den Softwareentwurf. Ein komponentenbasierter Entwurf (CBD) ist ein beliebter Ansatz in der Softwareentwicklung, der darauf abzielt, die Modularität und Wiederverwendung von Software zu verbessern. Dabei wird eine Softwarekomponente als Black-Box betrachtet, die über vordefinierte Schnitt-

stellen und Rollen bei der Softwareentwicklung, verkürzt die Gesamtentwicklungs- und Anlaufzeiten der Produktionslinien und verbessert die Wartung und Wiederverwendbarkeit der industriellen Steuerungssoftware.

SPS **Und wie sieht es mit der IEC61499 aus, die ja schon seit vielen Jahren die IEC61131-3 ablösen sollte.**

Sidorenko: Die Norm IEC61499 sollte die Einschränkungen der IEC61131 überwinden. Sie konzentriert sich auf die Entwicklung von verteilter Automatisierungszelligenz. Sie hat die Kapselung von Daten und Funktionen in den Funktionsblöcken erheblich verbessert und fortschrittliche verteilte Steuerungssysteme durch die

keine große Beachtung gefunden. Das liegt primär an ihrer Komplexität und den Unterschieden zur IEC61131, die damals bereits weit verbreitet war. Die Situation scheint sich in letzter Zeit zu ändern, da einige große Automatisierungsanbieter ihre kommerziellen Produkte auf der Grundlage der Norm IEC61499 entwickelt haben.

SPS **In wie weit passen moderne Programmierkonzepte wie Container und Docker in die SPS-Welt?**

Sidorenko: Ist die SPS-Programmierung bereit für eine moderne Programmierung mit Apps und Containern? Unserer Auffassung nach: Ja. Einige Automatisierungshersteller haben bereits gute Lösungen entwickelt, die zeigen, wie sich die Vernetzung einer klassischen SPS mit moderner Software verbessern lässt. Der zentrale Punkt ist dabei eine effiziente und performante Schnittstelle zur SPS bzw. eine Middleware, an die die SPS angebunden ist und die vor allem die



Moderne SPS haben sich zu vielseitigen und leistungsstarken Edge Devices entwickelt, die Echtzeitgarantien mit IT- und KI-Technologien kombinieren. Zur Unterstützung der neuen Generation skill-basierter, cyber-physischer Produktionssysteme ist jedoch eine bessere Integration von Low-Level-Code und aufgabenorientierter KI-Steuerung erforderlich.

Alexandr Sidorenko



stellen mit anderen Komponenten interagiert. Das übliche Muster von CBD besteht darin, die Funktionsblöcke auf der niedrigsten Abstraktionsebene an den Geräten auszurichten und sie dann zu zusammengesetzten Funktionsblöcken zu komponieren, die komplexere, aggregierte Komponenten darstellen, und so weiter. So entsteht eine natürliche Hierarchie, wie sie für Produktionsumgebungen typisch ist. Nach CBD können Steuerungsexperten den Low-Level-Code in ST oder C++ schreiben und als Bibliotheken von Funktionsblöcken bereitstellen, welche die Anwendungsingenieure dann in grafischen Entwicklungsumgebungen verwenden können. Dieser Ansatz verbessert die Trennung von Anlie-

Einführung eines ereignisbasierten Berechnungsmodells in die SPS-basierte Steuerung ermöglicht. Die IEC61499 folgt dem CBD. Die Steuerungsalgorithmen in den Funktionsbausteinen können nicht nur mit IEC61131-Sprachen, sondern auch mit den gängigen Hochsprachen implementiert werden. Das Netzwerk der Funktionsblöcke dient dabei der nahtlosen Integration von Softwarekomponenten, die in verschiedenen Sprachen geschrieben wurden, und gewährleistet die deterministische Ausführung der Steuerungsanwendung, die auf verschiedene Automatisierungsgeräte verteilt werden kann.

Barth: Die Norm ist in der Wissenschaft anerkannt, hat aber in der Industrie

Interoperabilität zu Domainfremder Software ermöglicht. Dadurch ist beispielsweise eine Maschinensteuerung um Funktionen erweiterbar, die mit der SPS eigenen Sprache nur schwer oder gar nicht umsetzbar wären.

Barth: Ein weiterer entscheidender Aspekt ist die Virtualisierung von Steuerungen in Form von Software-Containern. Diese trägt wesentlich zur Effizienz und Performance der Vernetzung zwischen verschiedenen Softwareeinheiten bei. In klassischen industriellen Steuerungssystemen ist eine Software meist an eine bestimmte Hardware gebunden. Daher erfolgt die Kommunikation zwischen Softwareinstanzen tradi-

- Anzeige -

TURN YOUR HEAD



- S7-SPS
- S7-Panels
- S7-llot

- Tools und Programme beibehalten
- Bestandsprojekte samt Retrofit langfristig fortführen
- S7-Anbindung an llot schnell, einfach und wirtschaftlich

INSEVIS for independent minds



Jetzt 2nd Source für S7 aktivieren www.insevis.de

- S7-Panel-SPS
- S7-Kompakt-SPS
- S7-Panel-HMI
- Peripherie
- Software
- Energiemanagement
- S7-llot Gateways



Eine standardisierte und performante Vernetzung der SPS mit Softwareapplikationen aus anderen Domänen ermöglicht flexiblere sowie kostengünstigere Automatisierungslösungen und eröffnet neue Ansätze für die Konvergenz von IT und OT.

Thomas Barth

tionell über das Netzwerk (Host-to-Host), da diese physisch voneinander getrennt sind.

Die zunehmende Virtualisierung von Automatisierungssoftware und die damit verbundene Entkopplung von Hardware leiten jedoch einen Paradigmenwechsel ein. Dieser ermöglicht es beispielsweise, eine SPS auf demselben Hostsystem zu betreiben wie eine Applikation in einer beliebigen Programmiersprache. Dadurch lässt sich eine effiziente Intra-Host-Kommunikation realisieren, die den bisher netzwerkbasierten Datenaustausch ersetzen kann. Dies wiederum eröffnet neue Ansätze, wie den Einsatz von Microservice-Konzepten in Kombination mit einer transparenten Middleware für Automatisierungssoftware.

SPS Wie läuft das in der Praxis ab?

Sidorenko: In der Praxis setzen wir solche Steuerungsarchitekturen in unseren Demonstratoren ein, um verschiedene Softwaresysteme miteinander zu verknüpfen, von der Ansteuerung eines IO-Link-Greifers über Python bis hin zur Synchronisation eines Roboters mit einer Linearachse. In letzterem Fall haben wir eine C++-Applikation entwickelt, die über die herstellerspezifische App2App-Middleware die Position der Linearachse in Echtzeit aus dem Ethercat-Master ausliest und diese über ROS2 zur Verfügung stellt. Diese Applikation wurde dann auf dem Antriebsregler deployt, auf dem gleichzeitig die virtuelle SPS betrieben wurde.

Darüber hinaus ermöglicht dieses Konzept die Auslagerung von Steuerungslogik in beliebige Programmiersprachen. Dies lässt sich beispielsweise gut mit dem Skill-Konzept (Maschinenfähigkeiten) kombinieren. Man unterscheidet dabei zwischen Atomic Skills und Composite Skills:

- Atomic Skills sind grundlegende Funktionen, wie z. B. das lineare Bewegen eines Roboters oder das Öffnen und Schließen eines Greifers.
- Composite Skills setzen mehrere Atomic Skills zu einer höherwertigen Fähigkeit zusammen – beispielsweise eine Pick&Place-Funktion, die Bewegung und Greifvorgang kombiniert.

Barth: Übertragen auf eine Produktionsmaschine bedeutet dies, dass deren Fähigkeiten über ein einheitliches Informationsmodell auf einem OPC UA-Server bereitgestellt werden. Dadurch ist es möglich die Maschine flexibel von einem Drittsystem, beispielsweise einem MES, je nach Auftragseingang anzusteuern.

Das Anbieten atomarer Skills über einen OPC UA-Server ist jedoch aufgrund von Performance-Einschränkungen oft nicht sinnvoll, da die einzelnen Funktionsaufrufe zu hohen Latenzen führen können. Anders verhält es sich bei einer performanten Intra-Host-Middleware, die eine schnelle Kommunikation innerhalb eines Systems ermöglicht. Eine SPS könnte dabei atomare Funktionalitäten einer Maschine bereitstellen, wie MoveLinearAxis oder MoveLinearAxisTo-

Waypoint. Diese lassen sich flexibel von einem externen Programm, z.B. in Python, kombinieren oder um zusätzliche Funktionen erweitern, etwa durch die Interaktion mit IT-Systemen.

Der Vorteil dieses Ansatzes liegt in der klaren Trennung der Logikebenen:

- Die SPS-Sprache bleibt auf jene Logik beschränkt, für die sie optimiert ist – wie die digitale Umsetzung von Verschaltungslogiken. Hochsprachen wie Python eignen sich hingegen für „High-Level“-Logiken, etwa die flexible Koordination von Maschinenfunktionen auf Basis komplexerer Entscheidungskriterien und Informationsquellen. ■



- Anzeige -

„u-software? Von Weidmüller? Das sind doch die mit den Klemmen?!“

Richtig, die Reihenklemme haben wir vor 75 Jahren erfunden. Und heute revolutioniert u-software Industrial IoT & Automatisierung. Neugierig?



Mehr Informationen unter:
www.weidmueller.de/u-software





Bild: Gutex

Wettbewerbsvorteil durch objektorientierte Programmierung

HOLZFASERN EINBLASEN

Im preissensitiven Umfeld benötigen Maschinenbauer vor allem zwei Dinge: eine kostengünstige, leistungsstarke Hardware und ein Entwicklungswerkzeug, das eine schnelle Markteinführung unterstützt. Wie die Vorteile genau aussehen, zeigt das Anwendungsbeispiel eines Herstellers von Einblasmaschinen für die Dämmtechnik, der in Sachen Automatisierung auf eine Systemlösung von Sigmatek setzt.

Holz ist ein Dämmstoff, der sich gut für die Isolation von Häusern eignet. Grund hierfür ist die Fähigkeit, Wärme zu speichern und sie zeitversetzt wieder abzugeben. Diese Phasenverschiebung beträgt ziemlich genau zwölf Stunden, sodass die Sonnenenergie, die tagsüber auf eine Fassade fällt, erst wieder nachts abgegeben wird, wenn sich die größte Hitze bereits gelegt hat. Da Häuser aus Massivholz teuer sind, braucht es Alternativen, um diese nützlichen Eigenschaften in der Breite nutzen zu können. Eine Alternative findet sich in der Verarbeitung von Holzschnitzeln zu schwer brennbaren Holzfasern, die mit Hilfe spezieller Maschinen in die Außenhülle des Hauses, bei Bedarf auch in Innenwänden und Böden zur Schalldämpfung, geblasen werden.

Ein Hersteller der dafür benötigten Einblasmaschinen ist die Firma Gutex aus Waldshut-Tiengen. Sie fertigt die Maschinen in verschiedenen Ausführungen, die sich etwa in der Aufnahme sowie in der Aufbereitung der Holzfasern unterscheiden. Sie verarbeiten je nach Ausführung einen oder mehrere Großballen mit je 270kg Gewicht. Die Ballen bestehen aus gepressten Holzfasern, die mit Hilfe einer Vorrichtung je nach Modell mit einem Kamm oder mit verzahnten Walzen aufgelöst werden. Von dieser Vorrichtung gelangen sie in ein Modul, das übrige Brocken und Klumpen auflöst. Anschließend fallen die Fasern in eine Zentralschleuse, die das Material über die Geschwindigkeit sowie mit Hilfe eines Schiebers portioniert, der die Größe des Einlasses reguliert. Von dort gelangen die nach dem First-in-First-out-Prinzip aufbereiteten Fasern über einen Schlauch zu

den Dämmplatten, wo sie mit einer Dichte von 38kg/m³ eingeblasen werden.

Allen Einblasmaschinen von Gutex gemein ist die Software, die sich modular erweitern und in ein übergeordnetes System integrieren lässt. Warum das so ist, erklärt Entwicklungsleiter Alexander Janke: „Wir stehen mit unseren Mitbewerbern in einem preissensitiven Umfeld. Daher brauchen wir eine Software, die auf allen verschiedenen Maschinen läuft.“

Beschleunigte Softwareentwicklung

„Mit dem objektorientiertem Engineering Tool Lasal von Sigmatek kann die Applikationssoftware individuell passend und trotzdem erweiterbar und pflegeleicht gestaltet werden. So lassen



sich Maschinenvarianten einfach realisieren“, betont Arno Schmied, Leiter Technik bei Sigmatek in der Schweiz. „Das ist für die Entwicklung zukunftsfähiger Maschinenkonzepte unerlässlich. Zudem ist die Software klar strukturiert und übersichtlich aufgebaut.“ Dieser Ansatz erfordert jedoch zuerst ein Softwarekonzept zu erstellen, um eine Basis zu schaffen, auf die alles weitere aufgebaut werden könne. „Jede Maschine braucht beispielsweise ein Gebläse für das Fördern des Materials. Dieses bildet die Basis auf der alles Weitere flexibel aufgebaut wird. So lässt sich bereits getesteter Code wiederverwenden, was durchgängige Modularität und Übersichtlichkeit bringt“, so Schmied weiter. „Dadurch wird ein Spaghetticode vermieden.“ Das wäre nicht nur aufwendig, sondern würde auch die Pflege und den Unterhalt des installierten Bestands erschweren. Außerdem beschleunigt die objektorientierte Programmierung bei weiteren Entwicklungen den Aufwand. So dauerte die Softwareentwicklung für den Maschinentyp Feeder Lift zwei Monate, die des Modells Feeder Drop nur noch zwei Wochen.

Remote-Systeme reduzieren Reisebedarf

Ein weiterer Vorteil des objektorientierten Software-Ansatzes ist eine schnellere Markteinführung. Die Maschine kann schon lange bevor sie physisch existiert, ausgiebig getestet werden. Simuliert wird aber nicht etwa mit einem digitalen Zwilling. Aus Kostengründen wird hierfür das im Engineering Tool Lasal integrierte Simulationstool Lars genutzt. Dort wurden Funktionsklassen erstellt, mit denen sich Datenpunkte setzen und Elemente schalten lassen. Die Software lässt sich ohne elektrische Anbindungen an die Maschine simulieren.

Dieses Vorgehen setzt eine enge Zusammenarbeit mit dem Kunden voraus, bei der die Software-Entwickler schon während der Konzeptphase miteinbezogen werden. Bei Sigmatek funktioniert das einwandfrei, so Jaenke. Es ist aber nicht nur das Entwicklungswerkzeug Lasal, das Gefallen bei Gutex findet. Ebenfalls überzeugt ist man von der

Bilder: Gutex



► Alle Gutex-Maschinen basieren auf einer einheitlichen Software. So lassen sich Maschinenvarianten einfach realisieren.



Bild: Sigmatek GmbH & Co KG

► Das Unternehmen Gutex ist auf Maschinen für die Verarbeitung von Holzschnitzeln als Dämmmaterial spezialisiert.

Hardware, die in den Einblasmaschinen zum Einsatz kommt. Konkret handelt es sich um kompakte S-Dias-Steuerungen mit integrierter Safety-Funktionalität sowie ETT-Webpanels mit HTML5-Visualisierung. Die Vorteile dieser Systemlösung kommen speziell im Bereich der Antriebstechnik zum Tragen, wo die Motoren je nach Bedarf mittels Frequenzumrichter vom Typ FDD 3000 oder mit den kompakten 48V-Servo-Drive DC101 aus der S-Dias-Motion-Reihe realisiert werden.

In diesem Zusammenhang unterstreicht Jaenke. „Wir können es uns schlichtweg nicht leisten, andere Systemlösungen zu

verwenden, da wir in einem Verdrängungswettbewerb stehen und für uns die Herstellkosten entscheidend sind. Auch die unkomplizierte Software-Lizenzpolitik von Sigmatek hilft uns bei der täglichen Arbeit.“ Außerdem weiß er die effiziente Zusammenarbeit zu schätzen. „Mit Arno Schmied haben wir einen erfahrenen Softwareentwickler an der Seite, der sich gut in der industriellen Einblastechnik auskennt. Das ist für unser Geschäftsfeld ausschlaggebend.“ ■

Sigmatek GmbH & Co KG
www.sigmatek-automation.com

i i-need.de/p/4225