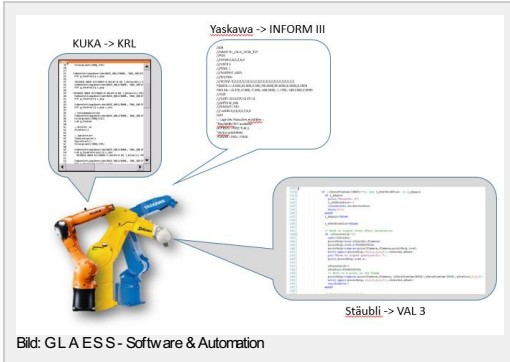


Erschienen am: 11.08.2017, Ausgabe SPS-MAGAZIN 8 2017

Roboterprogrammierung mit SPS-Entwicklungsumgebung

Von der SPS auf den Roboter

Um einen Roboter in die Produktionslinie zu integrieren, sind Anlagenbauer gezwungen, fehlendes Roboterwissen hausintern aufzubauen oder extern einzukaufen. Dabei gilt es, wichtige Fragen zu klären, wie: 'Welcher Roboter ist der richtige?' oder 'Wer kann den Roboter in Betrieb nehmen?' Darüber hinaus lässt sich fehlendes Roboter-Know-how mithilfe der Programmierung über eine SPS ausgleichen und macht die mühsame Abstimmung zwischen Roboterspezialisten und SPS-Programmierern überflüssig.



Aus der Sicht der Programmierung gibt es zwei Teams: Roboterprogrammierer und SPS-Programmierer. Befindet sich das Roboter-Know-how nicht im Haus wird hierfür Fremdpersonal eingekauft. Beide Teams entwickeln für ihre Bereiche die eigentlichen Ablaufprogramme sowie die Mensch-Maschinen-Schnittstellen, also die Visualisierungen, mit denen letztendlich die Anlagenabschnitte bedient werden. Verbunden über den Feldbus werden Schnittstellen mit Handshakes vereinbart.

Probleme bei der Instandhaltung heute

Trotz hoher Verfügbarkeit der Anlagen kommt es hin und wieder zu Störungen und damit zu Anlagenstillständen. In der Regel besitzt der Endkunde ebenfalls kein Roboter-Know-how, oftmals aber ein Instandhaltungsteam mit SPS-Programmiererfahrung. Ist der Anlagenbediener nicht in der Lage, den Fehler alleine zu lokalisieren und damit die Störung zu beseitigen, wird die firmeninterne Instandhaltung zu Hilfe gerufen. Routiniert wird das Programmiergerät an die SPS angeschlossen und versucht, den Fehler zu debuggen. Führt das Debugging der SPS nicht zum Erfolg, muss ein Roboterspezialist geordert werden. Erst dann kann die Ursache in Verbindung mit dem Roboter gefunden werden, was unter Umständen einen längeren Anlagen- und damit Produktionsstillstand nach sich zieht. Oftmals wird die Suche nach der Fehlerursache durch gegenseitige Schuldzuweisungen

behindert. Nicht nur während der Projektierungs-, der Programmier- und der Inbetriebnahme kommt es ggf. zu Reibungspunkten. Auch in der Zeit der Gewährleistung können getrennte Teams zu Mehraufwänden führen, die letztendlich den Gewinn schmälern - sowohl auf der Seite des Anlagenlieferanten, als auch beim Endkunden selbst wegen Produktionsausfällen. Hat der Anlagenlieferant einen passenden Partner mit Roboter-Know-how gefunden, der auch im Supportfall gut mit ihm zusammenarbeitet, führt eine weitere Kundenanforderung zu neuen Herausforderungen: Kunden verlangen mehr Flexibilität bei der Auswahl des Roboterherstellers. Der Anlagenlieferant hat beispielsweise die Partnerfirma beauftragt, für seine Lösung ein Roboterprogramm für einen Kuka-Roboter zu schreiben. Nun fordert der Kunde für eine nahezu identische Anlage z.B. einen Stäubli- oder einen Yaskawa-Roboter. Der Kuka-Roboter wird klassisch mit Kuka-KRL programmiert, während der Stäubli mit VAL 3 und der Yaskawa mit Inform III programmiert wird. Dabei unterscheiden sich schon die Entwicklungsumgebungen erheblich.

Roboterintegration morgen

Nun ist der Anlagenlieferant also in der Lage, in seiner gewohnten SPS-Entwicklungsumgebung mithilfe der Roboter-Befehls-Bibliotheken den Roboter selbst zu programmieren. Bei der Planung (Auswahl des richtigen Roboters) sowie bei der Grundinbetriebnahme (Installation Robotersoftware, Einrichten der Arbeitsräume, Vermessen der Werkzeuge und der Bases) wird er noch Unterstützung benötigen. Dieser Aufwand ist jedoch sowohl zeitlich als auch von den Kosten her gesehen ein relativ geringer Teil im Vergleich zum Gesamtaufwand. Die gesamte Programmierung nimmt der Anlagenlieferant selbst vor. Unterschiedliche HMI-Programmierungen sowie das Programmieren der Schnittstellen zwischen Roboter und SPS entfallen. Mit SPS-Robotik ist nur noch ein Programmiererteam nötig. Sowohl bei der Programmierung als auch bei der Inbetriebnahme vor Ort gibt es keine Abstimmungsprobleme zwischen Roboter- und SPS-Programmierer mehr. Nach der Inbetriebnahme ist der Anlagenlieferant selbst in der Lage, die gesamte Anlage, samt Roboter, selbst zu supporten. Da der SPS-Programmierer nun auf Handshakes mit dem Roboterprogramm verzichten kann, hat er die Möglichkeit, den Roboter auf völlig andere Weise in seinen Ablauf integrieren. Er behandelt nun selbst mehrere Roboter verschiedener Hersteller gleichzeitig in der SPS, wie den Rest der Anlage.

Probleme bei der Instandhaltung morgen

In der Nachtschicht bleibt die Anlage nun dennoch stehen, der Anlagenbediener findet den Fehler nicht und ordert die Instandhaltung. Der Instandhaltungsmitarbeiter verbindet sich mit seinem SPS-Programmiergerät und ist nun in der Lage, sowohl das Roboterprogramm, als auch die gesamte Peripherie mit seiner SPS-Entwicklungsumgebung zu debuggen. Umso schneller kann die Produktion weiterarbeiten.

Wie funktioniert die SPS-Robotik?

Der größte Irrtum ist die Annahme, dass SPS-Programmierer sich Gedanken über die Planung der Bahn machen und diese selbst programmieren müssen. Ohne tiefere mathematische Kenntnisse ist das beim Sechssachs-Roboter für den normalen Programmierer nicht möglich. Die Intelligenz des Roboters bleibt in der Robotersteuerung bestehen. Lediglich Befehle werden an die Steuerung gesendet, dort übersetzt und auch ausgeführt. Damit SPS und Robotersteuerung kommunizieren können, sind diese über einen Feldbus verbunden. Beispielsweise nutzen hier Kuka und Stäubli die Protokolle Ethercat, Profinet und Enternet/IP. Über den Feldbus werden sogenannte Befehlskommando-IDs mit Parametern (Roboterindex, Position, Geschwindigkeit, Beschleunigung) von der SPS an den Roboter übermittelt.

Verketten von Roboterbewegungen

Ein Roboterablauf besteht im Normalfall aus mehreren Positionen, bzw. mehreren Bewegungen. Wird jeder einzelne Befehl linear angefahren und mit der Befehlsübergabe der nächsten Bewegung gewartet, bis die aktuelle beendet wurde, entsteht ein ruckeliger und zackiger Ablauf. Um die Bewegung flüssiger zu machen, kommen a) wenn möglich PTP-Bewegungen zum Einsatz, b) gilt es, Punkte zu überschleifen und c) Befehle zu verketten. Das Überschleifen von Punkten bedeutet, dass Anwender, z.B. bei einem Stützpunkt zur Kollisionsvermeidung, den Punkt nicht exakt anfahren, sondern ihn lediglich als Anhalt für die Bewegung nutzen. Dann fährt dieser mit definiertem Abstand einen Bogen. Dadurch entstehen nicht nur flüssigere Bewegungen (runder Bewegungsablauf), sondern diese lassen sich auch wesentlich schneller fahren. Eine Befehlsverkettung bedeutet, die Befehle in einen Befehlsbuffer des Roboters zu schreiben. Das ermöglicht es ihm, das Überschleifen im Vorfeld einer Bewegung zu berechnen. Um zwei Befehle miteinander zu verketten, wird der Execute-Eingang des nachfolgenden Befehls mit dem Busy-Ausgang des ersten Befehls verknüpft.

SPS-Robotik und Anlagenvisualisierung

Grundsätzlich erlaubt es die SPS-Robotik, exakt dieselben Werkzeuge für die Visualisierung zu verwenden, die sonst für das SPS-Programm zum Einsatz kommen. So ist es denkbar, bei Verwendung einer Siemens-SPS die Anlage beispielsweise mit WinCC zu visualisieren, um damit den Roboter und die Anlage zu bedienen. Sämtliche Statusinformationen des Roboters, wie Positionsangaben, Fehler oder I/O-Zustände lassen sich via OPC DA, OPC UA oder andere Methoden auf beliebigen, gängigen Clients anzeigen.

Fotostrecke

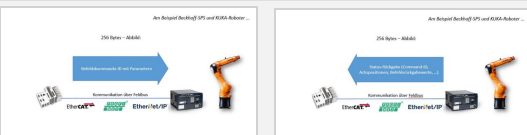


Bild 2: Bilder: Glaess - Software & Automation GmbH
Bild 3: Glaess - Software & Automation GmbH

Fazit

Wie lässt sich also ein Roboter in die Bedienung einer Bearbeitungsmaschine integrieren? Grundsätzlich gilt: Achsen lassen sich nur in der Betriebsart 'Automatik Extern' verfahren. Um ein externes Panel zu verwenden, sind ggf. roboterherstellerspezifische Mechanismen zu beachten.

www.glaess-software.de

Beliebte Themen

Simatic S7-1200: Das neue...
Moderne Technologie und m...
Bewegung optimal steuern...
Automatisierte Heizkrafta...
Mehr Produktivität - meh...

Andere TeDo-Seiten

www.i-need.de
GEBÄUDEDIGITAL
inVision
embeddedDesign
IT&Production

Weitere Informationen

Über TeDo
Stellenangebote
Für Autoren des SPS-MAGAZINS
Mediadaten
Sudoku



[Datenschutz](#)

[Rechtliche Hinweise](#)

[AGB](#)

[Impressum](#)

[Kontakt](#)



© 2017 TeDo-Verlag GmbH. Alle Rechte vorbehalten.