

SPS MAGAZIN

Zeitschrift für Automatisierungstechnik

B&Rs Orange Box schafft Durchblick

Industrial IoT für Bestandsanlagen

Highlights

- 18 Interview mit Dr. Jan Mrosik, CEO, Digital Factory, Siemens
- 64 Schwerpunkt Steckverbinder
- 84 Schwerpunkt Dezentrale Antriebstechnik

Marktübersichten

- 43 Industriecomputer
- 90 Schrittmotoren
- 107 Sicherheitslichtgitter
- 122 Durchfluss-/Strömungsmesssysteme

Produktübersichten

- 48 Codesys-Produkte
- 62 Display, Touchscreens, Tastaturen und Peripherieprodukte

Produktneuheiten

ab Seite 21



Bild: Systeme Helmholz GmbH



Wie Betreiber auf die Qualität von SPS-Bausteinen achten können

Vorgefertigte Bausteine: Getestet, erprobt und dokumentiert

Die meisten der heutigen Automatisierungssysteme bieten mittlerweile eine Programmiersprache nach IEC61131-3 an. Hierin wird allerdings nur die Programmiersprache geregelt und nicht der eigentliche Aufbau sowie die Struktur eines Automatisierungsprogramms. Bei der Erstellung der eigentlichen Funktionen hat der Programmierer immer noch alle Freiheiten, sodass unter Umständen auch Programmcodes entstehen können, die nur noch der Ersteller selbst versteht.

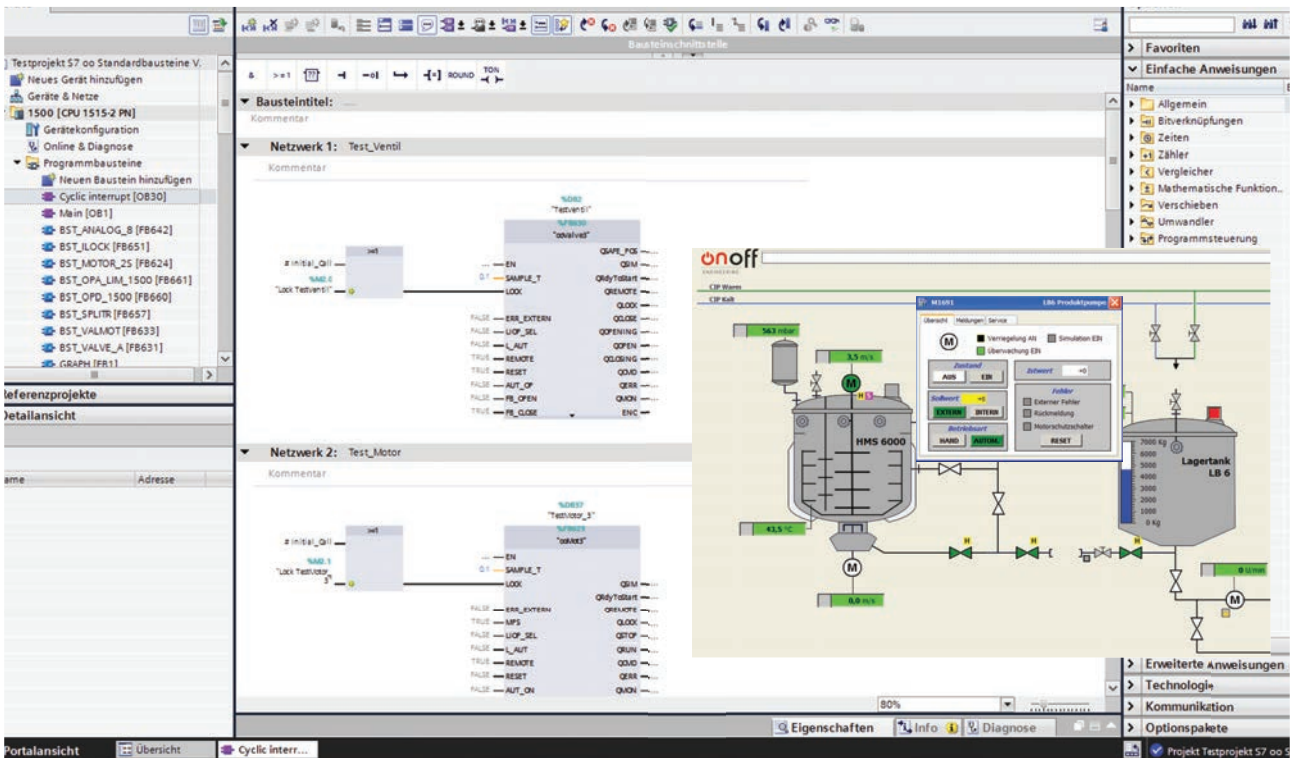
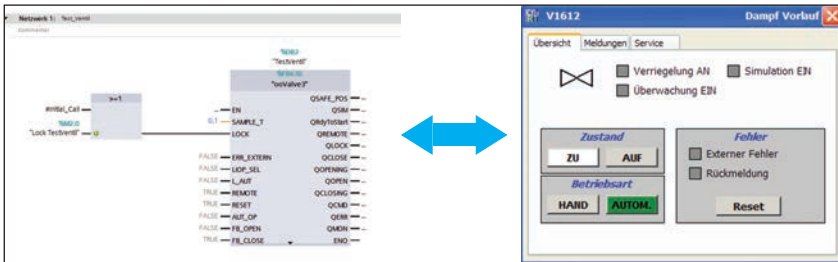


Bild: On/Off Engineering GmbH

Man muss Rad nicht immer neu erfinden, heißt es. Für einen Anlagenbetreiber bedeutet dies, dass es durchaus Sinn macht, auf Bausteinfunktionen eines Systemintegrators zurückzugreifen, die dieser durch seine jahrelange Erfahrung durch verschiedenste Einsatzfälle entwickelt und perfektioniert hat. Um sie in verschiedenen Anlagen wiederverwenden zu können, dürfen die Bausteine nicht zu groß sein (z.B. zur Steuerung einer ganzen Maschine), sie dürfen aber auch nicht zu klein sein, da sonst zu viel nachgearbeitet werden muss. In der Praxis hat es sich als sinnvoll erwiesen, für jeden Typ von Aktor und Sensor (z.B. Motor, Ventil, Regler usw.) einen eigenen Baustein zu verwenden. Diese können dann in der jeweiligen Applikation miteinander verschaltet werden. In den Standardbausteinen sind evtl. zwar Funktionen enthalten, die für diesen speziellen Einsatzfall nicht benötigt werden, vielleicht kommen jedoch in Zukunft Anforderungen hinzu, die dann bereits in den Bausteinen implementiert sind. Hilfreich können auch Bausteine sein, bei denen sich

Funktionen durch einfache Parameter zu- oder abschalten lassen. Das Augenmerk auf die Verwendung von Standardbausteinen sollte aber nicht nur in der SPS liegen. Heute gehört zu jeder Anlage auch ein Bedien- und Beobachtungssystem (HMI). Deshalb ist es sinnvoll, dass es zu jedem Standardbaustein in der SPS auch die passenden Bildsymbole, Meldungen und Bedienbilder für das HMI gibt, deren Schnittstelle standardisiert und einfach implementiert ist. Der Funktionsumfang steht für viele Betreiber natürlich im Vordergrund. Es sollte aber zusätzlich darauf geachtet werden, dass diese Bausteinmodule auch über einen Softwaremodultest beim Lieferanten nach gewissen Qualitätskriterien vorab getestet wurden. Dies ist im Pharma-Umfeld bereits im GAMP verankert, sollte aber durchaus in jeder Branche zur Anwendung kommen. Im Zweifel sollte man sich einfach die Testprotokolle der Inhouse-Tests vom Lieferanten vorab zeigen lassen, um bewerten zu können, wie professionell der Lieferant mit der Qualität der Softwareerstellung umgeht.



Zu jedem SPS-Standardbaustein gehört auch ein passendes Bedienbild im HMI.

Keine schreibgeschützten Bausteine

Bei vielen Lieferanten ist die Aktivierung eines Know-How-Schutz auf den Programmcode sehr beliebt. Damit kann nur der Lieferant selbst Änderungen oder Erweiterungen an den Programmen vornehmen. Der Endkunde hat dadurch auch keine Möglichkeit mehr, seinen Lieferanten zu wechseln, da ein möglicher neuer Lieferant verständlicherweise keine Verantwortung für die korrekte Funktion der vorhandenen, schreibgeschützten Programmteile übernehmen kann. Deshalb ist bereits im Vorfeld der Vergabe von Automatisierungsprojekten darauf zu achten, dass der Lieferant alle Programmteile mit kommentiertem Quellcode ausliefert.

Einfach und verständlich

In vielen frei erstellten Automatisierungsprogrammen findet man immer wieder einen komplexen Code ohne Struktur und Kommentierung, so dass man meinen könnte, der Ersteller wollte hier sein eigenes 'Kündigungsschutzprogramm' verewigen. Das geht oft sogar soweit, dass nicht einmal die Kollegen des Programmierers aus derselben Firma die Funktionen nachvollziehen können und so eine Abhängigkeit von einer einzelnen Person entstehen kann. Sobald diese Person nicht mehr verfügbar ist, haben sowohl Betreiber als auch der Anlagenbauer selbst ein großes Problem. Damit das Wartungspersonal des Anlagenbetreibers mit den Programmen umgehen kann, um z.B. Fehler in der Anlage zu diagnostizieren oder einfache Erweiterungen selbst durchzuführen, ist es unerlässlich, dass alle Programme in den SPS-Steuerungen klar strukturiert und gut verständlich kommentiert sind, auch die der verwendeten Standardbausteine.

Vorteile für alle

Nicht nur der Lieferant profitiert von der Verwendung von Standardbausteinen, in dem er die Funktionen durch seine Erfahrung immer weiter verfeinert und dies unabhängig von einzelnen Mitarbeitern in die Projekte einfließen lassen kann. Auch der Endkunde hat wesentliche Vorteile durch die Standardbausteine, weil er vorgetestete und praxiserprobte Bausteine bekommt. Durch eine gute Dokumentation ist das eigene Wartungspersonal auch selbst in der Lage, Änderungen

oder Erweiterungen an der Anlage vorzunehmen. So entsteht keine Abhängigkeit zum Lieferanten, die über den Lebenszyklus einer Anlage hinweg gesehen evtl. auch teuer werden könnte.

Fazit

Bei der Verwendung von speziell entwickelten SPS-Bausteinen entsteht für den Anlagenbetreiber eine starke Abhängigkeit zum Lieferanten der Anlage, teilweise sogar zum einzelnen Mitarbeiter dieser Firma. Abhilfe schafft nur die Verwendung von getesteten und dokumentierten Standardbausteinen. Dies hat Vorteile sowohl für den Anlagenbauer als auch für den Betreiber, der so Programm-

Bild: On/Off Engineering GmbH

```

63 // Vorortbetrieb / Fernbetrieb
64 IF (NOT #REMOTE AND #xREMOTE_OLD) THEN // Abfrage Vorortbetrieb
65   #xAUT_Save := #xAUT; // Sichern Betriebsart HAND / AUTOMATIK
66   #xREMOTE := #REMOTE; // FALSE = Vorortbetrieb
67   #xAUT := FALSE; // Automatik deaktivieren bei Vorortbetrieb
68   #xSP_LOC := 0.0; // Ortsollwert auf Null setzen
69   #xCMD := FALSE; // Motor ausschalten

```

Beispiel eines gut kommentierten Programmcodes

bausteine erhält, die wie ein Baukastensystem als Softwaremodule miteinander zu einem großen komplexen Programm verschaltet werden können. Diagnose und Anpassungen im Lebenszyklus der Anlage sind dann auch für den Betreiber kein Problem mehr. ■

Autor: Jürgen Flütter,
Vertrieb,
On/Off Engineering GmbH
www.onoff-group.de

Direkt zur Marktübersicht i-need.de

www.i-need.de/?f15663