

SPS MAGAZIN

Zeitschrift für Automatisierungstechnik

Lasal 'All-in-one': Software für durchgängige Automatisierung

(S.8)

Embedded Hardware
(S.35)

Kleine &
kompakte SPSen
(S.75)

Elektronikgehäuse und
Schaltschränke –
19"-Systeme (S.86)

Fernwirken und
Fernwarten
(S.94)

PC-basierte
Steuerungen
(S.126)

Beleuchtungen für die
Industrielle
Bildverarbeitung (S.112)

Sicherheitsbus-
Produkte
(S.137)

embeddedworld2008
Exhibition & Conference
(ab S.29)

Schneideklemmtechnik sorgt für
sichere Verbindungen:

Schnell Anschluss finden

(S.83)

Software für durchgängige Automatisierung



Bild 1: Mit Lasal Class, Lasal Screen, Lasal Service und Lasal Motion stellt das Lasal-Paket von Sigmatek eine durchgängige Plattform und eine einheitliche Umgebung für unterschiedliche Aspekte der Automatisierungssoftware dar.

Durchgängig entwickelte Automatisierungslösungen, bei denen von der Projektierung bis zur einzelnen Komponente alles nahtlos ineinander passt, sind ein entscheidender Wettbewerbsvorteil im Maschinenbau. Hierfür bietet Sigmatek die objektorientierte Entwicklungsumgebung Lasal an. Für die Antriebstechnik bietet Lasal Motion Entwicklern die Möglichkeit, mit nur einer Software Bewegungsabläufe zu programmieren.

Für die Entwickler von Fertigungsmaschinen ist längst nicht mehr nur die Auswahl der optimalen Komponenten entscheidend. Angesichts immer kürzer werdender Entscheidungszeiträume für Großinvestitionen wurden Effizienz und Geschwindigkeit der Entwicklung kompletter Automatisierungslösungen ebenso zum Kriterium wie die Flexibilität und die Wirtschaftlichkeit der Lösung. Daher werden Maschinen und Anlagen möglichst modular aufgebaut, um durch flexible Ausstattung mit unterschiedlichen Optionen rasch auf individuelle Anforderungen reagieren zu können. Außerdem führte der Wettbewerbsdruck zu einer Abkehr der Entwickler von

der gewohnten sequenziellen Software-Erstellung. Nur so können sie auch bei unterschiedlicher Ausstattung eine große Durchgängigkeit der Lösung von der Projektierung der Gesamtanlage bis hinunter in die kleinste Komponente gewährleisten. Schließlich will der Anwender nicht eine Maschine, sondern eine Lösung für eine bestimmte Fertigungsproblematik – mit einem Ergebnis, das bereits zum Zeitpunkt der Auftragserteilung in überprüfbarer Form feststeht.

Modularität als Prinzip

Sigmatek brachte in diesem Bereich vor einigen Jahren als Erster eine Projektierungs- und Entwick-

lungssoftware auf den Markt, die mit objektorientierter, grafischer Programmierung und modularem Aufbau die Software-Entwicklung in der Steuerungstechnik veränderte: Lasal. Die mittlerweile verfügbare zweite Version von Lasal Class trägt mit Übersichtlichkeit und Durchgängigkeit zur Effizienzsteigerung in der Software-Entwicklung bei und erhöht zudem die Verarbeitungsgeschwindigkeit. Wie in einem Baukastensystem werden dabei, vom Gesamtprojekt ausgehend, die einzelnen Funktionen hierarchisch gruppiert und verbunden. So entsteht eine durchgängige Modularität, die für Übersichtlichkeit sorgt und den Anwender zwingt, seine Software-Entwicklung zu

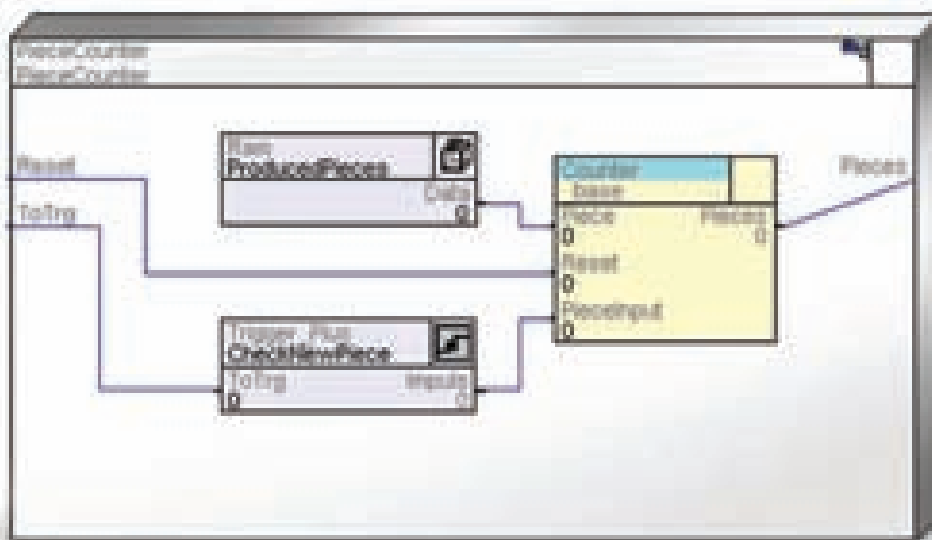


Bild 2: Klassen und Objekte sind in beliebiger Tiefe hierarchisch aufgebaut und vererben ihre Eigenschaften auf die nächste Ebene. Objekte in einem eventgetriggerten Client-Server-System sind nur aktiv, wenn sie durch äußere Ereignisse angestoßen werden und sparen dadurch CPU-Kapazität.

strukturieren. Wie die Hardware kann auch die Software durch unterschiedliche Ausstattung mit Modulen den jeweiligen Erfordernissen angepasst werden. Lasal beruht auf dem Prinzip von Software-Klassen. Dabei handelt es sich um in sich abgeschlossene (gekapselte) Programmbausteine für unterschiedliche Funktionalitäten. Abgelegt in sogenannten Klassenbibliotheken, die herstellerseitig mit gängigen Funktionsklassen gefüllt sind, können sie in unterschiedlichen Projekten oder Systemteilen einfach verwendet werden. Klassen enthalten den grundlegenden Programmcode der jeweiligen Funktion, wobei zum Aufbau eigener Klassen als Programmiergrundlage Structured Text (ST) ebenso verwendet werden kann wie Anweisungsliste (AWL) oder Kontaktplan (KOP) – alle drei nach IEC61131-3 – oder direkte Programmierung in ANSI-C. Zu (Programm-)Objekten werden die Klassen durch individuelle Parametrierung und durch Verbindung mit anderen Objekten. Dadurch erhält die grundsätzliche Funktion der Klasse ein definiertes, eigenständiges Verhalten. Mit der Möglichkeit zur 'Vererbung' von Klasseigenschaften kann jede hierarchische Programmebene die Eigenschaften der untersten Ebene nutzen, aber auch

verändern. Einmal getestete Objekte können wie Lego-Steine zu komplexen Programmstrukturen zusammengefügt werden, ohne in den Programmcode der untersten Ebene eingreifen zu müssen. So können etwa bei Inbetriebnahme

und Einstellung einer Anlage durch Parametrierung und Verknüpfung der verwendeten Objekte gefahrlos Anpassungen durchgeführt werden. Die einzelnen Softwarebausteine kommunizieren nach dem Prinzip von

Client und Server. Der Client als aktiver Teil der Kommunikation fordert vom Server die Information an, die dieser passiv zur Verfügung stellt. Dadurch kann im Vergleich zu konventionellen Systemen CPU-Leistung einge-



Bild 3: Lasal unterstützt Software-Entwickler mit zahlreichen Analyse-Tools.



Bild 4: Sigmatek-Applikationist Karl Baldauf parametriert Bewegungsabläufe.

spart werden, weil Bausteine nur dann aktiv werden, wenn sie durch ein äußeres Ereignis angestoßen werden.

Top-Down-Entwicklung

Lasal Class gibt Entwicklern die Möglichkeit, die Automatisierungssoftware 'von oben nach unten' zu entwickeln. Durch Entwurf der gesamten Automatisierungslösung in der Projektierungsphase entsteht ein Rahmen aus fertig beschriebenen Funktionalitäten, die später mit ihrer jeweiligen Detailprogrammierung befüllt werden können. Damit werden wesentliche Teile der Systementwicklung bereits in der Projektierung mit erledigt. Das eröffnet nicht nur Möglichkeiten zur frühzeitigen Überprüfung des späteren Ergebnisses, sondern hilft auch, wertvolle Zeit im Entwicklungsprozess zu sparen. Aus der in der Projektierung erstellten Baumstruktur mit vorhandenen oder noch leeren, aber definierten

Klassen leitet Lasal selbstständig Klassendefinitionen als Vorgabe für die einzelnen Entwickler ab. Dabei kann auf eine Fülle mitgelieferter Bibliotheken mit gebrauchsfertigen Klassen für Betriebssystemfunktionen, Hardware-Zugriffe und Standardklassen wie PID-Regler oder Datenlogger zurückgegriffen werden. Mithilfe der standardisierten Scriptsprache Python können aus fertigen Bausteinen vollautomatisch Projekte erstellt oder Anwendungen parametriert werden. Das ermöglicht das Generieren abgestimmter Software-Versionen für unterschiedliche Ausstattungsvarianten oder Ausprägungen für kundenspezifisch angepasste Serienmaschinen. Eine weitere Verkürzung der Entwicklungszeit bringt die Simulation noch nicht fertiggestellter Systemteile mit dem Windows-basierten Simulationstool Lars. Lasal stellt dem Entwickler unterschiedliche Möglichkeiten zur Analyse der fertigen Programme,

zur Fehlersuche und -korrektur zur Verfügung. Beispiele dafür sind ein Online-Debugger, ein Oszilloskop und ein Analyse-Tool für das Zeitverhalten im Multitasking.

Motion under Control

In der Antriebstechnik sorgt Lasal Motion für mehr Drive. Als eines von vier anwendungsspezifischen Software-Modulen deckt es die Antriebstechnik ab. Dazu enthält die Bibliothek Klassen für eine riesige Auswahl an ein- oder mehrachsigen Bewegungsabläufen und anderen Motion-Funktionen. Zusätzlich sind zahlreiche Drives, sowohl von Sigmatek als auch Fremdfabrikate, bereits als Objekte zur einfachen Verwendung hinterlegt. Interessant ist die Möglichkeit, die Dias-Drives von Sigmatek über Lasal Class und die Applikation zu parametrieren. Diese Funktion eliminiert nicht nur den Eingriff direkt am Gerät bei Tausch oder Umplanung, sie

ist auch der Schlüssel zu einer durchgängigen Datenhaltung über alle Teile eines Projektes hinweg. Aufgrund dieser Durchgängigkeit bleibt die Kontrolle über alle Aspekte eines Projektes vollständig in der Hand der Entwickler, was die Effizienz und Umsetzungsgeschwindigkeit bei Neuprojekten verbessert und Fehler, die sich in der dezentralen Wartung einschleichen können, verhindert. Ein weiterer Vorteil für Antriebstechnik-Entwickler ist die Möglichkeit, Bahnbewegungen zu definieren und die Umsetzung in Einzelachsbewegungen den vorgegebenen Bewegungsklassen zu überlassen. Und schließlich muss sich der Entwickler durch die optimale Systemintegration nicht um Details kümmern, sondern kann sich auf das Ergebnis konzentrieren.



Autor: Ing. Bernhard Gangl, Abteilungsleiter Software bei der Sigmatek GmbH & Co KG in Lamprechtshausen.



Autor: Dipl.-Ing. (FH) Andreas Rauhofer ist Produktmanager Antriebstechnik bei der Sigmatek GmbH & Co KG in Lamprechtshausen.

www.sigmatek.at