

Einführung und Übersicht

Vorwort

Thema dieses Buchs soll eine Form der Programmierung von Automatisierungssystemen sein, die textbasiert ist. Dabei steht der „Strukturierte Text“ (ST) bzw. SCL (Structured Control Language, Siemens) im Fokus. Doch ein Buch, das nur theoretische Ansätze beschreibt, wäre unanschaulich und wohl wenig hilfreich, daher sind zahlreiche Beispiele und Übungen integriert, die den Einstieg und Umstieg von anderen Programmierformen unterstützen.

Eine besondere Herausforderung war die Definition des Buchtitels und des Buchinhalts dieses Buchs. Ein reines Buch über ST wäre zu kurz gegriffen, da Siemens als führendes Unternehmen in der Ausbildung als auch bei der Anwendung mit dem SCL ein nicht zu vernachlässigendes Thema vorgibt. Zusätzlich ist das Umfeld des Strukturierten Texts zu berücksichtigen. Viele Anwender werden aus dem Bereich der Sprachen „C“ oder anderer textbasierter Programmierungen kommen, Einsteiger (Techniker, Studenten, interessierte Schüler) haben möglicherweise bereits Erfahrungen mit C-Control, Arduino, Raspberry Pi u. v. m. und werden nun einen Weg in den ST suchen. Daher stand alternativ der Titel „Textbasierte Programmierung“ oder „Skript-Sprachen“ im Raum.

Neben der Programmierung sind bekannte Steuerungen wichtig, auf denen die im Buch beschriebenen Themen angewendet werden können, die Hersteller dieser Steuerungen und die unterschiedlichen Unternehmen, die die entsprechenden Editoren entwickeln, mit denen die Programme für die jeweiligen Steuerungen erstellt und kompiliert werden.

Hier lag auch eine besondere Anforderung an dieses Buch: Genügend Detail- und Praxisnähe ohne dabei das eigentliche Thema „Strukturierter Text“ aus dem Auge zu verlieren.

Neben der Programmiersprache, dem Sprachaufbau, den Sprachelementen und der Syntax ist natürlich das Werkzeug wichtig, mit dem die Programmierung und alle anderen Aktivitäten durchgeführt werden können. Früher konnte man derartige Software als „Programmierung“ bezeichnen, zwischenzeitlich gehen die Funktionen einer entsprechenden Software weit über die reine Programmierung hinaus.

Das Leistungsspektrum moderner Automatisierungstools umfasst:

- Hardwarekonfiguration,
- Netzwerkkonfiguration,
- Einstellungen der Netzwerkverbindungen,
- Programmierung der einzelnen Steuerung,
- Erstellung von Visualisierungen,
- Dokumentationserstellung,
- Versionsverwaltung,
- Archivierung.

Siemens hat bei dem Begriff „TIA Portal“ (Totally Integrated Automation) für seine Software dieser Sichtweise Rechnung getragen. Andere Unternehmen sprechen von einem „Studio“. Der Autor suchte nach einem übergreifenden Begriff und entschied sich für „**IDE**“. Dieser Begriff ist eindeutig und wird beim „Suchen“ im Internet als auch bei Wikipedia sicher erkannt.

IDE ist ein Akronym für „integrated development environment“; auch als Untermenge „integrated design environment“ oder „integrated debugging environment“ interpretiert.

Anbieter von Editoren (IDEs)

Für den Bereich der Automatisierung werden in diesem Buch folgende Anbieter von Programmiersystemen insbesondere behandelt:

- | | | |
|----------------------------------|---|---------------------------|
| • 3S/Kempton | → | Codesys |
| • Phoenix Contact Software/Lemgo | → | Multiprog/PC Worx Express |
| • Siemens | → | TIA Portal |
| • logi.cals/St. Pölten | → | logi.CAD 3 |
| • Pilz/Ostfildern | → | PAS4000 |
| • Conrad Electronic | → | C-Control |

Damit sind nur die Softwareunternehmen genannt, die im Weiteren des Buchs beschrieben werden und offen für unterschiedliche Hersteller Programmiersysteme erstellen, erweitern und anpassen. Tatsächlich gibt es weltweit zahlreiche Softwareunternehmen, die für weitere Anbieter von Automatisierungsgeräten entsprechende Programmierwerkzeuge auf Basis des Strukturierten Texts (ST) erstellen. Die Auswahl verdeutlicht aber auch, dass sich dieses Buch beschränken musste, um nicht zu umfangreich und damit unübersichtlich zu werden.

Andererseits wird der Leser erfahren, welche Bedeutung der „Strukturierte Text“ in den verschiedenen Bereichen der Automatisierung (Gebäude, Industrie, Verfahrenstechnik, ...) erhält und wird gleichzeitig erkennen, dass es sehr unterschiedliche Steuerungen und Lösungsansätze (RunTimeSysteme für PCs) gibt, verbunden mit niedrigen Kosten (z. B. Raspberry Pi).

Dieses Buch dient auch als Umstiegshilfe für die Automatisierungsfachleute, die seit vielen Jahren, vielleicht Jahrzehnten mit der Anweisungsliste (AWL) erfolgreich gearbeitet haben. Hier wird sicher ein Umbruch in der Automatisierung stattfinden und dieses Buch soll eine sichere Hilfe bei den notwendigen Neuausrichtungen sein. Mit der S7-1200 des Unternehmens Siemens wird dieses Thema für die Ausbildung und die unteren Ebenen der Automatisierung ein „brennendes Thema“.

Gleichzeitig wendet sich dieses Buch an Automatisierungsfachleute, Projektleiter, Studenten, Techniker, aber auch an interessierte Programmierer des Elektrohandwerks, sowie Auszubildende in der Mechatronik und die entsprechenden Schulungseinrichtungen.

Zusätzlich kann es für Projektleiter ein wichtiges Orientierungsmittel sein, wie die Erstellung der Software zu realisieren ist. Die Wahl der jeweiligen Programmiersprache sollte nicht von den persönlichen Ansichten des Programmierers abhängen, sondern von den jeweiligen Nutzern in den verschiedenen „Lebensphasen“ einer Software.

Vorwort zur 2. Auflage

In der neuen Auflage tritt einerseits der Raspberry Pi mehr in den Fokus, andererseits werden Ausführungen, die sehr stark auf Siemens-Produkte gerichtet waren (bisher Kapitel 8), Teil eines neuen Buchs zum Thema SCL. Die stärkere Berücksichtigung des Raspberry Pi und linuxbasierter Automatisierungssysteme (wie Revolution Pi – siehe Kapitel 8) bildet auch eine neue Brücke zu anderen Hochsprachen (wie Python 2 bzw. 3) und Möglichkeiten, notwendige Skripte zu verfassen (wie Bash-Skripting).

Mit den neuen Anforderungen, z. B. des Raspberry Pi, rückt nun das Codesys 3.5 stärker in den Vordergrund. Entsprechend werden im Kapitel 11 Hinweise auf die Möglichkeiten und Vorgehensweisen der neueren Version des Codesys gegeben (in der 1. Auflage wurde bewusst ausschließlich Version 2.3 berücksichtigt).

Verstärkt wurden außerdem Konzepte zum Testen, Entwickeln, Trainieren und Anwenden eingearbeitet, bei denen man sehr kostengünstig mit Demo-Versionen arbeiten

kann. Wenn die Lösung später praktisch angewendet werden soll, ist der Anwender natürlich frei in der Auswahl der Hardware.

Einleitung

Unter Skript-Sprachen sollen in diesem Buch die Programmiersprachen bezeichnet werden, die „textbasiert“ und damit nicht grafisch sind.

Zu diesen Programmiermitteln (Skript-Programmierung) wurde in der Informatik stets gegriffen, um eigene Funktionen, Abläufe bzw. Makros zu erstellen. In der Automatisierung ist diese Form der Realisierung der Funktionserweiterung insbesondere bei leistungsfähigen Visualisierungsgeräten sehr früh angewendet worden.

Bedeutung für die Kleinststeuerungen

Aufbauend auf dem Thema „Kleinststeuerungen“ wird der Bedarf an einem Werkzeug zur Entwicklung leistungsfähiger, eigener Anwendungsfunktionen immer dringlicher. Zwar bieten einige Hersteller bereits die Möglichkeit, eigene Funktionen zu erstellen, doch kann zu diesem Zweck nur der sehr begrenzte Sprachvorrat der jeweiligen Steuerung genutzt werden.

Mit diesem Buch soll der zukünftige Entwicklungsweg aufgezeigt werden, eigene Bausteine mittels einer textbasierten Programmiersprache (Basic, C-Dialekte, ST (Strukturierter Text) oder SCL (Structured Control Language)) zu erstellen oder vollständige Automatisierungsaufgaben mit diesen Programmiersprachen zu lösen.

Allgemeiner Trend in der Automatisierung

Allgemein ist der Weg in der Automatisierung weg von der Anweisungsliste (AWL) als textbasierte Programmiersprache vorgegeben. Mit der Einführung des „SCL“ (Structured Control Language) statt der AWL für die S7-1200 (die von der Baugröße und dem Preis einer Kleinststeuerung nahe kommt) hat das Unternehmen Siemens diese Richtung wohl endgültig fixiert. Damit sollte auch in der Ausbildung und Anwendung kurzfristig entsprechend reagiert werden, gleichzeitig sollten sich auch erfahrene Programmierer die notwendigen Kenntnisse erarbeiten. Möglichkeiten bieten Kurse der einschlägigen Hersteller, herstellerneutrale Ausbildungseinrichtungen oder dieses Buch.

Zweck des Buchs

Dieses Buch soll nicht nur eine Hilfe zum Einstieg in die unterschiedlichen textbasierten Programmiersprachen bzw. Skript-Sprachen (Basic- oder C-Dialekte, Strukturierter Text (ST) bzw. SCL (Siemens)) geben, sondern auch zu einem besseren Lösungsansatz von Steuerungsaufgaben führen.

Das Buch in der Ausbildung

In der Ausbildung kann dieses Buch Auszubildenden und Studierenden aller Fachrichtungen, die die Automatisierungstechnik anwenden oder bei der Planung ihrer Aufgaben auf moderne Steuerungstechnik zurückgreifen müssen, eine fundierte Unterstützung bieten.

Dieser Aspekt gewinnt umso mehr an Bedeutung als während der Erstellung dieses Buchs immer häufiger in den Ausbildungseinrichtungen der Mechatroniker die Forderung aufgestellt wird, für die Prüfungen Steuerungstechnik oberhalb der Kleinststeuerungen zu verwenden. Tatsächlich werden von zahlreichen Unternehmen Steuerungssysteme mit Hardwarepreisen im Bereich der oberen Kleinststeuerungen angeboten, zu denen kostenlose Programmierpakete erhältlich sind (z. B. von der Fa. Phoenix Contact). Diese Steuerungen bieten jeweils auch den Strukturierten Text oder SCL an.

Raspberry Pi und C-Control

In der Ausbildung sollten darüber hinaus die C-Control-Systeme von Conrad Electronic oder Raspberry Pi interessant sein, da mit diesen Systemen äußerst kostengünstig textbasierte Programmiersprachen (C, Basic oder ST) erlernt und in praktischen Anwendungen genutzt werden können.

„3 Phasen“-Strategie in der Ausbildung

Wünschenswert wäre aus meiner Sicht in der Ausbildung eine „3 Phasen“-Strategie.

- Phase 1 (Grafische Methode)
Erlernen einer statischen Logik (Kontaktplan, Logikplan)
- Phase 2 (Grafische Methode)
Erlernen von Methoden zur Darstellung und Programmierung kausaler Abläufe, Erkennen von kausalen Abläufen (Programmablaufplan (PAP) oder GRAFCET bzw. Ablaufsprache (AS) nach IEC 61131-3 SFC)

- Phase 3 (Textbasierte Programmiersprache)
Erlernen von C, Basic oder ST bzw. SCL zur Erstellung spezieller Technologiebausteine, die in die grafischen Sprachen eingebunden werden

Anwendung des Buchs für den erfahrenen Anwender

Doch auch Programmierer, die bisher die AWL (Anweisungsliste) verwendeten, finden mit diesem Buch einen Einstieg bzw. Umstieg in die moderne Form der Programmierung eines Automatisierungssystems unter Verwendung einer textbasierten Programmiersprache.

Die Verwendung von selbst erstellten und geprüften Funktionsbausteinen verkürzt und entlastet die Inbetriebnahme einer Automatisierungssteuerung und bietet die jeweils geeignete Programmiersprache für den jeweiligen Aufgabenteil.

Es gibt gute Gründe für ST bzw. SCL:

- Die Portierung auf neue Programmiersysteme oder andere Steuerungshersteller ist relativ einfach.
- Sie sind sehr effizient und kompakt, um komplexe Aufgaben zu beschreiben (z. B. Regelungstechnik, Datenaufbereitungen und Datenkontrollen, Berechnungen u. v. m.).
- Wissen und Erfahrungen aus anderen Skript-Sprachen sind intuitiv anwendbar.
- Es können beliebige Texteditoren verwendet werden.
- Gefährliche Programmstrukturen (Rücksprünge) werden vermieden.

Das bedeutet jedoch nicht, dass damit andere Programmiersprachen (Kontaktplan, Funktionsplan oder Ablaufsprache) überflüssig sind.

Der Fokus verändert sich

Die Festlegung von Bausteinen, den Ein-/Ausgabe-Variablen und der Funktionalität eines FBs oder eines FCs rückt in den Mittelpunkt der Arbeit des Programmierers, ebenso die Prüfung, Freigabe und Versionsverwaltung. Die reine Funktionalität tritt in den Hintergrund; die Prüfbarkeit, Versionsverwaltung und Anwendungshilfen werden stärker zum Qualitätsmaßstab. Die Definition von Daten und Datenstrukturen ist ein weiteres, wichtiges Qualitätsmerkmal zu einer langfristigen und stabilen Nutzung einer Lösung in der Programmierung von Steuerungssystemen, ebenso für die Erweiterungs- und Anpassungsfähigkeit einer Automatisierungslösung.

Gleichzeitig schließt sich der Kreis zu den Logikmodulen oder Steuerrelais (LOGO!, easy, Nanoline u. v. m.). Diese Steuerungen bieten dem Anwender die Möglichkeit, mit dem Kontaktplan bzw. Logikplan fertige Funktionsbausteine anzuwenden. Die Hersteller bieten bereits eine Vielzahl von leistungsfähigen Bausteinen, die unterschiedliche Anwendungstechnologien unterstützen. Dennoch besteht oft der Wunsch, derartige Bausteine selber zu entwickeln oder durch Spezialisten entwickeln zu lassen. Daher bieten einige Steuerungshersteller (wie Crouzet/Valance, Frankreich) bereits heute an, eigene Bausteine auf der Basis Kontaktplan oder Funktionsplan zu entwickeln. Der zukünftig sicherste Weg dürfte jedoch in der Anwendung einer textbasierten Sprache bei der Entwicklung von Technologiebausteinen für Kleinsteuerungen oder Smart-Relais liegen.

Bewertung – Risiken

Ein großer Vorteil des deutschen Handwerks und der Industrie war das Reservoir der gut ausgebildeten Facharbeiter und Meister. Diese Aussage galt ebenfalls für die Automatisierung und die entsprechende Programmierung von Steuerungssystemen. Nun ist es wichtig, dass durch entsprechende Qualifizierungen dieser Mitarbeitergruppe auch mit den ständig wachsenden Möglichkeiten der Programmiersprachen das Mitarbeiterumfeld erhalten bleibt. Bereits seit einigen Jahren verwenden größere Steuerungsbauer komplett den strukturierten Text auf unterschiedlichen Steuerungen, dabei entsteht aber gleichzeitig eine hohe „Expertenabhängigkeit“, die in den weiteren Lebenszyklen einer Steuerungsaufgabe nachteilig sein kann. Vor- Nachteile des ST bzw. SCL sollen daher kurz gegenübergestellt werden.

Vorteile:

- Übertragbarkeit der Quelle auf alle Steuerungen, die mittels ST oder SCL programmiert werden können,
- Erstellung des Quellcodes durch beliebige Texteditoren,
- effektives Arbeiten mittels [COPY][PASTE][SUCHEN][ERSETZEN].

Doch es gibt auch Risiken, die dadurch entstehen können, dass Ausbildungseinrichtungen nicht zeitnah diesen Stoff vermitteln und in den Unternehmen nicht durch entsprechende betriebsinterne Vorschriften reagiert wird.

Risiken:

- Schlechtere Nachvollziehbarkeit der Software,
- längere Inbetriebnahmezeiten,
- höhere Abhängigkeit von Spezialisten/größerer Bedarf an Spezialisten.

Ideale Programmierumgebung:

- Bausteine zur Lösung geschlossener Aufgaben werden in den Büros der Fachabteilungen definiert, im ST oder SCL realisiert, geprüft, verriegelt und freigegeben.
- Diese Bausteine werden im Funktionsplan angewendet.
- Logische Verriegelungen (Zuweisungen von Motoren, Pumpen, Leuchtmelder, ...) werden im Kontaktplan realisiert.
- Kausale Abläufe werden in GRAFCET, AS bzw. SFC programmiert. Als Hilfssprache zur Eingabe der Aktionen und Transitionen (Weiterschaltbedingungen) werden ST oder FUB bzw. KOP verwendet.

Eine besondere Aufgabe kommt dabei den Schulen und Ausbildungseinrichtungen zu. Sie müssen kurzfristig diese Programmierform (ST) im Unterrichtsstoff einbinden und Fortbildungskurse in den Verbänden anbieten. Neben Anderen sind in diesem Thema auch die Innungen der Industrie und des Handwerks, die Ausbildungseinrichtungen großer Unternehmen und weitere Schulungsstätten gefragt.

Die Sprache des Technikers und Ingenieurs ist die Zeichnung

Der Kernsatz jeder technischen Ausbildung wird durch die Skript-Programmierung in Frage gestellt: „Die Sprache des Technikers und Ingenieurs ist die Zeichnung“. Es werden zwar PAP und Struktogramme als Entwurfssprache angeboten, es gibt jedoch keine Compiler, die den Entwurf in einen „Strukturieren Text“ übersetzen könnten. Umso mehr ist die Ausbildung gefordert, entsprechende Entwurfs- bzw. Spezifikationsmittel im Unterricht zu vermitteln und anzuwenden. Darin liegt auch der Grund, dass die mittels ST erstellten Bausteine in den Beispielen dieses Buchs in KOP, FUP oder AS, SFC, GRAFCET angewendet werden.

Kapitelübersicht – Beschreibung

Kapitel 1 Einführung – Übersicht über die Programmiersprachen und die entsprechenden Hersteller

Programmiersprachen:

- ST
- SCL
- STL