1 Einführung in die Thematik und inhaltliche Übersicht zu den Kapiteln

In **Kapitel 1** geht es zunächst um die grundsätzliche Arbeit mit dem Siemens TIA Portal im Hinblick auf die SCL-Programmierung und um die unterschiedlichen CPU-Typen mit ihren spezifischen Hardwarekonfigurationen, die im TIA Portal verwendet werden können. Die unterschiedlichen Möglichkeiten zur Simulation von erstellten Anwenderprogrammen werden ausführlich dargestellt und die Inbetriebnahme einer realen CPU wird in diesem Kapitel kleinschrittig erklärt. Dieses Kapitel wurde durch die komplette Neuerstellung auf den aktuellen Stand – insbesondere der Hardware – gebracht.

Kapitel 2 stammt bis auf wenige Aktualisierungen komplett aus der ersten Auflage dieses Buchs von Herrn Ulrich Kanngießer mit einem detailliert dargestellten Schnelleinstieg zur Arbeit mit der CPU S7-1215C und ersten Programmbeispielen zur SCL-Programmierung.

In **Kapitel 3** geht es grundlegend um das Thema „Variablen in der Automatisierungstechnik" als Voraussetzung zum Umgang mit den unterschiedlichen Datentypen in der Projektierung von Automatisierungslösungen. Auch dieses Kapitel stammt mit Ausnahme einiger Aktualisierungen und Anpassungen meinerseits von Ulrich Kanngießer.

Der erste inhaltliche Schwerpunkt zur SCL-Programmierung erfolgt in dem sehr umfangreichen **Kapitel 4** mit Entwurfsmethoden zur Programmgestaltung und zur konkreten Programmierung von Grundverknüpfungen, Ausgängen mit Speicherverhalten sowie Zeit- und Zählfunktionen bis hin zu den Programmschleifen, die anhand von detailliert dargestellten Beispielen aus der Praxis ausführlich erklärt werden. Eine Leistungsüberwachung, die mit einer IF...THEN...ELSE-Anweisung projektiert wird, eine Fußgängerampel als zeitgeführte Ablaufsteuerung mit einer CASE...OF-Anweisung und die Artikelerkennung in einem Hochregallager mit der Anwendung der FOR...TO...BY...DO-Schleife gehören zu den dargestellten Projekten dieses von mir erweiterten Kapitels.

In **Kapitel 5** werden die unterschiedlichen Programmbausteine des TIA Portals gezeigt, wie Funktionen FC und Funktionsbausteine FB, sowie deren Unterschiede in der Handhabung der Projektierung. Weiterhin werden in diesem Kapitel die Organisationsbausteine einer CPU S7-1200 gezeigt und in ihrer Funktion erklärt.

Mit der Datenspeicherung innerhalb eines Projekts und der Anwendung von Datenbausteinen schließt dieses erweiterte Kapitel ab.

Im größtenteils neu gestalteten **Kapitel 6** geht es dann mit der Anwendung von vorkonfigurierten Templates, beispielsweise Zeit- und Zählfunktionen und eines PID-Regelbausteins, wieder konkret um die Programmentwicklung mit SCL. Die Anwendung und Parametrierung dieser Templates wird an dieser Stelle anhand von Beispielen ausführlich dargestellt und erklärt.

Aufbauend auf Kapitel 3 erfolgt in **Kapitel 7** eine Vertiefung im Bereich der unterschiedlichen Datentypen in der Automatisierungstechnik. Das Thema „Komplexere Daten und Datenstrukturen" führt ein in die Erstellung und Anwendung von PLC-Datentypen. Auf der Grundlage von Ulrich Kanngießers Ausführungen führt dieses Kapitel kleinschrittig und nachvollziehbar in die komplexen Datenstrukturen ein und verhilft den Lernenden zu einem sicheren Umgang mit unterschiedlichen Datentypen und mit zusammengesetzten Datenstrukturen.

Vertiefte SCL-Programmierung mit dem CPU-System S7-1500 bietet das komplett neu gestaltete **Kapitel 8** mit dem Schwerpunkt der Wort- und Analogwertverarbeitung. Mit ausführlich dargestellten Anwendungsbeispielen aus der Praxis, wie die Projektierung von Reglern, werden hier die Vorteile der SCL-Programmierung besonders deutlich. Eine projektierte Motorüberwachung mit dem Datentyp VARIANT schließt dieses Kapitel mit einer komplexen Thematik ab.

In **Kapitel 9** werden praxisrelevante Anwendungen in SCL anhand von Projektierungsbeispielen gezeigt und detailliert erklärt. Hier werden Standardfunktionen wie Zählbausteine, Flankenauswertungen oder Skalierungsbausteine nach eigenen Vorstellungen mit SCL-Codes erstellt. Eine vergleichsweise Darstellung der Projektierung einer Ablaufsteuerung mit einer CASE...OF-Anweisung in SCL und einer Projektierung mit der Ablaufsprache GRAPH runden dieses komplett neu gestaltete Kapitel ab.

Im neu gestalteten **Anhang** befindet sich eine Übersicht zu den gängigen und etablierten Begriffen im Bereich der Automatisierungstechnik.

1.1 Arbeiten mit SCL im TIA Portal

Die Kernaufgabe für das Siemens TIA Portal (TIA = Totally Integrated Automation) als Engineering-Tool ist die Projektierung und Programmierung von Steuerungs- und Automatisierungsanlagen. Das TIA Portal stellt für die Entwicklung von intelligenten Automatisierungslösungen mehrere Programmiersprachen nach IEC 61131-3 zur Verfügung. Es gibt die vielfach verwendeten grafischen Darstel-

lungen Funktionsplan FUP, Kontaktplan KOP und Graph sowie die textbasierten Programmiersprachen Anweisungsliste AWL und Structured Control Language SCL.

Es ist grundsätzlich möglich, in den grafischen Programmierdarstellungen FUP und KOP einzelne SCL-Netzwerke einzufügen. Mit dieser Programmgestaltung können einfache Verknüpfungen übersichtlich in grafischer Form mit dem FUP und aufwendige Berechnungen mit dem SCL-Code erstellt werden.

Zu beachten ist in diesem Zusammenhang, dass nicht alle Programmierdarstellungen in allen CPU-Typen und Programmbausteinen zur Verfügung stehen. Im CPU-Typ S7-1200 stehen AWL und Graph nicht zur Verfügung. Eine Programmierung einer Ablaufsteuerung mit Graph ist in den anderen CPU-Typen nur in einem Funktionsbaustein FB möglich.

Die Programmierung mit SCL ist in allen CPU-Typen uneingeschränkt möglich.

1.2 Erstellen eines einfachen Beispielprogramms mit SCL

Die Arbeit mit der Programmiersprache SCL unterscheidet sich deutlich von den grafischen Programmiersprachen Funktionsplan FUP und Kontaktplan KOP. Während in FUP und KOP speziell in kleineren Projekten häufig mit direkter Adressierung und im Programm auch mit physischen Operanden gearbeitet wird, verwendet man in der hochsprachenorientierten SCL-Programmierung grundsätzlich Variablen, die auch als Formaloperanden bezeichnet werden. Die Variablen haben keine physikalische Adresse, stehen aber funktionell mit dem passend deklarierten Datentyp stellvertretend als reservierte Speicherbereiche für den angeschlossenen Sensor oder Aktor zur Verfügung. In der folgenden Projektierung soll exemplarisch die Vorgehensweise bei der Erstellung eines SCL-Programms mit Variablen und Datentypen gezeigt werden.

1.2.1 Projektierungsbeispiel Pressensteuerung

Ein Drehstrommotor, der eine Werkstattpresse steuert, soll im Zweihandbetrieb mit zwei Tastern NO gefahren werden können. Eine Kontrollleuchte signalisiert den Motorbetrieb, eine entsprechende Motorschutzeinrichtung ist vorzusehen.

Für die Sensoren und Aktoren der Steuerung werden Variablen bestimmt, die in der Bausteindeklaration zunächst keine konkreten physikalischen Adressen erhalten, sondern in ihrer Bezeichnung allgemein gehalten werden, also wird statt der Bezeichnung S1_Ein für den ersten Eintaster NO die Bezeichnung x_Eintaster_1

gewählt. Das x steht dabei für eine Bit-Information mit den Signalzuständen 0 oder 1.

Für dieses einfache SCL-Programm wird als Programmbaustein eine Funktion FC gewählt, die folgendermaßen deklariert und programmiert wird.

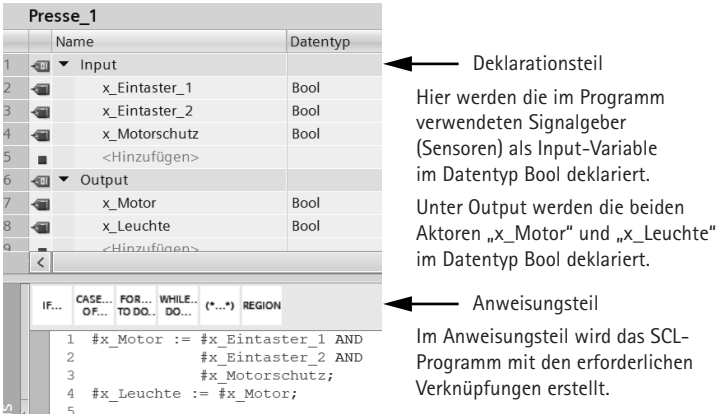

Bild 1.1 Funktion „Presse_1“ in SCL-Darstellung

Im Anweisungsteil sind oben die wichtigsten SCL-Befehle wie IF, CASE...OF oder FOR...TO DO ausgewiesen, die per Mausklick in das Programm gezogen und parametriert werden können. Der eigentliche Editierbereich ist fortlaufend nummeriert und es wird im Gegensatz zur grafischen Programmierung FUP nicht in Netzwerken, sondern bei größeren Programmen mit Regionen gearbeitet. Eine weitere Unterscheidung von FUP und SCL ist, dass in der Bearbeitung von Zuweisungen von links nach rechts gearbeitet wird. In der Zeile 1 (links) ist die Zuweisung der Output-Variablen „x_Motor“ zu sehen. Nach einer Zuweisung erfolgt immer die Zeichenfolge := und anschließend folgen die AND-Verknüpfungen der Input-Variablen „x_Eintaster_1“, „x_Eintaster_2“ und „x_Motorschutz“. Die Verknüpfung wird grundsätzlich mit einem Semikolon abgeschlossen.

In Zeile 4 ist die Zuweisung der zweiten Output-Variable „x_Leuchte“ mit der Abfrage der Output-Variablen „x_Motor“ programmiert. Dies ist möglich, da beide Output-Variablen die gleichen Einschaltbedingungen haben. Auch diese Programmzeile wird mit einem Semikolon abgeschlossen. Das SCL-Programm zur Steuerung der Werkstattpresse ist nun fertiggestellt.

Auf eine Einhaltung der korrekten Syntax ist unbedingt zu achten, wobei das Programm mit einer integrierten Syntaxprüfung während des Programmierens durch das Signalisieren von Fehlern wie beispielsweise falsche Datentypen oder

Syntaxfehler hinweist. Mit der Funktion „Übersetzen“ erfolgt eine Überprüfung des fertiggestellten SCL-Bausteins, sodass eventuell lokalisierte Fehler beim Übersetzen angezeigt werden.

Jetzt kann der erstellte Baustein %FC1 „Presse_1“ im übergeordneten Baustein aufgerufen und mit physischen Operanden beschaltet werden. Ein physischer Operand besteht beispielsweise aus dem symbolischen Operanden „S1_Ein“ und dem absoluten Operanden % I0.0, wobei der absolute Operand die Adresse (Anschlusspunkt) an der digitalen Eingangsbaugruppe DI 0 ist.

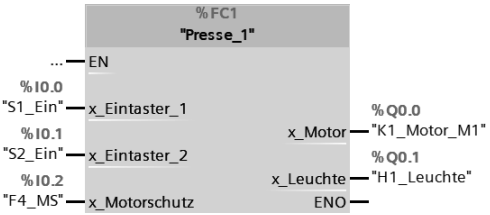

Bild 1.2 Aufruf des FC im übergeordneten Baustein

Der Baustein FC kann nur temporäre Variablen verarbeiten, das bedeutet, dass die Signalzustände nur während der Bausteinbearbeitung gültig sind. Nach der Abarbeitung des Bausteins FC werden alle Variablen grundsätzlich neu aufgefrischt. In einem Funktionsbaustein FB stehen zusätzlich statische Variablen zur Verfügung, die auch nach der Abarbeitung des Bausteins ihre Gültigkeit behalten. Dies ist möglich, da der Funktionsbaustein FB über ein Gedächtnis in Form eines Instanz-Datenbausteins verfügt.

An dieser Stelle sei darauf hingewiesen, dass die unterschiedlichen Bausteinarten des TIA-Portals in Kapitel 5 in ihren Eigenschaften ausführlich erklärt werden.

1.2.2 Arbeiten mit Projekten im Siemens TIA Portal

Die Neuauflage dieses Buch ist größtenteils auf dem zu dieser Zeit aktuellen Softwarestand TIA Portal V19 ausgearbeitet worden. Sollte ein Projekt in einer älteren Softwareversion erstellt worden sein, so ist es problemlos auf die aktuelle Version V19 anpassbar, indem man dieses Projekt einfach hochrüstet. In **Bild 1.3** ist das Anwenderprogramm „Analog_1_2022“ mit der Version V17 erstellt worden.

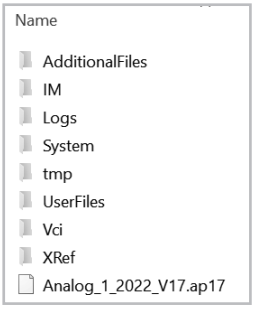

Bild 1.3 Projektordner mit Softwarestand V17

Durch Öffnen der Ausführungsdatei „Analog_1_2022_V17“ und der Endung „ap17“ wird das Projekt auf dem Stand der Softwareversion V17 geöffnet und kann wie in **Bild 1.4** dargestellt auf die Version V19 hochgerüstet und somit auf den aktuellen Stand gebracht werden.

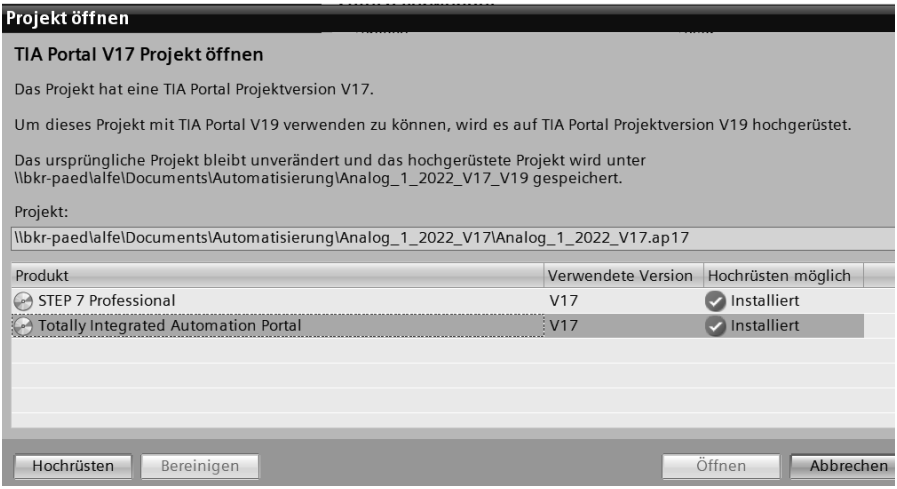

Bild 1.4 Menüpunkt Hochrüsten auf den Softwarestand V19

Nach dem Hochrüsten kann die Projektansicht geöffnet werden und in dem Projekt gearbeitet werden. Es bietet sich an, dass Projekt direkt nach dem Hochrüsten einmal zu speichern, damit der alte Projektzustand auch in der neuen Version V19 zunächst erhalten bleibt.