

**Jörg Kahlert**

**Einführung in die  
Programmierung der LOGO!**  
in Funktionsbausteinsprache

**Mit einer Einführung in GRAFCET**

**Stand: 11/2016**

**Über den Autor:**

Dr.-Ing. Jörg Kahlert leitet ein Ingenieurbüro für Software-Engineering & Automatisierungstechnik in Hamm. Er lehrt die Fächer "Steuer- und Regelungstechnik" und "Automatisierungstechnik" an der SRH Fachhochschule Hamm sowie an der Fachschule für Technik am Max-Born-Berufskolleg in Recklinghausen.

## Vorwort

Die Programmierung von speicherprogrammierbaren Steuerungen (SPS) als wesentliche Teilaufgabe bei der Automatisierung von Prozessen war bis vor einigen Jahren noch ausnahmslos entsprechend ausgebildeten Experten vorbehalten, die in teilweise etwas kryptisch anmutenden Programmiersprachen wie der berüchtigten *Anweisungsliste* oder auf Basis primitiver und unhandlicher grafischer Editoren die zugehörigen Steuerprogramme entwickelten und in Betrieb nahmen. Obwohl der PC seinerzeit längst seinen Siegeszug nicht nur in die Büros, sondern auch in die Werkstatthallen angetreten hatte, verlangten einige Hersteller zur Programmierung ihrer Steuerungen zudem noch die Anschaffung spezieller „Programmiergeräte“, bei denen Anschaffungskosten, Baugröße und Leistungsvermögen nicht in allen Fällen in einem gesunden Verhältnis standen.

Diese Zeiten sind vorbei. Mit dem Erscheinen kompakter, preisgünstiger und dennoch leistungsfähiger Logikmodule mit komfortablen grafischen PC-Programmierumgebungen wie der im Rahmen dieses Lehrbuchs betrachteten LOGO!-Baureihe der Firma SIEMENS oder auch der easy-Serie der Firma EATON – von erfahrenen Anwendern zunächst häufig als „Spielzeug“ abgetan – hat die Automatisierungstechnik einen enormen Zuwachs an potentiellen Anwendern gewonnen. Dies gilt ausdrücklich nicht nur für den industriellen Bereich, sondern auch für Privatanwender, die sich mit der Automatisierungstechnik im Hinblick auf ihre berufliche Weiterbildung auseinandersetzen möchten (oder müssen), oder aber – um den Begriff der Spielzeugs noch einmal aufzugreifen – mehr oder weniger erwachsene „Spielkinder“ sind, die reale Automatisierungsprobleme für den „Hausgebrauch“ lösen wollen; man denke in diesem Zusammenhang nur an das hochaktuelle und omniprésente Thema *Smart Home*. Somit profitieren von den aktuellen Kleinststeuerungen alle Benutzergruppen: Dem erfahrenen Industrieanwender steht am unteren Ende der SPS-Palette für Low-Budget-Projekte eine Alternative zu herkömmlichen Relaissteuerungen zur Verfügung, Schulen und Hochschulen können ihre Automatisierungstechnik-Labore mit einer Vielzahl von Arbeitsplätzen ausstatten, ohne gleich Tausende von Euro investieren zu müssen, und der technikaffine Privatanwender wird in die Lage versetzt, ohne allzu großen finanziellen und materiellen Aufwand kleinere Automatisierungsprojekte in kürzester Zeit bearbeiten zu können.

In diesem Sinne hoffe ich, dass das vorliegende Lehrbuch allen genannten Anwendergruppen zumindest ein wenig Unterstützung beim Einstieg in die Programmierung der LOGO! bietet.

*Hamm, im Oktober 2016*

*Jörg Kahlert*



# Inhalt

<b>1</b>	<b>Einführung</b>	<b>9</b>
<b>2</b>	<b>Installation und Konfigurierung der Begleit-Software</b>	<b>12</b>
2.1	Installation .....	12
2.2	Einstellen der IP-Adresse .....	12
2.3	Erforderliche Einstellungen in <i>LOGO!Soft Comfort</i> .....	13
<b>3</b>	<b>Beschreibung des Anlagenmodells</b>	<b>15</b>
3.1	Aufbau des Modells .....	15
3.2	Übersicht Sensoren/Aktoren.....	20
3.3	Datenaustausch zwischen Modell und LOGO! .....	23
<b>4</b>	<b>Die LOGO! 0BA8</b>	<b>25</b>
4.1	Das LOGO! 0BA8 Basismodul.....	25
4.2	Erweiterungsmodule .....	27
4.3	Anschließen von Sensoren und Aktoren.....	28
4.4	Schließer vs. Öffner .....	31
<b>5</b>	<b>Die Programmierumgebung LOGO!Soft Comfort V8.0</b>	<b>34</b>
5.1	Komponenten der Bedienoberfläche .....	34
5.2	Erstellen und Testen des Programms .....	35
5.2.1	Anlegen eines neuen Projekts .....	35
5.2.2	Einfügen und Bearbeiten von Blöcken .....	37
5.2.3	Verbinden von Blöcken .....	40
5.2.4	Vermeidung offener Blockausgänge .....	43
5.2.5	Beschriften von Diagrammen.....	44
5.2.6	Testen von Programmen im Simulationsmodus.....	46
5.2.7	Übertragen von Programmen auf die LOGO!.....	49
5.2.8	Online-Test von Programmen .....	51
5.2.9	Konfigurierung von Netzwerkeingängen .....	53
<b>6</b>	<b>Digitalwertverarbeitung</b>	<b>55</b>
6.1	Logische Grundsaltungen .....	55
6.1.1	UND-Verknüpfung.....	55
6.1.2	ODER-Verknüpfung .....	57
6.1.3	NICHT-Verknüpfung.....	58
6.2	Zusammengesetzte Schaltungen .....	59

6.2.1	NAND-Verknüpfung.....	59
6.2.2	NOR-Verknüpfung.....	60
6.2.3	Antivalenz (XOR-Verknüpfung).....	61
6.2.4	Äquivalenz (XNOR-Verknüpfung).....	64
6.2.5	Beispiel Hebebühne.....	65
6.3	Programme mit Speicherfunktionen.....	68
6.3.1	Merker.....	69
6.3.2	RS-Speicherbausteine (Flip-Flops).....	71
6.3.3	Wendeschüttschaltungen mit Selbsthaltung.....	74
6.3.4	Richtungserkennung über Flankendetektoren.....	78
6.3.5	Rolltorsteuerung.....	83
6.4	Zeitfunktionen.....	86
6.4.1	Bausteine zur Ein-/Ausschaltverzögerung.....	86
6.4.2	Bausteine zur Impulserzeugung.....	92
6.4.3	Weitere Zeitfunktionen.....	101
6.5	Zähler.....	105
<b>7</b>	<b>Programmierung von Ablaufsteuerungen</b>	<b>113</b>
7.1	Grundlagen von Ablaufsteuerungen und GRAFCET.....	113
7.2	Realisierung in Funktionsbausteinsprache.....	117
7.3	Rücksetzen der Ablaufsteuerung.....	124
7.4	Setzen und Rücksetzen von Aktionen.....	127
7.5	Weitere Typen von Transitionsbedingungen.....	130
7.6	Alternativverzweigungen (Ablaufauswahl).....	133
7.7	Weitere Aktionstypen.....	139
7.7.1	Bedingte Aktionen.....	140
7.7.2	Speichernde Aktion bei Ereignis.....	141
7.7.3	Verzögerte Aktion.....	143
7.7.4	Zeitbegrenzte Aktion.....	144
7.7.5	Zeitabhängige Zuweisungsbedingung.....	145
7.8	Beispiel Aufzugsteuerung.....	146
7.9	Simultanverzweigungen.....	164
<b>8</b>	<b>Analogwertverarbeitung</b>	<b>180</b>
8.1	Analoge vs. digitale Größen.....	180
8.2	Skalierung von Analogsignalen.....	181
8.3	Analoge Sonderfunktionen der LOGO!.....	185
8.4	Analogeingabe.....	186
8.5	Analogausgabe.....	189
<b>9</b>	<b>Regelung</b>	<b>202</b>
9.1	Aufgaben der Regelungstechnik.....	202
9.2	Steuern oder Regeln?.....	203
9.3	Regelkreise im Wirkungsplan.....	207
9.4	Festwert- und Folgeregelung.....	208

9.4.1	Festwertregelung .....	209
9.4.2	Folgeregelung .....	209
9.5	Elemente und Größen des Regelkreises .....	210
9.6	Anforderungen an eine Regelung .....	211
9.7	Typen von Reglern .....	212
9.8	Temperaturregelung mit Zweipunkt-Regler .....	212
9.8.1	Zweipunkt-Regler ohne Schaltdifferenz (Hysterese) .....	212
9.8.2	Temperaturregelung mit Zweipunkt-Regler ohne Schaltdifferenz .....	213
9.8.3	Zweipunkt-Regler mit Schaltdifferenz (Hysterese) .....	216
9.8.4	Temperaturregelung mit Zweipunkt-Regler mit Schaltdifferenz .....	216
9.9	Temperaturregelung mit PI-Regler .....	219
9.9.1	Der PI-Regler .....	219
9.9.2	Der PI-Baustein der LOGO! .....	221
9.9.3	PI-Regelung mit Analogausgang.....	225
9.9.4	PI-Regelung mit PWM-Ausgang .....	227
<b>10</b>	<b>Display und Cursortasten</b> .....	<b>232</b>
10.1	Steuern der Hintergrundbeleuchtung.....	232
10.2	Konfigurierung von Meldetexten .....	235
10.2.1	Grundlagen von Meldetexten .....	235
10.2.2	Allgemeine Einstellungen des Meldetext-Bausteins.....	239
10.2.3	Zusammenstellung des Meldetextes.....	242
10.2.4	Meldungen im Simulationsmodus bzw. Online-Test.....	246
10.2.5	Anwendungsbeispiel PI-Regelung .....	247
10.3	Zeitgesteuerter Wechsel zwischen mehreren Anzeigen.....	250
10.4	Cursortasten .....	253
<b>11</b>	<b>UDF-Programmierung</b> .....	<b>256</b>
11.1	Erstellen und Einschränkungen einer UDF .....	256
11.2	Beispielprojekt: 4-Bit-Ringregister .....	263
<b>12</b>	<b>Sonstige Funktionen</b> .....	<b>269</b>
12.1	Zufallsgenerator.....	269
12.2	Aufzeichnen von Daten (Datenlogger-Baustein).....	271
12.3	Programmgesteuerte Änderung von Parametern .....	281
12.4	Webserver .....	283
12.4.1	Aktivierung des Webserver .....	284
12.4.2	Anmeldung beim Webserver .....	285
12.4.3	Bedienung der LOGO! und Anzeige von Meldetexten .....	286
12.4.4	Anzeigen und Bearbeiten von Variablenspeichertabellen .....	289
	<b>Literatur</b> .....	<b>293</b>
	<b>Stichwortverzeichnis</b> .....	<b>295</b>





# 1 Einführung

Das universelle Logikmodul<sup>1</sup> LOGO! der Firma Siemens hat mit der aktuellen Baureihe OBA8 eine Leistungsfähigkeit erreicht, die bisher ausschließlich größeren (und damit wesentlich teureren) speicherprogrammierbaren Steuerungen (SPS) vorbehalten war. Aufgrund des äußerst günstigen Preises<sup>2</sup> ist die LOGO! zunehmend nicht mehr nur für die Industrie von Interesse, sondern insbesondere auch für den Privatanwender, der einfache Automatisierungsaufgaben beispielsweise in der Haustechnik (Markisensteuerung, Klimatisierung von Wintergärten oder Gewächshäusern, Lichttechnik, ...) ohne Fremdhilfe und mit geringem Zeit- und Kostenaufwand lösen möchte. Eine Vielzahl unterschiedlicher Erweiterungsmodule, beispielsweise zur Analogwertverarbeitung oder Textausgabe, ermöglicht eine einfache Anpassung der Hardwarekonfiguration an komplexere Aufgabenstellungen. Die zugehörige Programmieroberfläche *LOGO!Soft Comfort* in der aktuellen Version V8.0 ermöglicht dabei die komfortable Konfigurierung und Programmierung der LOGO! über einen herkömmlichen PC.

Der Fokus des vorliegenden Lehrbuchs liegt auf der Programmierung der LOGO! in der sogenannten *Funktionsbausteinsprache* (häufig auch als *Funktionsplan* bezeichnet), einer grafisch orientierten, intuitiven und leicht erlernbaren Programmiersprache, wie sie in ähnlicher Form auch auf allen größeren Steuerungen wie beispielsweise der S7-Baureihe von Siemens zu finden ist. Im Gegensatz zum *Kontaktplan*, ebenfalls eine Standard-Programmiersprache für SPS und auch Bestandteil von *LOGO!Soft Comfort*, besitzt die Funktionsbausteinsprache einen höheren Abstraktionsgrad und erleichtert dem Programmieranfänger daher den Einstieg, während der an elektrische Stromlaufpläne angelehnte Kontaktplan typischerweise eher für Anwender mit einer ausgeprägten Affinität zur Elektrotechnik bevorzugt wird.

Obwohl *LOGO!Soft Comfort* sehr weitreichende Möglichkeiten bietet, ein erstelltes Programm vor der Inbetriebnahme an der realen Anlage zunächst einmal im Simulationsmodus zu testen, ist diese Vorgehensweise für einen Einstieg in die Programmierung recht ineffizient und „öde“. Daher wird im Rahmen dieses Werks auf ein *virtuelles Anlagenmodell* einer Fabrik zurückgegriffen, welches eine Vielzahl von technischen Komponenten (Hebebühne, Presse, Lastenaufzug, ...) enthält, die sich wie die realen Einrichtungen über die LOGO! steuern lassen. Damit lassen sich alle erarbeiteten Beispielprogramme sehr praxis- und realitätsnah austesten, ohne dass tatsächlich auch nur eine einzige Leitung verlegt oder ein

---

<sup>1</sup> Andere übliche Bezeichnungen: Steuerrelais, Kleinsteuerung, Kleinststeuerung, Micro-SPS

<sup>2</sup> Bei Drucklegung ca. EUR 150,- für ein LOGO! Basismodul

einzigster Sensor oder Aktor mit der LOGO! verbunden werden muss.<sup>3</sup> Das erstellte Steuerprogramm läuft dabei aber nicht im Simulationsmodus von LOGO!Soft Comfort, sondern auf der LOGO! selbst, und verhält sich damit exakt so wie später im Einsatz an der realen Anlage.

Das nachfolgende Kapitel beschreibt zunächst die Installation und Inbetriebnahme der Begleit-Software zum Buch, welche einerseits das angesprochene virtuelle Anlagenmodell, andererseits aber auch Musterlösungen für alle im Buch besprochenen Aufgabenstellungen umfasst. Kapitel drei schließt daran mit einer detaillierten Beschreibung des Anlagenmodells inklusive aller (virtuellen) Sensoren und Aktoren an.

Eine kurze Übersicht über die Leistungsfähigkeit des LOGO! 0BA8 Basismoduls sowie die momentan verfügbaren Erweiterungsmodule bietet Kapitel vier. Hier finden Sie zudem Hinweise zum Anschluss von Sensoren und Aktoren; diese sind allerdings bewusst knapp gehalten, da zur Nutzung des virtuellen Anlagenmodells wie bereits erwähnt keinerlei „Verdrahtung“ erforderlich ist. An die Beschreibung der LOGO!-Hardware schließt sich in Kapitel fünf dann eine kurze Einführung in die Bedienung der Programmierumgebung LOGO!Soft Comfort V8.0 an.

Mit Kapitel sechs tauchen wir dann erstmalig in die Programmierung der LOGO! ein. Wir beginnen dabei mit einfachen logischen Grundsaltungen, die wir dann zu zusammengesetzten Schaltungen kombinieren. Einen Schwerpunkt bilden Programme mit Speicherfunktionen, die in der Praxis von elementarer Bedeutung für praktisch alle Steuerungsaufgaben sind. Eine Reihe von Beispielen für die Anwendung von Zeitfunktionen und Zählern schließen das Kapitel ab.

In Kapitel sieben lernen wir Ablaufsteuerungen als einen häufig auftretenden Spezialfall von Steuerungsproblemen kennen. Wir erfahren, wie sich diese über den Umweg über die Beschreibungssprache GRAFCET mit Hilfe der Funktionsbausteinsprache realisieren lassen. Dabei werden wir drei schon recht komplexe Anlagen – einen Lastenaufzug, ein Tauchbad und eine PKW-Waschanlage – Schritt für Schritt automatisieren.

Die anschließenden Kapitel acht und neun beschäftigen sich mit der Analogwertverarbeitung auf der LOGO! und der Regelung technischer Größen als typischem Anwendungsfall dafür. Dabei stehen die Umwandlung digitaler Größen in analoge (und umgekehrt) im Mittelpunkt unseres Interesses.

Die Standardversion des LOGO! Basismoduls besitzt ein integriertes Display, das über LOGO!Soft Comfort z. B. zur Text- oder Grafikausgabe genutzt werden kann. Hinzu kommen Cursortasten, die primär zur Steuerung des LOGO!-Menüs dienen, aber auch zur Eingabesteuerung für das laufende Programm herangezogen werden können. Kapitel zehn

---

<sup>3</sup> *Sensoren* sind alle technischen Komponenten, die Informationen über den Zustand der zu automatisierenden Anlage an die LOGO! liefern, also mit den *Eingängen* der LOGO! verbunden sind. Dies können mechanische Signalgeber wie Schalter oder Taster sein, Lichtschranken, Temperaturfühler und vieles mehr. *Aktoren* hingegen geben Stellbefehle oder Informationen von der LOGO! an die Anlage weiter, sind also an die *Ausgänge* der LOGO! angeschlossen. Dazu gehören beispielsweise Motoren bzw. Motorschütze, Pumpen, Ventile, Heizwendel oder auch alle Arten von Anzeigeleuchten [7].

erläutert anhand einiger Beispiele, was bei der Programmierung diesbezüglich zu beachten ist.

Bei größeren speicherprogrammierbaren Steuerungen lassen sich komplexere Programme durch Erstellung benutzerdefinierter Funktionsbausteintypen deutlich übersichtlicher gestalten. Auch die LOGO! verfügt in Form der *User Defined Functions* (UDFs) über ein entsprechendes Feature, wenn auch in deutlich einfacherer Form. In Kapitel elf erstellen wir zwei einfache UDFs und binden diese in ein übergeordnetes Programm ein.

Mit Kapitel zwölf schließen wir unsere Einführung in die LOGO!-Programmierung ab. Hier finden sich einige für den „Normalanwender“ weniger wichtige Funktionen der LOGO!, die anhand von Beispielen kurz erläutert werden. Dazu gehören beispielsweise die Möglichkeit, im Betrieb Daten auf einer SD-Karte aufzuzeichnen oder der Fernzugriff auf die LOGO! über den integrierten Webserver. Letzterer ist eine der vielen Neuerungen der Baureihe 0BA8 und ermöglicht nicht nur die Beobachtung von Signalen praktisch über jeden Internet-Browser oder ein Smartphone, sondern bei Bedarf auch die Umparametrierung der LOGO! während des laufenden Betriebs von jedem beliebigen Ort.