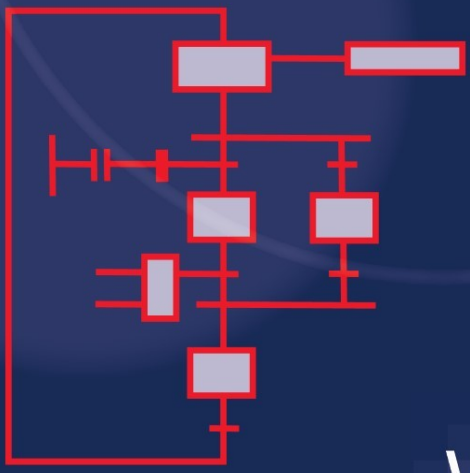
Karl Heinz John · Michael Tiegelkamp

mit CD-ROM · mit CD-ROM ·

SPS-Programmierung mit IEC 61131-3

Konzepte und Programmiersprachen, Anforderungen an Programmiersysteme, Entscheidungshilfen

4., neubearbeitete Auflage



VDI

Springer

SPS-Programmierung mit IEC 61131-3

4., neubearbeitete Auflage

Karl Heinz John • Michael Tiegelkamp

SPS-Programmierung mit IEC 61131-3

Konzepte und Programmiersprachen, Anforderungen an Programmiersysteme, Entscheidungshilfen

4., neubearbeitete Auflage

Springer

Karl Heinz John
Irrlrinnig 13
91301 Forchheim
Deutschland
karlheinz.john@gmx.de

Michael Tiegelkamp
Kurpfalzstr. 34
90602 Pyrbaum
Deutschland
Michael.Tiegelkamp@gmx.de

ISBN 978-3-642-00268-7 e-ISBN 978-3-642-00269-4
DOI 10.1007/978-3-642-00269-4
Springer Dordrecht Heidelberg London New York

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Einbandentwurf: WMXDesign GmbH, Heidelberg

Gedruckt auf säurefreiem Papier

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media (www.springer.com)

Vorwort zur 4. überarbeiteten Auflage

In den letzten Jahren setzte sich die IEC 61131 (bis 1998 „IEC 1131“) als Programmierstandard in der Automatisierungs-Industrie auf breiter Front durch. Von kleinen bis zu den großen SPS-Herstellern bieten heute sehr viele Programmiersysteme an, die diese Norm als Grundlage benutzen. Zusätzliche Standards und Empfehlungen – wie z.B. für Motion Control - erweitern die IEC 61131 um Funktionalität, die sich aus den steigenden Anforderungen des Markts ergeben.

Eine der wichtigsten Weiterentwicklungen stellt die Norm IEC 61499 (ehemals IEC 1499) dar. Ihr ist Kap. 9 gewidmet und ihre Bedeutung für dezentrale SPS-Systeme wird in Kap 7.9 angesprochen.

Heute liegt die IEC 61131 in einer zweiten Version vor. In der vorliegenden 4. Buchauflage wurden die zahlreichen Änderungen und Ergänzungen dieser Norm eingebracht. Ein umfangreiches Stichwort-Verzeichnis (Index) am Ende des Buchs erleichtert eine schnelle Themensuche.

Die beiliegenden DVD und CD enthalten die vollständigen Demo-Versionen zweier Programmiersysteme in aktuellen Versionen. Dadurch kann der Leser Erkenntnisse aus diesem Buch sofort umsetzen und sein Wissen durch praktische Anwendung festigen.
Wir danken den Firmen SIEMENS AG und infoteam Software GmbH für die Bereitstellung dieser beiliegenden Software.

Besonders bedanken wir uns wieder bei Hans-Peter Otto, Mitglied in den IEC- und DKE-Normungsgremien, der uns tatkräftig unterstützte und auch selbst weitere Anregungen mitnahm.

Wir freuen uns über das große Interesse an diesem Buch und danken unseren aufmerksamen Lesern für viele Anregungen und entdeckte Druckfehler.

Karl-Heinz John Michael Tiegelkamp

Winter 2008

Inhaltsverzeichnis

1 Einleitung

Durch den rasanten Fortschritt bei Leistung und Miniaturisierung in der Mikrotechnologie erschließt sich die Speicherprogrammierbare Steuerung (SPS) immer neue Märkte. Speziell konzipierte Regel- und Steuerhardware oder PC-basierte Controller, erweitert um realzeitfähige Hard- und Software, kontrollieren höchst komplexe Automatisierungsprozesse. Neu hinzugekommen ist das Thema „Sicherheitsgerichtete Steuerung“ zur Verhinderung von Personenschäden durch Maschinen im Produktionsprozess.

Die SPSen decken durch ihre unterschiedlichen Ausprägungen ein weites Aufgabenspektrum ab — dies reicht von kleinen Knotenrechnern in Netzwerken und dezentral eingesetzten *Kompaktgeräten* bis hin zur modularen, fehlertoleranten Hochleistungs- SPS. Sie unterscheiden sich in Leistungsmerkmalen wie Verarbeitungsgeschwindigkeit, Vernetzungsfähigkeit oder das Angebot verwendbarer Peripherie-Baugruppen.

Der Begriff SPS sei im Weiteren als Technologie verstanden; dies weicht vom amerikanischen Begriff PLC (Programmable Logic Controller) ab, der darunter lediglich eine bestimmte Hardware-Architektur — im Gegensatz zu den PC-Controllern (*Soft Logic Array*) — versteht. Mit den IEC 61131 Programmiersprachen lassen sich klassische SPSen, embedded Controller, Industrie- PCs, ja sogar Standard- PCs programmieren, wenn geeignete Hardware zum Anschluss an Sensoren und Aktoren vorhanden ist (z.B. Feldbuskarte).

Dieses Spektrum der Leistungsfähigkeit seitens der Hardware erfordert eine entsprechende Unterstützung durch geeignete Programmierwerkzeuge, die eine kostengünstige und qualitätsbewusste Erstellung einfacher wie umfangreicher Anwender-Programme ermöglicht, beispielsweise:

- Gleichzeitige Unterstützung mehrerer SPS-Programmiersprachen,
- Änderung der Programme „on-line“ in der SPS,
- Rückdokumentation der Programme aus der SPS,
- Wiederverwendbarkeit von SPS-Programm-Bausteinen,
- Test und Simulation der Anwenderprogramme „off-line“,
- Integrierte Projektierungs- und Inbetriebnahme-Werkzeuge,
- Qualitätssicherung, Projektdokumentation,
- Verwendung von Systemen mit offenen Schnittstellen.

Auf der Basis von PCs entstanden in den letzten 10 Jahren immer leistungsfähigere *SPS-Programmierwerkzeuge*.

Die bisher verwendeten klassischen SPS-Programmiersprachen wie Anweisungsliste, Kontaktplan oder Funktionsplan stoßen an ihre Grenzen. Die Anwender fordern einheitliche, herstellerübergreifende Sprachkonzepte, höhere Programmiersprachen und Entwicklungswerkzeuge, wie sie im Bereich der PC-Welt bereits seit längerem üblich sind.

Mit Einführung der internationalen Norm IEC 1131 (inzwischen unbenannt in IEC 61131) wurde eine Basis für eine einheitliche SPS-Programmierung geschaffen, die effiziente Konzepte der Software-Technologie berücksichtigt. Die Norm liegt inzwischen in einer überarbeiteten 2. Ausgabe vor, die vollständig in das Buch eingearbeitet ist.

1.1 Gegenstand des Buchs

Dieses Buch hat zum Ziel, eine verständliche Einführung in die Konzepte und Sprachen der Norm IEC 61131 zu geben. Dabei werden einfache Beispiele verwendet, um die Ideen und Anwendung der neuen SPS-Programmiersprachen zu erklären. Ein größeres Beispiel-Programm fasst jeweils die Ergebnisse eines Abschnitts nochmals zusammen.

Das Buch versteht sich als Einführung und Erklärungshilfe für Leute in Ausbildung und Praxis, welche die Möglichkeiten der neuen Norm kennenlernen und nutzen möchten. Das Buch beschreibt die Methodik der Norm aus einer herstellerunabhängigen Sicht. Spezielle Ausprägungen und Varianten einzelner Programmiersysteme sollten den jeweiligen Handbüchern vorbehalten sein.

Es setzt geringe Kenntnisse im Umgang mit Personal Computern und Grundkenntnisse im Bereich der SPS-Technik voraus (siehe dazu [Berger-03] oder [Berger-06]). Auch erfahrene SPS-Programmierer finden hier Informationen, die ihnen die Arbeit mit diesen Programmiersystemen erleichtern. Das Buch legt Wert darauf, den Standard selbst zu beschreiben und weniger die entsprechenden Varianten der Programmiersysteme auf dem Markt.

Berufsschüler und Studenten erhalten ein Nachschlagewerk, das ihnen systematisch das Erlernen des neuen Programmierstandards erleichtert.

Mit Hilfe des „Einkaufsberaters“ kann der Leser zusätzlich SPS-Programmiersysteme selbst und individuell beurteilen, siehe beiliegende CD.

Formale Inhalte und Aufbau der IEC werden praxisgerecht dargestellt, schwierige Sachverhalte im Zusammenhang erläutert und Interpretationsspielräume sowie Erweiterungsmöglichkeiten aufgezeigt.

Dieses Buch soll helfen, dem Leser konkrete Antworten auf folgende Fragen zu geben:

- wie erfolgt die Programmierung nach IEC 61131? Was sind die wesentlichen Ideen der Norm, wie werden sie praxisgerecht umgesetzt?
- wo liegen die Vorteile der internationalen Norm IEC 61131 gegenüber der sonstigen Mikrocontroller- oder PC- Programmierung?
- was müssen heutige Programmiersysteme leisten, um konform zur IEC 61131 zu sein und diese Norm zu erfüllen?
- worauf sollte man als Anwender bei der Auswahl eines SPS-Programmiersystems mindestens achten; welche Kriterien sind für die Leistungsfähigkeit von Programmiersystemen ausschlaggebend?

Kapitel 2 stellt die drei Grundbausteine der Norm vor: **Programm, Funktion und Funktionsbaustein.** Ein Einführungsbeispiel, das die wichtigsten Sprachelemente der Norm beinhaltet und gleichzeitig einen Überblick über ihre Programmiermethodik bietet, schafft einen Einstieg in die Begriffswelt der IEC 61131.

Kapitel 3 beschreibt die **gemeinsamen Sprachelemente** der fünf Programmiersprachen, sowie die Möglichkeit der Datenbeschreibung mit Hilfe von Deklarationen.

Es schließen sich in **Kapitel 4** die **fünf Programmiersprachen** der IEC 61131 an, die ausführlich erläutert und jeweils durch ein umfangreicheres Beispiel ergänzt werden.

Die Mächtigkeit der IEC 61131 beruht u.a. auf einer einheitlichen Beschreibung häufig verwendeter Bausteine, den sogenannten **Standard-Funktionen** und **-Funktionsbausteinen**. Ihre Definition und Verwendungsmöglichkeiten sind in **Kapitel 5** beschrieben.

Nach der Programmierung erfolgt mittels der **Konfiguration** eine Zuweisung der Programme und Daten an die Eigenschaften und Hardware der ausführenden SPS. Dies ist in **Kapitel 6** zu finden.

Der SPS-Markt entwickelte sich zu einer Technologie mit sehr spezifischen Anforderungen. Diese **Besonderheiten der Programmerstellung** mit einer SPS sowie die Umsetzung mit den Mitteln der IEC 61131 sind das Thema von **Kapitel 7**.

Kapitel 8 fasst die wichtigsten Eigenschaften der Norm aus Kapitel 2 bis 7 zusammen. Die wesentlichen Vorteile der Norm und konformer Programmiersysteme können nochmals nachgelesen werden.

Kapitel 9 stellt die Norm **IEC 61499** vor, die verteilte Automatisierungsprozesse behandelt. Sie basiert auf IEC 61131-3, erweitert diese jedoch ganz erheblich um eine Sichtweise, die sich aus den Anforderungen an hohe Parallelität und Dezentralisierung moderner Automatisierungsaufgaben ergibt.

Kapitel 10 erläutert die Benutzung der mitgelieferten CD. Sie beinhaltet Programmierbeispiele dieses Buchs, einen Einkaufsberater in Tabellenform, sowie ablauffähige Versionen zweier IEC -Programmiersysteme.

Die sich anschließenden **Anhänge** liefern weitere Detail-Informationen zu den Kapiteln.

Das **Glossar** aus **Anhang I** stellt alphabetisch die wichtigsten Begriffe dieses Buchs mit einer Kurzerklärung zusammen.

Anhang J enthält das Literaturverzeichnis, in dem neben **Büchern** auch **Referenzen** auf Fachaufsätze zum Thema IEC 61131-3 enthalten sind.

In **Anhang K** ist ein allgemeiner Index zu finden, der zum Auffinden von Stichworten sehr hilfreich sein kann.

1.2 Die Norm IEC 61131

Die mehrteilige Norm IEC 61131 fasst Anforderungsdefinitionen für *SPS-Systeme* zusammen. Die Anforderungen betreffen die SPS-Hardware und das Programmiersystem.

Die Norm umfasst sowohl gängige Konzepte der bisherigen SPS-Programmierung als auch Erweiterungen um neue Programmiermethoden.

Die *IEC 61131-3* sieht sich als **Richtlinie zur SPS-Programmierung**, nicht als starr bindende Vorschrift. Die enorme Menge der Festlegungen führte dazu, dass die Programmiersysteme nur einen mehr oder weniger großen Teil der Norm realisiert haben, die durchaus eine unterschiedliche Ausprägung haben können. Diesen Umfang hat der SPS-Hersteller zu dokumentieren: will er normkonform sein, hat er nachzuweisen, in welchen Teilen er die Norm erfüllt oder nicht.

Dazu beinhaltet die Norm 62 Tabellen (engl.: **feature tables**) mit Anforderungsdefinitionen, die vom Hersteller jeweils mit Vermerken („ist erfüllt; ist nicht implementiert; folgende Teile sind erfüllt:...“) zu versehen sind.

Die Norm bildet so den **Maßstab**, an dem sich sowohl Hersteller als auch Anwender orientieren können, um festzustellen, in welchem Maß sich ein Programmiersystem an den Standard hält, d.h. konform zur IEC 61131-3 ist.

Zum Nachweis der Konformität definiert die PLCopen, Abschn. 1.3, weitere Tests für Konformitätsklassen, die von unabhängigen Instituten durchgeführt werden können.

Die Norm wurde von der Arbeitsgruppe SC65B WG7 (anfangs: SC65A WG6) der internationalen Normungsgruppe **IEC (International Electrotechnical Commission)** erarbeitet, die sich aus Vertretern verschiedener SPS-Hersteller, Softwarehäusern und Anwendern zusammensetzt. Dies hat den Vorteil, dass sie von den meisten SPS-Herstellern als Richtlinie akzeptiert wird. So hat sich die

IEC 61131-3 in den letzten Jahren als einziger Weltstandard zur SPS-Programmierung durchgesetzt.

1.2.1 Ziele und Nutzen der Norm

Durch die ständig steigende Komplexität der SPS-Systeme wachsen die Kosten für:

- die Ausbildung der Anwendungsprogrammierer,
- die Erstellung der immer größeren Programme,
- die Implementierung immer komplexerer Programmiersysteme.

Die SPS-Programmiersysteme folgen, zeitlich verzögert, dem Trend des Software-Massenmarkts in der PC-Welt. Wie dort lässt sich der Kostendruck vor allem durch Vereinheitlichung, Synergien sowie Wiederverwendung bereits erstellter Software mildern.

Hersteller (SPS- Hardware und -Software).
Mehrere Hersteller können gemeinsam in die – mit der heute vom Markt geforderten Funktionalität – viele Millionen € teure Programmiersoftware investieren.

Die Grundausprägung eines Programmiersystems ist durch die Norm weitgehend festgelegt. Basissoftware wie Editoren kann, bis auf spezifische Teile wie Codegeneratoren oder „on-line“-Module, gemeinsam genutzt werden. Die Marktdifferenzierung erfolgt über sogenannte Zusatzbausteine zum Grundpaket, die in speziellen Marktsegmenten benötigt werden, sowie über die SPS-Hardware. Durch den Zukauf von Fremdprodukten lassen sich die Entwicklungskosten wesentlich verringern. Die Fehleranfälligkeit neu entwickelter Software wird durch die Verwendung bereits erprobter Software stark reduziert

Der Entwicklungsaufwand heutiger Programmierwerkzeuge ist durch die geforderte Funktionalität deutlich gewachsen. Durch den Zukauf von Softwarekomponenten (nur ein Teil der Programmierhersteller bietet dies an) oder Komplettsystemen kann die Zeit bis zur Markteinführung wesentlich verkürzt werden, so dass mit der raschen Hardware-Entwicklung Schritt gehalten werden kann.

Anwender.
Anwender arbeiten oft gleichzeitig mit SPS-Systemen unterschiedlicher Hersteller. Mussten bislang Mitarbeiter mehrere Programmier-Ausbildungen durchlaufen, beschränkt sich dies bei IEC 61131-3- Systemen auf die Bedienspezifika der einzelnen Programmiersysteme und zusätzliche Besonderheiten der SPSen. Systemspezialisten und Ausbildungspersonal können daher eingespart, SPS-Programmierer flexibler eingesetzt werden.

Die Anforderungsdefinitionen der Norm erleichtern die Auswahl geeigneter Programmiersysteme, denn sie sind dadurch vergleichbarer geworden.

Ist auch in nächster Zeit nicht abzusehen, dass vollständige Anwenderprogramme ohne Anpassung zwischen unterschiedlichen *SPS-Systemen* ausgetauscht werden können, so entsprechen sich doch Sprachelemente und Programmstruktur bei den verschiedenen IEC- Systemen. Dies erleichtert die Portierung auf Fremdsysteme.

1.2.2 Geschichte und Bestandteile

Die Norm *IEC 61131* stellt eine Zusammenfassung und Fortschreibung verschiedener Normen dar. Sie verweist allein auf 10 internationale Standards (IEC 50, IEC 559, IEC 617-12, IEC 617-13, IEC 848, ISO/AFNOR, ISO/IEC 646, ISO 8601, ISO 7185, ISO 7498). Sie beinhalten Vorschriften über den verwendeten Zeichencode, Definition der verwendeten Nomenklatur oder den Aufbau grafischer Darstellungen.

Bereits in der Vergangenheit wurden mehrere Versuche unternommen, eine Norm der SPS-Programmier-Technologie zu etablieren. Der IEC 61131- Standard ist die erste Norm, welche die nötige internationale (und industrielle) Akzeptanz erhält. Die wichtigsten Vorläuferpapiere zur IEC 61131 sind in der Tab. 1.1 zusammengestellt.

Jahr	deutsch	international
1977	DIN 40 719-6 (Funktionspläne)	IEC 848
1979		Start der Arbeitsgruppe für den ersten IEC 61131 Draft
1982	VDI-Richtlinie 2880, Blatt 4 SPS-Programmiersprachen	Fertigstellung erster Draft IEC 61131; Aufteilung in 5 Unterarbeitsgruppen
1983	DIN 19239 SPS-Programmierung	Christensen Report (Allen Bradley) SPS Programmiersprachen
1985		Erste Ergebnisse der IEC 65 A WG6 TF3

Tab. 1.1. Wichtige Vorläufer der IEC 61131-3

DIN EN mit anschließender Nummer bezeichnet die deutsche Norm, bei der internationalen (in englischer Sprache) Version steht dafür die Bezeichnung IEC, gefolgt von der gleichen Nummer. Die Abkürzung Ed. steht für Edition und gibt den jeweiligen Ausgabestand an. TR kürzt **Technical Report** ab und bezeichnet eine nicht- bindende Erklärung zur Norm.

Die *Norm* umfasst (Stand Dezember 2007) sieben Teile. In der nachfolgenden Übersicht sind die jeweiligen Titel aufgeführt, in Klammern sind das Jahr der Erstveröffentlichung sowie der aktuellen Fassung zu finden:

- DIN EN 61131-1 Ausgabe 2: Allgemeine Informationen (1994; 2004) [DIN EN 61131-1]
- DIN EN 61131-2 (VDE 0411 T 500): Betriebsmittelanforderungen und Prüfungen (1994; 2007) [DIN EN 61131-2]
- DIN EN 61131-3 Ed. 3: Programmiersprachen (1993; 2003); die nächste Überarbeitung erfolgt voraussichtlich 2010 [DIN EN 61131-3]
- IEC 61131-4 TR Ed 2.0: User guidelines (1995; 2004) [IEC 61131-4 TR]
- DIN EN 61131-5: Kommunikation (2001) [DIN EN 61131-5]
- DIN EN 61131-7: Fuzzy-Control-Programmierung (2000) [DIN EN 61131-7]
- DIN EN 61131-8 TR Ausgabe 2: Leitlinien für die Anwendung und Implementierung von Programmiersprachen für Speicherprogrammierbare Steuerungen (1997; 2005) [DIN EN 61131-8]. Dieses Papier wurde auch unter der Bezeichnung DIN EN 61131-3 Beiblatt 1 veröffentlicht.

Zusätzlich werden vereinzelt **Corrigenda** zu den Normen veröffentlicht, welche Fehlerbeschreibungen der aktuell gültigen Ausgabe beinhalten und einen Vorschlag zur Behebung vorwegnehmen.

Teil 1: Allgemeine Informationen:
Teil 1 enthält „allgemeine Begriffsbestimmungen und typische Funktionsmerkmale, die eine SPS von anderen Systemen unterscheiden. Hierzu zählen Standardeigenschaften wie beispielsweise die zyklische Bearbeitung des Anwenderprogramms mit dem gespeicherten Abbild der Eingangs- und Ausgangswerte oder die Arbeitsteilung zwischen Programmier-, Automatisierungs- und Bedien- und Beobachtungsgerät."

Teil 2: Betriebsmittelanforderungen und Prüfungen:
Dieser Teil „definiert die elektrischen, mechanischen und funktionellen Anforderungen an die Geräte sowie entsprechende Typprüfungen. Es sind die Umgebungsbedingungen (Temperatur, Luftfeuchte usw.) und Beanspruchungsklassen der Steuerungen und Programmiergeräte vorgegeben.".

Teil 3: Programmiersprachen:
Hier „wurden die weltweit verbreiteten SPS-Programmiersprachen in einer harmonisierten und zukunftsweisenden Überarbeitung aufgenommen."

Über formale Definitionen, durch lexikalische, syntaktische und (teilweise) semantische Beschreibungen sowie Beispiele werden das grundlegende Softwaremodell und die Programmiersprachen festgelegt.

Teil 4: Anwenderrichtlinien
Der vierte Teil ist „als Leitfaden konzipiert, um den SPS-Anwender in allen Projektphasen der Automatisierung zu beraten. Die praxisgerechten Hinweise reichen von der Systemanalyse über die Gerätewahl bis zur Wartung."

Teil 4 wurde in der Erstversion in Deutschland unter dem Titel „Beiblatt zu DIN EN 61131" veröffentlicht.

Teil 5: Kommunikation:
Dieser Teil „behandelt die Kommunikation von SPSen unterschiedlicher Hersteller miteinander und mit anderen Geräten."

In Zusammenarbeit mit der ISO 9506 (Manufacturing Message Specification; MMS) erfolgt die Definition von Konformitätsklassen, die SPSen zur Kommunikation (z.B. über Netz) nutzen können. Dies umfasst die Funktionen Geräteanwahl, Datenaustausch, Alarmverarbeitung, Zugriffskontrolle und Netzwerkverwaltung.

Teil 6: Sicherheitsgerichtete SPS:
Aktuell arbeitet das Normungskomitee an der Ersterstellung der IEC 61131-6, „Sicherheitsgerichtete SPS". Ziel ist die Aufbereitung der Anforderung der Sicherheitsnorm IEC 61508 („Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme") und der Maschinenanforderung IEC 62061 („Sicherheit von Maschinen - Funktionale Sicherheit sicherheitsbezogener elektrischer, elektronischer und programmierbar elektronischer Steuerungssysteme") an SPSen..

Teil 7: Fuzzy Control Language:
Das Ziel dieses Normteils ist es, dem Hersteller und dem Anwender ein gemeinsames Verständnis zur Integration von Fuzzy kontrollierten Anwendungen auf Basis der IEC 61131-3 zu geben. Ein weiteres Ziel ist die Portierungsmöglichkeit von Fuzzy Programmen zwischen unterschiedlichen Hersteller.

Teil 8: Leitlinien für die Anwendung und Implementierung von Programmiersprachen für Speicherprogrammierbare Steuerungen:
Dieses Dokument liefert Interpretationen zu offen gebliebenen Fragen der Norm. Es enthält Richtlinien zur Implementierung, Verwendungshinweise für den Endanwender sowie programmiertechnische Hilfestellungen.

Die Norm beschreibt eine moderne Technologie und ist damit einem starken Innovationsdruck ausgesetzt, weshalb auch eine nationale und internationale Fortentwicklung der erreichten Ergebnisse erfolgt.

Dieses Buch beschäftigt sich mit dem Teil 3 „Programmiersprachen", kurz IEC 61131-3. Es beinhaltet die letzten Änderungen und Erweiterungen, welche 2003 mit der Ausgabe 2 eingearbeitet wurden.

Zu Inhalt und Entstehung der Norm IEC 61131 siehe auch [Rolle-98].

1.3 Organisation PLCopen

Die *PLCopen* [PLCopen Europe] ist eine hersteller- und produktunabhängige internationale Vereinigung. Ihr gehören viele SPS-Hersteller, Softwarehäuser und unabhängige Institute in Europa und Übersee an. Die Mitglieder aus den verschiedenen Branchen konzentrieren sich auf die Harmonisierung im Bereich der Steuerungsprogrammierung sowie auf die Anwendungs- und Softwareschnittstellenentwicklung im Umfeld der IEC 61131-3.

Zur Reduzierung der Kosten im industriellen Engineering wurden einheitliche Vorgaben und Ausführungsrichtlinien erstellt. Das Ergebnis dieser Arbeit sind zum Beispiel standardisierte Bibliotheken für verschiedene Anwendungsbereiche, die Ausarbeitung eines Konformitätslevels für Programmiersprachen und Schnittstellen für einen verbesserten Softwareaustausch. Die Experten der PLCopen- Mitglieder sind in technischen Komitees organisiert und definieren diese offenen Standards zusammen mit dem Endanwender.

1.3.1 Ziele von PLCopen

PLCopen wurde 1992 direkt nach der Veröffentlichung der Norm IEC 61131-3 gegründet. Der Steuerungsmarkt war damals ein sehr heterogener Markt mit verschiedensten Programmiermethoden für viele unterschiedliche SPS-Typen. Heute ist die IEC 61131-3 ein weltweit akzeptierter Programmierstandard und viele Software- und Hardware-Hersteller bieten Produkte auf der Basis dieses Standards an, der sich auf diese Weise – dank PLCopen – in vielen verschiedenen Maschinen und anderen Anwendungsbereichen wieder findet.

Heute sind die Herausforderungen auf dem Markt der industriellen Steuerungen ganz andere und die PLCopen folgt den Anforderungen des Marktes, wobei der Kernbereich eine effizientere Automatisierung durch die Definition allgemeiner Standards ist. Die aktuellen Themen sind:

- *Motion Control- und Sicherheits-Funktionen*,
- *XML-Datenaustauschformat* zur Standardisierung der Basisdaten von IEC-Projekten in Software-Systemen und
- *Benchmarking*- Projekte zur Ausarbeitung eines detaillierten Benchmark-Standards .

Weitere, neue Automatisierungsaufgaben werden auch in Zukunft durch neue industrielle Anforderungen und neue Produkte entstehen und es wird die Aufgabe der PLCopen bleiben, eine globale Harmonisierung herbeizuführen und ein einheitliches Verständnis zu schaffen.

Es handelt sich bei der PLCopen somit um kein weiteres Normungsgremium, sondern um eine Interessengemeinschaft, die existierende Standards zur inter-

nationalen Akzeptanz verhelfen möchte. Detaillierte Information ist über das Internet abrufbar (http://www.PLCopen.org).

1.3.2 Gremien und Arbeitsgebiete

Die PLCopen ist in mehrere Komitees aufgeteilt, die jeweils ein Arbeitsgebiet behandeln:

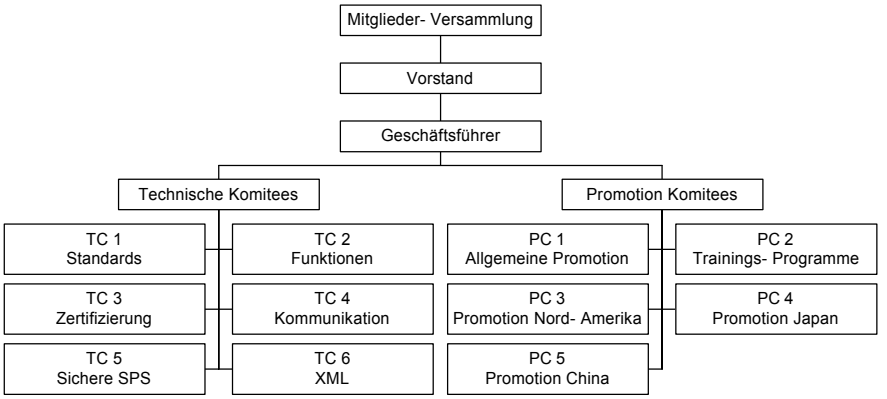


Abb. 1.1. Komitees der PLCopen

Die technischen Komitees erarbeiten Richtlinien und Vorgaben zum gemeinsamen Vorgehen; die Promotion- Gremien sind für Marketingmaßnahmen verantwortlich.

Die Arbeit in den Komitees wird ausschließlich von Vertretern einzelner Firmen und Institute vorgenommen; dies garantiert, dass die erarbeiteten Papiere in den Unternehmen umgesetzt werden können.

1.3.3 Ergebnisse

Durch vorbereitende Maßnahmen der Promotion Komitees konnten mehrere Messen in Europa, USA und Fernost beschickt werden. Workshops und Fortbildungsseminare brachten der PLCopen die erwünschte internationale Anerkennung.

Als Diskussionsforum von Anwendern, Hersteller und Software-Häusern konnten respektable technische Ergebnisse erzielt werden:

- *Zertifizierung* für Hersteller von SPS-Programmiersystemen,
- *Austauschformat* für Anwenderprogramme.

Die Komitees im Einzelnen:

TC 1 – Standards.
Dieses Komitee ist die Schnittstelle zu den internationalen Normengremien der IEC und OPC [OPC Foundation]. Mitglieder des Komitees sammeln Vorschläge zur Verbesserung oder Fehlerkorrektur für das verantwortliche IEC 61131 Normungsgremium *IEC 65B WG7 working group* [IEC IEC 65B WG7] ; sie entwickeln gemeinsame Positionen unter ihren Mitgliedern, die sie wiederum den Normengremien vorschlagen und Neues aus den Gremien publizieren. Insbesondere wurden hier Verbesserungen für die 2. Ausgabe der Norm eingebracht.

TC 2 – Functions.
Seine Mitglieder definieren Bibliotheken von Funktionsblöcken. Als Marktstandard hat sich beispielsweise die PLCopen Motion Bibliothek durchgesetzt. Dieses Dokument untergliedert sich in folgende Teile:

- Part 1: Basisdokumentation,
- Part 2: Erweiterungen, zusätzliche Funktionsblöcke,
- Part 3: Anwender- Richtlinien (Richtlinien und Beispiele für den Anwender,
- Part 4: Interpolation, fokussiert auf koordinierte Multiachsen Bewegungen im 3D- Raum,
- Part 5: Homing (diese Funktion dient der Referenzierung einer speziellen mechanischen Position).

TC 3 – Certification.
Um die IEC 61131-3 Konformität von Programmiersystemen zu prüfen, wurden mehrere Zertifizierungsstufen definiert und Testsoftware erstellt. Die Prüfung erfolgt über PLCopen akkreditierte Institute, um die angestrebte Qualität des Nachweises zu garantieren. Folgende Stufen sind definiert:

- **Base Level (BL):** Ursprüngliche Basisdefinition. Sie schreibt den wesentlichen Aufbau eines Programms nach IEC 61131-3 Methodik vor sowie die Erklärung des Programmierherstellers, sich an der Norm zu orientieren.
- **Reusability Level (RL):** Funktionen und Funktionsbausteine sind soweit kompatibel, dass sie auf unterschiedliche Programmiersysteme mit diesem Level portiert werden können.
- **Conformity Level (CL):** die strengste Konformitätsprüfung. Durch die Vielzahl von Datentypen in IEC 61131-3 (26) nutzen die Programmiersysteme oft nur einen Teil. Beim Konformitätstest werden nun alle Datentypen, welche der Hersteller für sein System angibt, einer strengen Kontrolle unterworfen.

Die Benchmark Workgroup ist ebenfalls dem TC 3 zugeordnet. Sie definiert Spezifikationen und Tests für Skripts, welche eine reproduzierbare und portierbare Leistungsmessung von Programmiersystemen erlauben.

TC 4 – Communication.
TC 4 bearbeitet Definitionen an der Schnittstelle zwischen IEC 61131-3 Programmiersystemen und Kommunikationssystemen wie Profibus oder CAN.

TC 5 – Safe Software.
Dieses Komitee stellt Empfehlungen auf, um IEC 61131-3 Systeme in sicherheitskritischen Umgebungen einsetzen zu können. Dies schließt insbesondere die neuen Normen IEC 61508 und IEC 61511 mit ein. Weiterhin werden Richtlinien für Anwender erstellt, um Sicherheitsaspekte in sicherheitskritischen Anwendungen zu gewährleisten.

TC 6 – XML.
TC 6 definiert ein XML Schema, um IEC 61131-3 Anwenderprogramme und -projekte in XML zu beschreiben. Dies schließt die textuellen wie grafischen Sprachen, Variablendeklaration und Konfiguration ein. Diese Spezifikation unterstützt

- den Austausch von Bausteinen zwischen Systemen
- die Schnittstelle zu anderen Softwarepaketen wie Dokumentations-, Simulations- oder Verifikationswerkzeuge.

PC 1 – General Promotion.
Das Komitee ist zuständig für Europa sowie Gebiete, die keine eigene Promotionsabteilung besitzen (vorhanden für Nord- Amerika, China und Japan) [PLCopen Europe].

PC 2 – Common Training.
PC 2 erstellte Unterlagen für IEC 61131-3 Trainings, welche von akkreditierten Trainingunternehmen durchgeführt werden können [PLCopen Europe].

PC 3 – Promotion North America.
Die Gruppe ist für Promotion- Aktivitäten in Nordamerika verantwortlich [PLCopen North America].

PC 4 – Promotion Japan.
PC 4 ist für Promotion- Aktivitäten in Japan zuständig und arbeitet eng mit der dortigen lokalen PLCopen Niederlassung zusammen [PLCopen Japan].

PC 4 – Promotion China.
PC 5 ist für Promotion- Aktivitäten in China zuständig und arbeitet eng mit der dortigen lokalen PLCopen Niederlassung zusammen [PLCopen China].

2 Bausteine der IEC 61131-3

In diesem Kapitel werden die wesentlichen Sprachelemente der Norm IEC 61131-3 in ihrer Bedeutung und Verwendung erläutert. Dabei werden aufeinander aufbauende praxisorientierte Beispiele verwendet.

Der Leser wird in die Begriffe und Denkweise der IEC 61131-3 eingeführt. Unter Verzicht auf die formale Sprachdefinition der Norm [DIN EN IEC 61131-3] werden die grundlegenden Zusammenhänge und Konzepte ausführlich erläutert.

Im ersten Abschnitt dieses Kapitels erfolgt der „kompakte" Einstieg in die Begriffswelt mit Hilfe eines Beispiels, das die wichtigsten Sprachelemente der Norm beinhaltet und gleichzeitig einen Überblick über die Methodik der SPS-Programmierung nach IEC 61131-3 bietet.

Dabei bildet der Begriff „*POE*" (*Programm-Organisationseinheit*) einen wichtigen roten Faden zum Verständnis der neuen Sprachkonzepte.

Für die Formulierung von Beispielen wird in diesem Kapitel die Programmiersprache Anweisungsliste (AWL) gewählt, da diese Notation den meisten SPS-Programmierern vertraut ist. Sie besitzt im europäischen SPS-Markt die größte Verbreitung und erschließt sich dem Leser schnell durch ihre einfache Syntax.

Die Programmiersprache AWL selbst wird in Abschn. 4.1 erläutert.

2.1 Einstieg in die neue Norm

Die IEC 61131-3 umfasst nicht nur die SPS-Programmiersprachen selbst, sondern bietet umfassende Konzepte und Richtlinien zum Aufbau eines SPS-Projekts.

Ziel dieses Abschnitts ist es, einen raschen Überblick über wesentliche Begriffe zu geben, ohne dabei auf Einzelheiten einzugehen. Ein gemeinsames Beispiel ergänzt die Erklärungen. Detaillierte Informationen sind den sich anschließenden Abschnitten und Kapiteln zu entnehmen.

2.1.1 Aufbau von Bausteinen

POEs entsprechen den *Bausteinen* bei bisherigen Programmiersystemen. POEs können sich gegenseitig mit oder ohne Parameter aufrufen. Eine POE bildet die kleinste unabhängige Software-Einheit des Anwenderprogramms, wie es ihr Name bereits ausdrückt.

Es gibt drei Arten von POEs: *Funktion (FUN)*, *Funktionsbaustein (FB)* und *Programm (PROG)*, in dieser Reihenfolge mit zunehmender Funktionalität versehen. FUNs unterscheiden sich von FBs vor allem dadurch, dass sie bei gleichen Eingangswerten immer dasselbe Ergebnis (als Funktionswert bzw. Ausgangswerte) zurückliefern, d.h. kein „Gedächtnis“, sprich keine Zustandsinformationen besitzen dürfen. Dagegen arbeiten FBs jeweils auf einem eigenen Datensatz, können sich also Zustandsinformationen „merken“ (*Instanzbildung*). Programme (PROG) bilden den „Kopf“ eines SPS-Anwenderprogramms und besitzen die Möglichkeit, auf SPS-Peripherie zuzugreifen bzw. diese den übrigen POEs zur Verfügung zu stellen.

In der IEC 61131-3 sind die Aufrufschnittstellen und das Verhalten häufig benötigter *Standard-Funktionen (Std.-FUN)* wie für Arithmetik oder Vergleiche sowie *Standard-Funktionsbausteine (Std.-FB)* wie Zeiten oder Zähler fest vordefiniert.

Deklarationen von Variablen.
Die IEC 61131-3 benutzt *Variablen*, um Informationen zu speichern und zu verarbeiten. Variablen entsprechen den bisher in SPS-Systemen verwendeten Merkern. Allerdings muss ihr Speicherort nicht mehr vom Anwender manuell vergeben werden (Vergabe absoluter Adressen), sondern sie werden vom Programmiersystem automatisch verwaltet und besitzen einen fest zugeordneten *Datentyp*.

Die IEC 61131-3 gibt mehrere Datentypen vor (Bool, Byte, Integer, ...), die sich beispielsweise durch die Anzahl der Bits und Vorzeichenbehandlung unterscheiden. Es können vom Anwender auch eigene Datentypen definiert werden (Strukturen, Felder).

Variablen können auf eine physikalische Adresse abgebildet werden und eine Batteriepufferung (für Stromausfall) erhalten.

Variablen besitzen unterschiedliche Ausprägungen. Sie können außerhalb einer POE definiert (deklariert) und programmweit benutzt werden, als Aufrufparameter in die POE eingebracht werden oder lokal für die POE von Bedeutung sein. Dazu werden sie zur Deklaration in Variablenarten eingeteilt. Sämtliche in einer POE benutzten Variablen sind im Deklarationsteil der POE bekanntzugeben.

Der Deklarationsteil einer POE kann unabhängig von der verwendeten Programmiersprache einheitlich in textueller Form erstellt werden. Teile davon (Ein- und Ausgangsparameter der POE) sind auch grafisch darstellbar.

```
VAR_INPUT                          (* Eingangsvariable *)
  GueltigFlag   : BOOL;            (* binärer Wert *)
END_VAR
VAR_OUTPUT                         (* Ausgangsvariable *)
  Drehzahl      : REAL;            (* Gleitpunktwert *)
END_VAR
VAR RETAIN                         (* lokale Variable, batteriegepuffert *)
  MotorNr       : INT;             (* Ganzzahl mit Vorzeichen *)
  MotorName     : STRING [10];     (* Zeichenfolge für Text *)
  NotAus AT %IX2.0 : BOOL;         (* Eingangsbit 2.0 der Peripherie *)
END_VAR
```

Bsp. 2.1. Beispiel für typische Variablendeklarationen einer POE

Bsp. 2.1 zeigt, dass in der dazugehörigen POE ein Integerwert (16 Bit inkl. Vorzeichen) mit Namen MotorNr verwendet wird und ein 10 Zeichen langer Text MotorName zur Verfügung steht. Die Variable NotAus wird an den physikalischen SPS-Signaleingang 2.0 gelegt (AT). Diese drei Variablen sind nur in dieser POE bekannt (lokal) und können dort manipuliert werden, nach einem Stromausfall bleiben ihre Werte erhalten (RETAIN). Der Wert der Eingangsvariablen GueltigFlag wird von der aufrufenden POE zur Verfügung gestellt und besitzt die booleschen Werte FALSE oder TRUE. Die POE liefert als Ausgangsparameter den Gleitpunktwert von Drehzahl zurück.

Die booleschen Werte TRUE und FALSE können auch gleichwertig mit „1“ und „0“ bezeichnet werden.

Anweisungsteil einer POE.
Der Anweisungsteil folgt dem Deklarationsteil und beschreibt die Befehle, welche die SPS ausführen soll.

Eine POE wird in einer der textuellen Programmiersprachen Anweisungsliste (AWL) oder Strukturierter Text (ST) bzw. in den grafischen Darstellungen Kontaktplan (KOP) oder Funktionsbausteinsprache (FBS) programmiert. Während in AWL maschinennah programmiert werden kann, bietet die IEC 61131-3 mit ST eine höhere Programmiersprache. KOP ist für boolesche (binäre) Verknüpfungsschaltungen geeignet. Mit FBS können sowohl boolesche (binäre) als auch arithmetische Verknüpfungen in grafischer Darstellung programmiert werden.

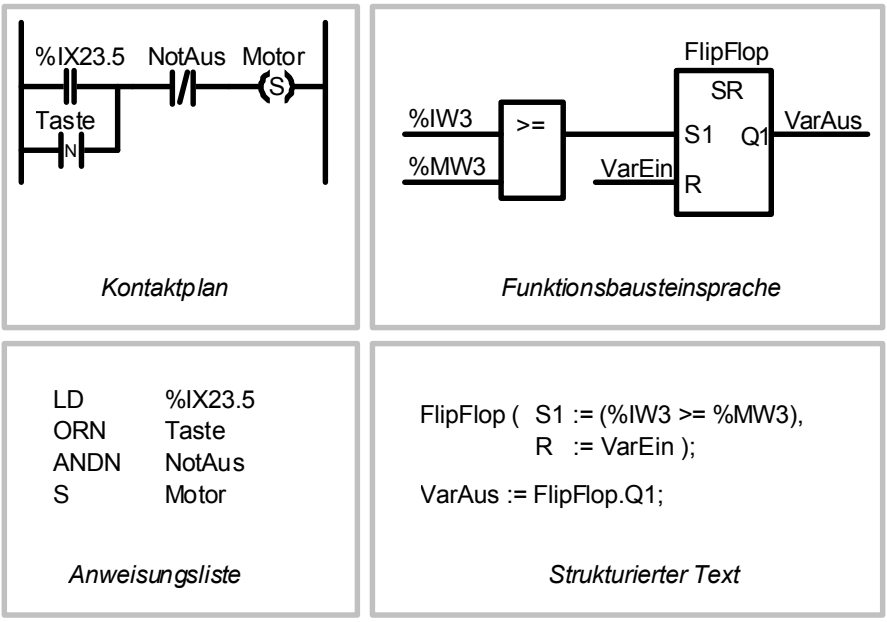


Abb. 2.1. Einfache Beispiele zu den Programmiersprachen KOP, FBS, AWL und ST. Die Darstellungen in KOP und AWL sowie in FBS und ST sind jeweils zueinander äquivalent.

Zusätzlich kann die Ablaufsprache (AS) verwendet werden, um die Struktur (den Aufbau) eines SPS-Programms zu beschreiben, indem die sequentiellen und parallelen Abläufe dargestellt werden. Die darin enthaltenen AS-Teilabschnitte (Schritte und Transitionen) können voneinander unabhängig in einer der IEC 61131-3-Sprachen formuliert werden.

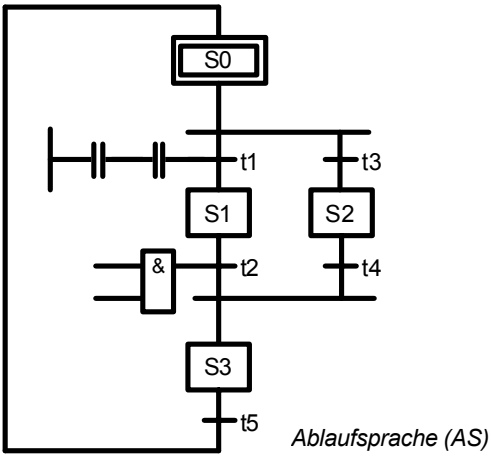


Abb. 2.2. Schematisches Beispiel zur Darstellungsart AS. Die Ausführungsteile zu den Schritten (S0 bis S3) und Transitionsbedingungen (t1 bis t5) werden in der gewünschten Programmiersprache erstellt.

Abb. 2.2 zeigt ein AS-Beispiel: Die Schritte S0, S1 und S3 werden sequentiell ausgeführt, alternativ zu S1 kann S2 zur Ausführung kommen. Die Transitionen t1 bis t5 stellen die Freischaltebedingungen dar, um von einem Schritt zum nächsten zu gelangen.

2.1.2 Einführungsbeispiel in AWL

Im Folgenden wird das Beispiel eines IEC 61131-3-Programms vorgestellt, für das in Abb. 2.3 zunächst seine POE-Aufrufhierarchie charakterisiert wird.

Dieses Beispiel ist nicht als ausführbares Programm ausformuliert, sondern dient lediglich zum Aufzeigen der POE-Struktur.