

# Inhalt

<b>1</b>	<b>Vorbemerkung</b>	<b>9</b>
<b>2</b>	<b>Einleitung und Begriffe</b>	<b>13</b>
2.1	Übersicht Automatisierungsarchitekturen	21
2.2	Funktionale Anforderungen an die Programmiersprache	22
2.2.1	Benötigter Befehlsumfang	24
2.3	Nicht-funktionale Anforderungen an die Programmiersprache	25
<b>3</b>	<b>Übersicht über die IEC 61131-3</b>	<b>27</b>
3.1	Struktur eines SPS-Programms	36
3.2	Vorgehensweise bei der Erstellung von Automatisierungssoftware	37
<b>4</b>	<b>Applikationsbeispiel Sortieranlage</b>	<b>41</b>
4.1	Grobe Struktur der Sortieranlage und dynamischer Ablauf	42
4.2	Beschreibung der einzelnen Stationen	43
4.2.1	Materiallager	44
4.2.2	Stempel	45
4.2.3	Sortierstrecke	46
<b>5</b>	<b>Vorgehen bei der Erstellung modularer Applikationssoftware</b>	<b>49</b>
5.1	Wiederverwendung und Modularität	49
5.1.1	Vorgehensweise zur Identifikation von wieder verwendbaren Modulen am Beispiel der Sortieranlage	51
5.1.2	Schnittstellen	58
5.1.3	Umsetzung einer modularen Struktur in CoDeSys	64

<b>6</b>	<b>Klassische IEC 61131-3 Sprachmittel</b>	<b>69</b>
6.1	Ausgewählte Sprachen der klassischen IEC 61131-3	69
6.1.1	Deklarationsteil	69
6.1.2	Anweisungsteil in Ablaufsprache (AS)	72
6.1.3	Anweisungsteil in Strukturiertem Text (ST)	79
6.1.4	Vergleich von ST und AS bezüglich des Zeitverhaltens	84
<b>7</b>	<b>Objektorientierte Erweiterung der IEC 61131-3</b>	<b>87</b>
7.1	Neue Sprachmittel und Schlüsselwörter	87
7.2	Der erste Schritt zur Objektorientierung	89
7.2.1	Aufruf von Operationen eines Objekts	90
7.3	Der zweite Schritt zur Objektorientierung: Klassen	91
7.3.1	Definition von Klassen	91
7.3.2	Erstellung von Varianten durch Vererbung	93
7.4	Standardisierung durch Interfaces	93
7.5	Ausblick auf die weitere Entwicklung von CoDeSys	94
7.6	Umsetzung der Module des Applikationsbeispiels	94
7.6.1	Definition der Klassen	94
7.6.2	Definition der Schnittstellen	97
7.6.3	Programmierung der Module	99
<b>8</b>	<b>Projektstruktur und Prozessanbindung</b>	<b>107</b>
8.1	Verknüpfung der Applikation mit einer Steuerung	109
8.1.1	Verbindung zu einem angeschlossenen Gerät	109
8.1.2	Konfiguration der Task einer Steuerung	110
8.1.3	Bausteine in Applikationen	114
8.2	Ein-/Ausgabesystem konfigurieren	116
8.3	Verknüpfung von Applikationsvariablen mit Prozessvariablen	117
8.3.1	Direkte Adressierung	118
8.3.2	Symbolische Adressierung	119
8.3.3	Instanzbezogene Adressierung (VAR_CONFIG Konzept)	119
8.3.4	Zentrale Zuordnung über den Gerätemanager (Mapping)	122
8.4	Konfiguration und Prozessanbindung am Beispiel der Stempelanlage	123
8.4.1	Anlegen der Hardwarekonfiguration	123
8.4.2	Verknüpfung der Variablen der Applikation mit den I/O-Adressen der Klemmen über den Gerätedialog	126

---

<b>9</b>	<b>Verwendung von Bibliotheken</b>	<b>129</b>
9.1	Einbinden und Nutzen von Bibliotheken	129
9.2	Versionen von Bibliotheken	131
9.3	Namensräume von Bibliotheken	132
9.4	Beispiel für die Verwendung eines Bibliotheksbausteins	132
9.5	Bibliotheken erstellen	133
9.6	Mitgelieferte Bibliotheken	133
9.7	Geräte-Bibliothek	134
<b>10</b>	<b>Diagnose und Störmeldungen</b>	<b>137</b>
10.1	Fehlererkennung	138
10.2	Maßnahmen und Überführung in einen fehlerfreien Zustand	140
10.2.1	Symptomale Fehlerbehandlung	140
10.2.2	Ursächliche Fehlerbehandlung	142
10.3	Fehlerbehandlung in einem modularen System	143
<b>11</b>	<b>Visualisierung und Objektorientierung</b>	<b>147</b>
11.1	Der Editor	147
11.2	Visualisierung für Klassen anhand des Applikationsbeispiels	148
<b>12</b>	<b>Werkstofforientierter Ablauf – Ein alternativer Ansatz zur Umsetzung der Applikation</b>	<b>153</b>
<b>13</b>	<b>Anhang</b>	<b>159</b>
13.1	Strukturierter Text	159
13.2	Ablaufsprache (AS)	160
13.3	Funktionsbausteinsprache (FBS)	164
<b>14</b>	<b>Glossar</b>	<b>167</b>
<b>15</b>	<b>Abkürzungen</b>	<b>179</b>
<b>16</b>	<b>Referenzen</b>	<b>181</b>
<b>17</b>	<b>Stichwortverzeichnis</b>	<b>185</b>