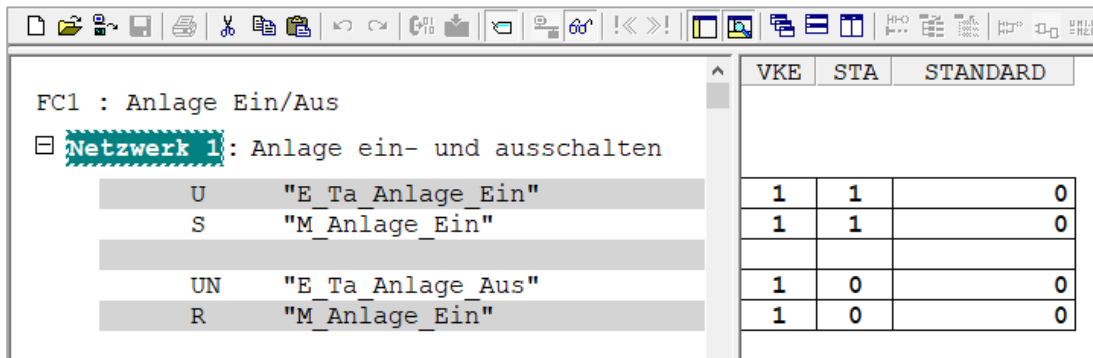


Schulung «AWL-Basic-Kurs»

STEP7-classic



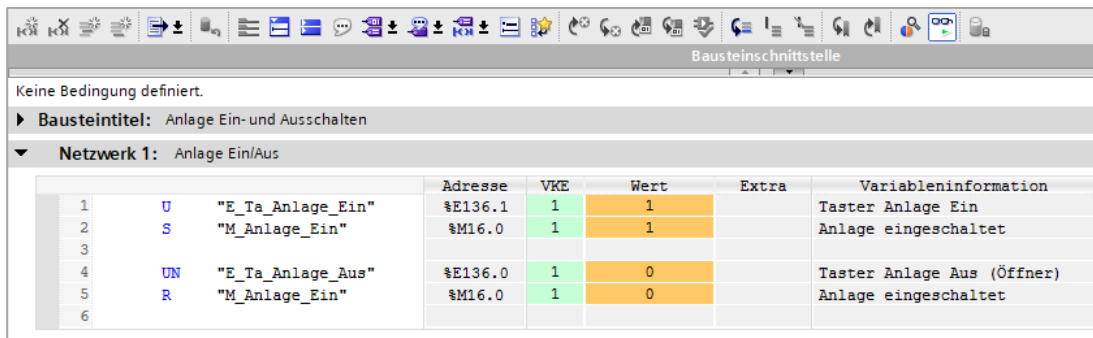
The screenshot shows the STEP7-classic interface. On the left, a network diagram is displayed with the following rungs:

- Rung 1: U "E_Ta_Anlage_Ein"
- Rung 2: S "M_Anlage_Ein"
- Rung 3: UN "E_Ta_Anlage_Aus"
- Rung 4: R "M_Anlage_Ein"

On the right, a table shows the status of the network:

| VKE | STA | STANDARD |
|-----|-----|----------|
| 1 | 1 | 0 |
| 1 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 0 | 0 |

TIA-Portal



The screenshot shows the TIA-Portal 'Bausteinschnittstelle' (Component Interface) for the 'Anlage Ein- und Ausschalten' network. The interface displays the following data:

| Adresse | VKE | Wert | Extra | Variableninformation |
|--------------------------------|-----|------|-------|----------------------------|
| 1 U "E_Ta_Anlage_Ein" §E136.1 | 1 | 1 | | Taster Anlage Ein |
| 2 S "M_Anlage_Ein" §M16.0 | 1 | 1 | | Anlage eingeschaltet |
| 3 | | | | |
| 4 UN "E_Ta_Anlage_Aus" §E136.0 | 1 | 0 | | Taster Anlage Aus (Öffner) |
| 5 R "M_Anlage_Ein" §M16.0 | 1 | 0 | | Anlage eingeschaltet |
| 6 | | | | |

- 1 Einführung AWL
- 2 AWL-Operationen
- 3 Anhang

Inhaltsverzeichnis

| | | |
|------------|--|-----------|
| 1 | Einführung AWL | 5 |
| 1.1 | Grundlagen | 5 |
| 1.1.1 | Programmiersprachen | 5 |
| 1.1.2 | Compiler | 5 |
| 1.1.3 | Unterschied S7-300 und S7-400 | 5 |
| 1.1.4 | Aufbau AWL-Befehlszeile | 6 |
| 1.1.5 | Oberflächensprache und Mnemonik | 7 |
| 1.1.6 | Adressierungsverfahren | 9 |
| 1.2 | Übung P1 – AWL-Editor im STEP7-classic | 10 |
| 1.2.1 | Vorlageprojekt in CPU laden | 11 |
| 1.2.2 | Zwei neue Bausteine OB1 und FC1 einfügen | 13 |
| 1.2.3 | FC1 im OB1 aufrufen | 15 |
| 1.2.4 | Anlage Ein/Aus programmieren und testen | 16 |
| 1.2.5 | Quelle für FC1 Importieren und übersetzen | 17 |
| 1.2.6 | Bausteinkonsistenz prüfen | 19 |
| 1.2.7 | Befehl «NOP 0» | 22 |
| 1.3 | Übung P2 – AWL-Editor im TIA-Portal | 24 |
| 1.3.1 | Vorlageprojekt in CPU laden | 25 |
| 1.3.2 | Externe Quelle hinzufügen und FC im OB1 aufrufen | 26 |
| 2 | AWL-Operationen | 30 |
| 2.1 | Übersicht | 30 |
| 2.1.1 | Hilfe zu AWL in STEP7-classic | 31 |
| 2.1.2 | Hilfe zu AWL im TIA-Portal | 32 |
| 2.2 | CPU-Register | 33 |
| 2.2.1 | S7-1500 | 33 |
| 2.3 | Statuswort | 34 |
| 2.3.1 | Wann werden Operationen durchgeführt | 35 |
| 2.3.2 | Erstabfrage | 36 |
| 2.3.3 | ODER-Flag | 36 |
| 2.3.4 | BIE-Bit | 36 |
| 2.3.5 | A0 / A1 abhängige Verknüpfungen | 37 |

| | | |
|-------------|---|-----------|
| 2.3.6 | Überlauf abhängige Verknüpfungen..... | 37 |
| 2.4 | Klammerstack..... | 38 |
| 2.4.1 | Funktionsweise | 38 |
| 2.4.2 | Schreiben in Klammerstack | 39 |
| 2.4.3 | Lesen aus Klammerstack..... | 39 |
| 2.5 | Übung P3 – OR, STA, VKE und /ER im STEP7-classic..... | 40 |
| 2.5.1 | Quelldatei einfügen, übersetzen und laden | 41 |
| 2.5.2 | Befehl CLR und SET | 43 |
| 2.5.3 | Statusbit STA und VKE | 44 |
| 2.5.4 | Statusbit /ER | 45 |
| 2.5.5 | Statusbit OR..... | 48 |
| 2.5.6 | Statusanzeige zurücksetzen..... | 49 |
| 2.6 | Übung P4 – Status binäre Verknüpfungen im TIA-Portal | 50 |
| 2.6.1 | Quelldatei einfügen, übersetzen und laden..... | 51 |
| 2.6.2 | Statusbit durch platzieren des Cursors anzeigen | 52 |
| 2.7 | Akkumulatoren | 53 |
| 2.7.1 | AKKU1..... | 54 |
| 2.7.2 | AKKU2 | 54 |
| 2.7.3 | AKKU3 und AKKU4 | 54 |
| 2.7.4 | Laden..... | 55 |
| 2.7.5 | Transferieren | 56 |
| 2.8 | Übung P5 – Operationen mit AKKU1 und AKKU2..... | 58 |
| 2.8.1 | Quelldatei einfügen, übersetzen und laden..... | 59 |
| 2.8.2 | Laden und Transferieren | 59 |
| 2.8.3 | Akkumulator-Operationen..... | 62 |
| 2.8.4 | Arithmetik-Anweisungen | 64 |
| 2.8.5 | Wortverknüpfungen..... | 66 |
| 2.8.6 | Schiebe- und Rotieroperationen | 68 |
| 2.9 | Vergleicher | 70 |
| 2.10 | Umwandler | 72 |
| 2.11 | S5-Timer..... | 73 |
| 2.11.1 | Zeitvorgabe S5TIME | 73 |
| 2.11.2 | Datenablage im Systemspeicher | 73 |
| 2.12 | Übung P6 – Vergleicher, Umwandler und S5-Timer | 74 |
| 2.12.1 | INT-Variablen vergleichen..... | 75 |

| | | |
|-------------|---|------------|
| 2.12.2 | Umwandlung von negativen Zahlen..... | 76 |
| 2.12.3 | Analogwert normieren [0..27648 → 0..500kg] | 77 |
| 2.12.4 | Verzögerung Gewicht grösser als 250kg | 78 |
| 2.13 | Sprünge | 80 |
| 2.13.1 | VKE abhängige Sprungoperationen | 80 |
| 2.13.2 | A0 / A1 abhängige Sprungoperationen | 81 |
| 2.13.3 | Überlauf abhängige Sprungoperationen..... | 81 |
| 2.13.4 | BIE abhängige Sprungoperationen (AWL-Advanced-Kurs)..... | 81 |
| 2.13.5 | Spezielle Sprungoperationen (AWL-Advanced-Kurs) | 81 |
| 2.13.6 | Sprungziel (Marke) | 82 |
| 2.14 | Übung P7 – Analogwertverarbeitung mit Sprüngen | 83 |
| 2.14.1 | Quelldatei einfügen, übersetzen und laden..... | 84 |
| 2.14.2 | Analogwerte auf Maximum begrenzen | 84 |
| 2.14.3 | Umschaltung Gewichtsanzeige links oder rechts..... | 86 |
| 2.15 | Übung P8 – Standardbaustein Motor | 88 |
| 2.15.1 | Standardbaustein «FB_Stand_Motor» erstellen..... | 89 |
| 2.15.2 | Motor Transportband in FUP aufrufen und testen | 91 |
| 2.15.3 | Motor Stanze in AWL aufrufen und testen | 93 |
| 2.15.4 | Baustein aufruf Motor Transportband für AWL optimieren | 95 |
| 2.16 | Übung P9 – Schrittkette «Stanze» mit Sprungverteiler | 97 |
| 2.16.1 | Standardbaustein «FB_Stand_Motor» erweitern..... | 98 |
| 2.16.2 | Schrittkette im FB100 mit Sprungverteiler programmieren | 100 |
| 2.17 | Übung P10 – Programm von FUP auf AWL umschreiben..... | 103 |
| 2.17.1 | FUP-Programm mit Simulation inkl. HMI testen | 104 |
| 2.17.2 | OB1 und OB123 in AWL programmieren..... | 105 |
| 3 | Anhang..... | 107 |
| 3.1 | Schlüsselwörter | 107 |
| 3.2 | Unterschied Gleitpunkt- und REAL Zahlen | 108 |
| 3.3 | Links | 109 |
| 3.4 | Handbücher | 109 |
| 3.5 | PDF | 109 |

1 Einführung AWL

1.1 Grundlagen

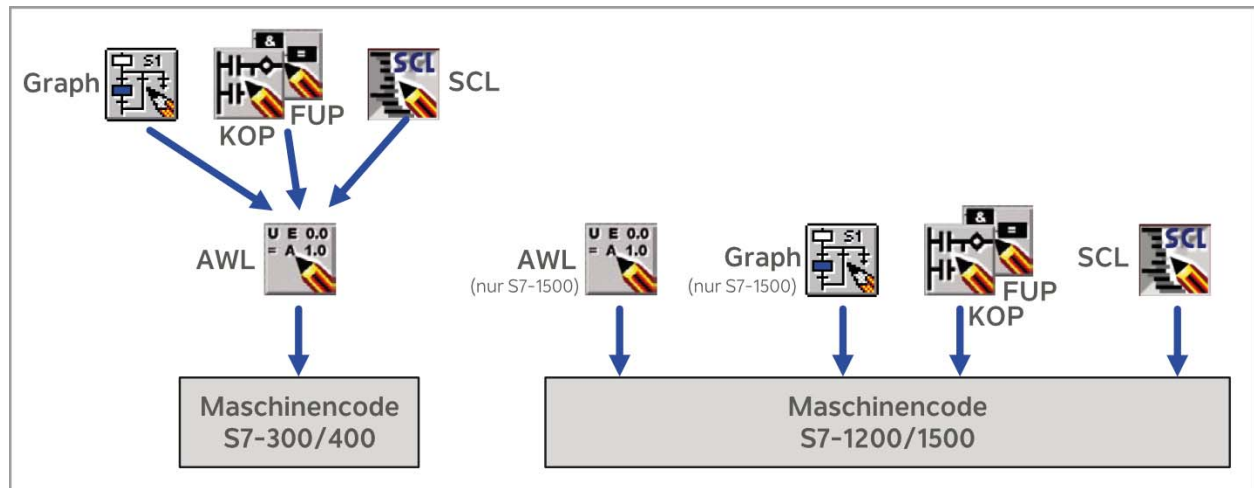
Die Programmiersprache AWL (Anweisungsliste) nach der Norm IEC DIN EN 61131-3 ist eine maschinennahe textbasierende Programmiersprache. In AWL werden einzelne Anweisungen in der Reihenfolge geschrieben wie die CPU diese dann abarbeiten soll.

Ein **quellorientierter Editor** ermöglicht die Erstellung von AWL-Quellen, die anschließend in Bausteine übersetzt werden können.

1.1.1 Programmiersprachen

| Systemfamilie | S7-200 | S7-300 | S7-400 | S7-1200 | S7-1500 |
|---------------|--------|--------|--------|---------|---------------|
| AWL | ✓ | ✓ | ✓ | ✗ | ✓ emuliert |
| KOP/FUP | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCL | ✗ | ✓ | ✓ | ✓ | ✓ |
| Graph | ✗ | ✓ | ✓ | ✗ | ✓ |
| CFC | ✗ | ✓ | ✓ | ✗ | ✓ |
| CEM | ✗ | ✗ | ✗ | ✗ | ✓ |

1.1.2 Compiler



1.1.3 Unterschied S7-300 und S7-400

Der interne Aufbau einer S7-300er CPU und einer S7-400er CPU ist unterschiedlich. Während die S7-400 das Programm direkt abarbeitet, wird das Programm der S7-300er beim Laden übersetzt. Man spricht deshalb bei der S7-400 von **ASIC-Maschinen** und bei der S7-300 von **Compiler-Maschinen**.