Ausbildungsunterlage für die durchgängige Automatisierungslösung Totally Integrated Automation (T I A)

MODUL B2

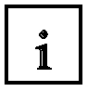
Analogwertverarbeitung

Diese Unterlage wurde von der Siemens AG, für das Projekt Siemens Automation Cooperates with Education (SCE) zu Ausbildungszwecken erstellt.
Die Siemens AG übernimmt bezüglich des Inhalts keine Gewähr.

Wir danken der Fa. Michael Dziallas Engineering und den Lehrkräften von beruflichen Schulen sowie weiteren Personen für die Unterstützung bei der Erstellung der Unterlage

SEITE:

Die folgenden Symbole führen durch dieses Modul:

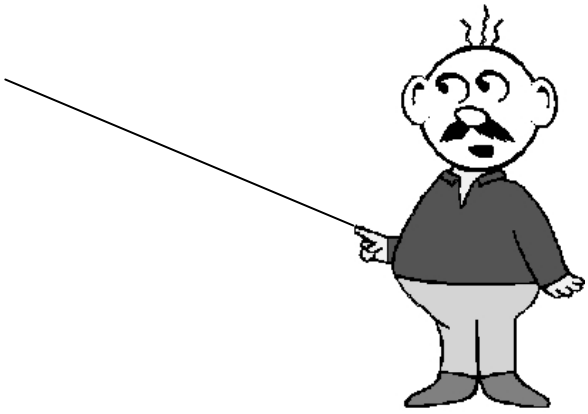


Information

Programmierung

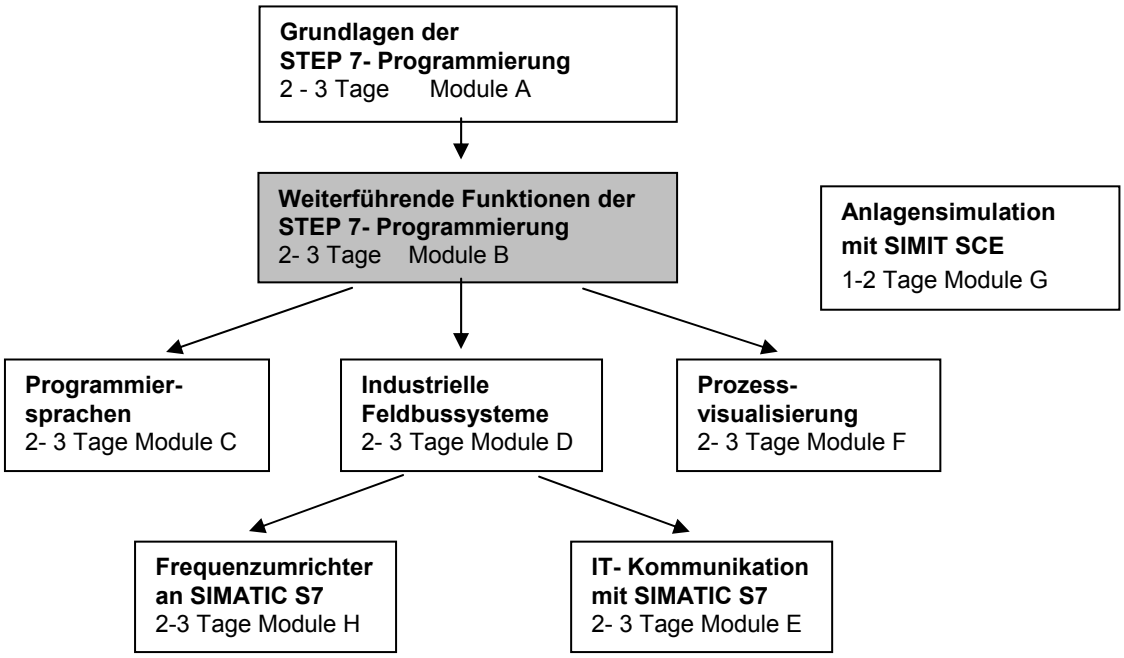
Beispielaufgabe

Hinweise



1. VORWORT

Das Modul B2 ist inhaltlich der Lehreinheit **‚Weiterführende Funktionen der STEP 7-Programmierung'** zugeordnet.



Lernziel:

Der Leser soll in den folgenden Schritten lernen wie Analogwerte in einer SIMATIC S7 eingelesen, verarbeitet und ausgegeben werden.

- Analoge Signale
- Datentypen in STEP 7
- Mathematische Operationen
- Umwandeln von Datentypen in STEP 7
- Analogwert einlesen und normieren
- Analogwert normieren und ausgeben

Voraussetzungen:

Für die erfolgreiche Bearbeitung dieses Moduls wird folgendes Wissen vorausgesetzt:

- Kenntnisse in der Handhabung von Windows
- Grundlagen der SPS- Programmierung mit STEP 7 (z.B. Modul A3 - ‚Startup' SPS- Programmierung mit STEP 7)

Benötigte Hardware und Software

1 PC, Betriebssystem Windows XP Professional mit SP2 oder SP3 / Vista 32 Bit Ultimate und Business / Server 2003 SP2 mit 600MHz (nur XP) / 1 GHz und 512MB (nur XP) / 1 GB RAM, freier Plattenspeicher ca. 650 - 900 MB, MS-Internet-Explorer 6.0 und Netzwerkkarte

2 Software STEP7 V 5.4

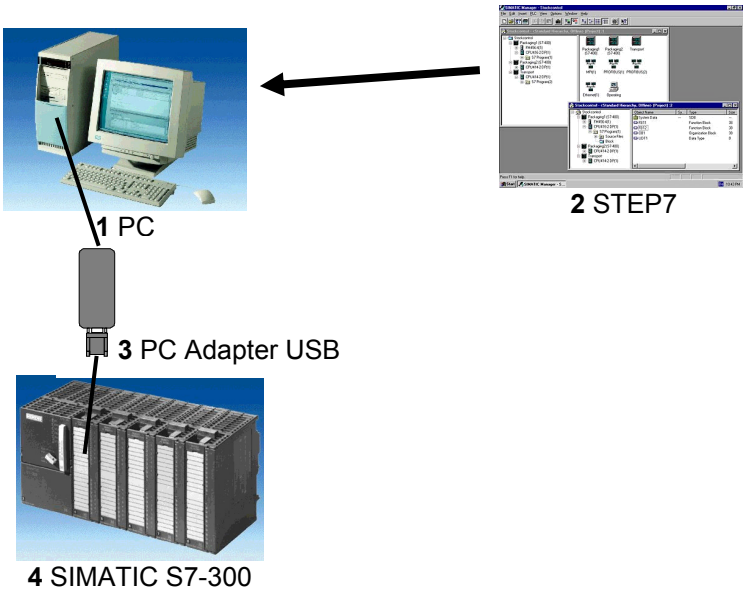
3 MPI- Schnittstelle für den PC (z.B. PC Adapter USB)

4 SPS SIMATIC S7-300 mit mind. einer analogen Ein-/Ausgabebaugruppe, bei der an einem Analogwerteingang ein Potentiometer oder ein anderer analoger Signalgeber angeschlossen ist.

Außerdem muss an mindestens einem Analogausgang eine Analogwertanzeige angeschlossen sein.

Beispielkonfiguration:

- Netzteil: PS 307 2A
- CPU: CPU 314
- Digitale Eingänge: DI 16x DC24V
- Digitale Ausgänge: DO 16x DC24V / 0,5 A
- Analoge Ein-/ Ausgänge: AI 4/ AO 2 x 8Bit



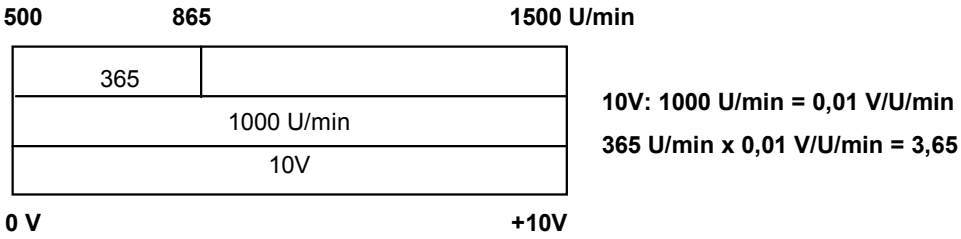
2. ANALOGE SIGNALE

Im Gegensatz zu einem binären Signal, das nur die beiden Signalzustände „Spannung vorhanden +24V„ und „Spannung nicht vorhanden 0V„ annehmen kann, können analoge Signale innerhalb eines bestimmten Bereichs beliebig viele Werte annehmen. Ein typisches Beispiel für einen Analoggeber ist ein Potentiometer. Je nach Stellung des Drehknopfes kann hier bis zum Maximumwert ein beliebiger Widerstand eingestellt werden.

Beispiele für analoge Größen in der Steuerungstechnik:

- Temperatur -50 ... +150°C
- Durchfluss 0 ... 200l/min
- Drehzahl 500 ... 1500 U/min
- u.s.w.

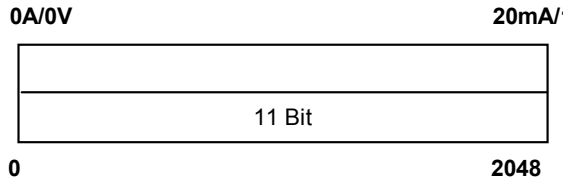
Diese Größen werden mit Hilfe eines Messumformers in elektrische Spannungen, Ströme oder Widerstände umgewandelt. Soll z.B. eine Drehzahl erfasst werden, kann der Drehzahlbereich von 500 ... 1500 U/min über einen Messumformer in einen Spannungsbereich von 0 ... +10V umgewandelt werden. Bei einer gemessenen Drehzahl von 865 U/min würde dann der Messumformer einen Spannungswert von + 3,65 V ausgeben.



Diese elektrische Spannungen, Ströme oder Widerstände werden dann an einer Analogbaugruppe angeschlossen die dieses Signal digitalisiert.

Hinweis: Einige Analogbaugruppen können dabei verschiedene Signalarten verarbeiten. Dies muss dann in der Hardwarekonfiguration eingestellt werden. Beachten Sie hier bitte die Hinweise in den Gerätehandbüchern.

Werden analoge Größen mit einer SPS verarbeitet, so muss der eingelesene Spannungs-, Strom- oder Widerstandswert in eine digitale Information umgewandelt werden. Diese Wandlung bezeichnet man als Analog - Digital - Wandlung (A/D- Wandlung). Dies bedeutet, das z.B. der Spannungswert von 3,65V in eine Reihe von Binärstellen als Information hinterlegt wird. Je mehr Binärstellen hierbei für die digitale Darstellung verwendet werden , umso feiner wird die Auflösung. Hätte man z.B. für den Spannungsbereich 0 ... +10V nur 1 Bit zur Verfügung, könnte nur eine Aussage getroffen werden, ob die gemessene Spannung im Bereich 0 .. +5V oder im Bereich +5V ... +10V liegt. Mit 2 Bit kann der Bereich schon in 4 Einzelbereiche unterteilt werden, also 0 ... 2,5 / 2,5 ... 5 / 5 ... 7,5 / 7,5 ... 10V. Gängige A/D- Wandler in der Steuerungstechnik wandeln mit 8 oder 11 Bit.
Dabei haben Sie mit 8 Bit 256 Einzelbereiche und mit 11 Bit eine Auflösung von 2048 Einzelbereichen.



10V: 2048 = 0,0048828
→ es können Spannungs-unterschiede <5mV erkannt werden

3. DATENTYPEN IN STEP7

In der SIMATIC S7 gibt es eine Vielzahl unterschiedlicher Datentypen, mit denen unterschiedliche Zahlenformate dargestellt werden. Im Folgenden wird eine vollständige Auflistung der elementaren Datentypen gegeben.

Typ und Beschreibung	Größe in Bits	Format-option	Bereich und Zahlendarstellung niedrigster bis höchster Wert	Beispiel
BOOL (Bit)	1	Bool-Text	TRUE/FALSE	TRUE
BYTE (Byte)	8	Hexadezimal	B#16#0 bis B#16#FF	B#16#10
WORD (Wort)	16	Dualzahl	2#0 bis 2#1111_1111_1111_1111	2#0001_0000_0000_0000
		Hexadezimalzahl	W#16#0 bis W#16#FFFF	W#16#1000
		BCD	C#0 bis C#999	C#998
		Dezimalzahl (o.V.)	B#(0,0) bis B#(255,255)	B#(10,20)
DWORD (Doppelwort)	32	Dualzahl	2#0 bis 2#1111_1111_1111_1111_1111_1111_1111_1111	2#1000_0001_0001_1000_1011_1011_0111_1111
		Hexadezimalzahl	DW#16#0000_0000 bis DW#16#FFFF_FFFF	DW#16#00A2_1234
		Dezimalzahl (o.V.)	B#(0,0,0,0) bis B#(255,255,255,255)	B#(1,14,100,120)
INT (Ganzzahl)	16	Dezimalzahl	-32768 bis 32767	1
DINT (Ganzzahl,32 bit	32	Dezimalzahl	L#-2147483648 bis L#2147483647	L#1
REAL (Gleitpunktzahl)	32	IEEE Gleitpunktzahl	Oberer Grenze: +/-3.402823e+38 Untere Grenze: +/-1.175495e-38	1.234567e+13
S5TIME (Simatic-Zeit)	16	S7-Zeit in Schritten von 10 ms	S5T#0H_0M_0S_10MS bis S5T#2H_46M_30S_0MS und S5T#0H_0M_0S_0MS	S5T#0H_1M_0S_0MS S5TIME#1H_1M_0S_0MS
TIME (IEC-Zeit)	32	IEC-Zeit in Schritten von 1ms, Ganzzahl mit Vorzeichen	-T#24D_20H_31M_23S_648MS bis T#24D_20H_31M_23S_647MS	T#0D_1H_1M_0S_0MS TIME#0D_1H_1M_0S_0MS
DATE (IEC-Datum)	16	IEC-Datum in Schritten von 1 Tag	D#1990-1-1 bis D#2168-12-31	DATE#1994-3-15
TIME_OF_DAY (Uhrzeit)	32	Uhrzeit in Schritten von 1ms	TOD#0:0:0.0 bis TOD#23:59:59.999	TIME_OF_DAY#1:10:3.3
CHAR (Zeichen)	8	ASCII-Zeichen	´A´, ´B´ usw.	´B´

Hinweis: Für die Analogwertverarbeitung spielen die Datentypen **‚INT'** und **‚REAL'** eine große Rolle, da eingelesene Analogwerte als Ganzzahlen im Format **‚INT'** vorliegen und für eine exakte Weiterbearbeitung wegen der Rundungsfehler bei **‚INT'** nur Gleitpunktzahlen **‚REAL'** in Frage kommen.'

4. RECHENOPERATIONEN

4.1 RECHNEN MIT DEN FESTPUNKTZAHLEN (INT UND DINT)

Mit Festpunktzahlen sind die mathematischen Grundoperationen Addition, Subtraktion, Multiplikation und Division möglich. Dabei bleiben jedoch die Stellen hinter dem Komma unberücksichtigt, was zu Rundungsfehlern bei der Division führt.

Operation	Größe in Bit	Funktion
+I	16	Addiert den Inhalt des niederwertigen Worts der AKKUs 1 und 2 und speichert das Ergebnis im niederwertigen Wort von AKKU 1.
-I	16	Subtrahiert den Inhalt des niederwertigen Worts von AKKU 1 vom Inhalt des niederwertigen Worts von AKKU 2 und speichert das Ergebnis im niederwertigen Wort von AKKU 1.
*I	16	Multipliziert den Inhalt der niederwertigen Wörter der AKKUs 1 und 2 und speichert das Ergebnis (32 Bit) in AKKU 1.
/I	16	Dividiert den Inhalt des niederwertigen Worts von AKKU 2 durch den Inhalt des niederwertigen Worts von AKKU 1. Das Ergebnis wird im niederwertigen Wort von AKKU 1 gespeichert. Der Divisionsrest wird im höherwertigen Wort von AKKU 1 gespeichert.
+D	32	Addiert den Inhalt der AKKUs 1 und 2 und speichert das Ergebnis in AKKU 1.
-D	32	Subtrahiert den Inhalt von AKKU 1 vom Inhalt von AKKU 2 und speichert das Ergebnis in AKKU 1.
*D	32	Multipliziert AKKU 1 mit dem Inhalt von AKKU 2 und speichert das Ergebnis in AKKU 1.
/D	32	Dividiert den Inhalt von AKKU 2 durch den Inhalt von AKKU 1 und speichert den Quotienten in AKKU 1.
MOD	32	Dividiert den Inhalt von AKKU 2 durch den Inhalt von AKKU 1 und speichert den Divisionsrest als Ergebnis in AKKU 1.

4.2 RECHNEN MIT GLEITPUNKTZAHLEN (REAL)

Mit Gleitpunktzahlen lassen sich eine Vielzahl an mathematischen Operationen durchführen. Dabei werden in diesem Format immer auch die Nachkommastellen berücksichtigt.

Operation	Funktion
+R	Addiert die Gleitpunktzahlen (32 Bit, IEEE-FP) in den AKKUs 1 und 2 und speichert das 32-Bit-Ergebnis in AKKU 1.
-R	Subtrahiert die Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1 von der Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 2 und speichert das 32-Bit-Ergebnis in AKKU 1.
*R	Multipliziert die Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1 mit der Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 2 und speichert das 32-Bit-Ergebnis in AKKU 1.
/R	Dividiert die Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 2 durch die Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1. Das 32-Bit-Ergebnis wird in AKKU 1 gespeichert.
ABS	Bildet den Absolutwert einer Gleitpunktzahl (32-Bit, IEEE-FP) in AKKU 1. Das Ergebnis wird in AKKU 1 gespeichert. Die Operation wird ausgeführt, ohne die Statusbits zu berücksichtigen oder zu beeinflussen.
SQRT	Berechnet die Quadratwurzel der Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1 und speichert das 32 Bit-Ergebnis in AKKU 1.
SQR	Berechnet das Quadrat der Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1 und speichert das 32 Bit-Ergebnis in AKKU 1.
LN	Berechnet den natürlichen Logarithmus der Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1 und speichert das 32 Bit-Ergebnis in AKKU 1.
EXP	Berechnet den Exponentialwert der Gleitpunktzahl (32 Bit, IEEE-FP) zur Basis e und speichert das 32 Bit-Ergebnis in AKKU 1.
SIN	Berechnet den Sinus der Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1 und speichert das 32 Bit-Ergebnis in AKKU 1.
COS	Berechnet den Cosinus der Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1 und speichert das 32 Bit-Ergebnis in AKKU 1.
TAN	Berechnet den Tangens der Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1 und speichert das 32 Bit-Ergebnis in AKKU 1
ASIN	Berechnet den Arcussinus der Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1 und speichert das 32 Bit-Ergebnis in AKKU 1.
ACOS	Berechnet den Arcuscosinus der Gleitpunktzahl (32 Bit, IEEE-FP) in AKKU 1 und speichert das 32 Bit-Ergebnis in AKKU 1.
ATAN	Berechnet den Arcustangens der Gleitpunktzahl (32 Bit, IEEE-FP) in

4.3 DATENTYP- UMWANDLUNGSOPERATIONEN

Da oftmals die Zahlen nicht in den, für die Weiterverarbeitung notwendigen Zahlenformaten vorliegen, müssen diese häufig mit Hilfe der Umwandlungsoperationen angepasst werden.

Operation	Funktion
BTI	BCD wandeln in Ganzzahl (16 Bit). Diese Operation wandelt einen binär-codierten Dezimalwert in AKKU 1 in eine Ganzzahl (16 Bit) um.
BTD	BCD wandeln in Ganzzahl (32 Bit). Diese Operation wandelt einen binär-codierten Dezimalwert in AKKU 1 in eine Ganzzahl (32 Bit) um.
ITB	Ganzzahl (16 Bit) wandeln in BCD. Diese Operation wandelt eine Ganzzahl (16 Bit) im niederwertigen Wort von AKKU 1 in einen binär-codierten Dezimalwert um.
ITD	Ganzzahl (16 Bit) wandeln in Ganzzahl (32 Bit). Diese Operation wandelt eine Ganzzahl (16 Bit) im niederwertigen Wort von AKKU 1 in eine Ganzzahl (32 Bit) um.
DTB	Ganzzahl (32 Bit) wandeln in BCD. Diese Operation wandelt eine Ganzzahl (32 Bit) in AKKU 1 in einen binär-codierten Dezimalwert um.
DTR	Ganzzahl (32 Bit) wandeln in Gleitpunktzahl (32 Bit, IEEE-FP). Diese Operation wandelt eine Ganzzahl (32 Bit) in AKKU 1 in eine Gleitpunktzahl (32 Bit, IEEE-FP) um.
RND	Runden zur Ganzzahl Diese Operation rundet die umgewandelte Zahl zur nächsten Ganzzahl. Wenn der Bruch der umgewandelten Zahl genau zwischen einem geraden und einem ungeraden Ergebnis liegt, wählt die Operation das gerade Ergebnis.
RND+	Runden zur nächst höheren Ganzzahl Diese Operation rundet die umgewandelte Zahl zur kleinsten Ganzzahl, die größer als oder gleich der umgewandelten Gleitpunktzahl ist.
RND-	Runden zur nächst niederen Ganzzahl Diese Operation rundet die umgewandelte Zahl zur größten Ganzzahl, die kleiner als oder gleich der umgewandelten Gleitpunktzahl ist.
TRUNC	Runden mit Abschneiden Diese Operation wandelt den ganzzahligen Teil der Gleitpunktzahl um.

Hinweis: Im Fall der Analogwertverarbeitung liegt der Analogwert in **‚INT'** vor und soll für eine exakte Weiterbearbeitung wegen der Rundungsfehler bei **‚INT'** in eine Gleitpunktzahl **‚REAL'** umgewandelt werden. Da dies nicht direkt geht muss hier erst mit **‚ITD'** in **‚DINT'** und dann mit **‚DTR'** in **‚REAL'** umgewandelt werden.

5. ANALOGWERTE EINLESEN/AUSGEBEN

Analogwerte werden als Wortinformationen in die SPS eingelesen bzw. ausgegeben. Der Zugriff auf diese Worte geschieht mit den Anweisungen:

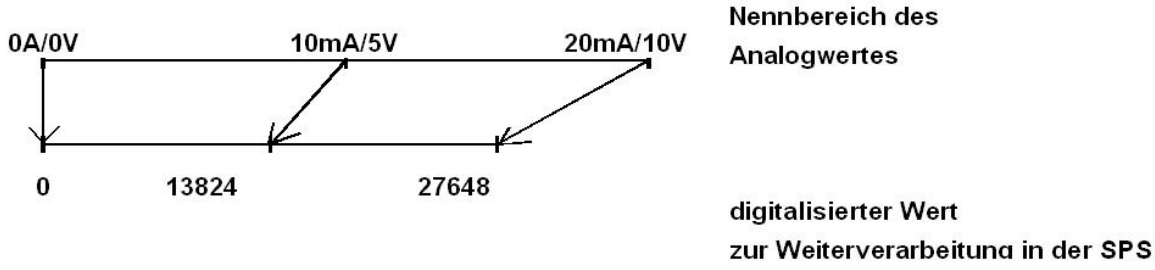
L	PEW x	für ‘Lade Analogeingangswort’
T	PAW x	Für ‘Transferiere Analogausgangswort’

Jeder Analogwert („Kanal„) belegt ein Peripherieeingangs- bzw. Peripherieausgangswort. Das Format ist **‚INT’** eine Integer- Ganzzahl.

Die Adressierung der Ein- bzw. Ausgangsworte richtet sich nach der Baugruppen- Anfangsadresse. Steckt die Analogbaugruppe auf Steckplatz 4, dann hat sie die Default- Anfangsadresse 256. Die Anfangsadresse jeder weiteren Analogbaugruppe erhöht sich je Steckplatz um 16.
Diese Default- Anfangsadresse kann in der Hardwarekonfigurationstabelle unter Detailansicht nachgesehen werden.

Die Adresse des ersten Analogeingangs wäre somit bei einer Analogbaugruppe auf Steckplatz 6 PEW 288, die des zweiten Analogeingangs PEW 290, die des ersten Analogausgangs folglich PAW 288 u.s.w..

Die Analogwerttransformation zur Weiterverarbeitung in der SPS (digitalisiert) ist bei Analogein- und Analogausgängen gleich.
Für die Baugruppe SM334, mit 4 Analogeingängen und 2 Analogausgängen, mit den Analogwertbereichen
0 bis 10V sowie 0 bis 20mA sehen die digitalisierten Wertebereiche wie folgt aus:



Diese digitalisierten Werte müssen häufig durch entsprechende Weiterverarbeitung in der SPS noch normiert werden.

5.1. ANALOGWERT EINLESEN UND NORMIEREN

Liegt ein Analogeingangswert als digitalisierter Wert vor, so muss dieser zumeist noch normiert werden, bevor er in der SPS weiterverarbeitet werden kann.
Ebenso erfolgt üblicherweise die Analogausgabe auf das Peripherieausgangswort erst nach einer Normierung des Ausgabewertes.

In STEP7- Programmen wird zur Normierung auf die Rechenoperationen zurückgegriffen.
Damit dies möglichst exakt erfolgen kann müssen die Werte zum Normieren noch in den Datentyp REAL umgewandelt werden, damit die Rundungsfehler minimal sind.

Aufgabe:

Im folgenden Beispiel wird ein Wert von 0 bis 10V mit einer Analogbaugruppe SM334 auf Steckplatz 6 eingelesen (PEW288). Er liegt anfangs als GANZZAHL(16 Bit) vor und soll auf 100 bis 1000 im Gleitpunktformat normiert und in dem Merkerdoppelwort MD10 abgespeichert werden.

Lösung in AWL:

```
L     PEW 288      //Analogwert einlesen 0 bis 10 V entspricht 0 bis 27648 Ganzzahl (16 Bit)
ITD                //Wert von Ganzzahl (16 Bit) in Ganzzahl (32 Bit) umwandeln
DTR                // Wert von Ganzzahl (32 Bit) in Gleitpunktzahl umwandeln
L     2.7648e+4    //
/R                 //Division mit Gleitpunktzahl 27648
L     9.000e+2     //
*R                 // Multiplikation mit Gleitpunktzahl 900 (1000-100)
L     1.000e+2     //
+R                 // Addition mit Gleitpunktzahl 100 (Offset)
T     MD10         //normierter Wert 100 bis 1000 im Gleitpunktformat
```

Eine weitere Möglichkeit zum Normieren bietet der Baustein **FC105 SCALE CONVERT**.
Sie finden den Baustein FC105 in der Bibliothek **Standard Library** unter **TI-S7 Converting Blocks**.

Werte skalieren: FC105

Die Funktion Werte skalieren (SCALE) wandelt einen ganzzahligen Wert (IN) in einen Realzahlenwert um, der in physikalischen Einheiten zwischen einem unteren und einem oberen Grenzwert (LO_LIM und HI_LIM) skaliert wird. Das Ergebnis wird in den Parameter OUT geschrieben. Die Funktion SCALE arbeitet mit der folgenden Gleichung:

OUT = [((FLOAT (IN) – K1)/(K2–K1)) * (HI_LIM–LO_LIM)] + LO_LIM

Die Konstanten K1 und K2 werden unterschiedlich gesetzt, je nachdem, ob der Eingabewert BIPOLAR oder UNIPOLAR ist.

· BIPOLAR: Es wird angenommen, dass der ganzzahlige Eingabewert zwischen -27648 und 27648 liegt, deshalb sind K1 = -27648,0 und K2 = +27648,0.

· UNIPOLAR: Es wird angenommen, dass der ganzzahlige Eingabewert zwischen 0 und 27648 liegt, deshalb sind K1 = 0,0 und K2 = +27648,0.

Ist der ganzzahlige Eingabewert größer als K2, dann wird der Ausgang (OUT) an HI_LIM gebunden und ein Fehler ausgegeben. Ist der ganzzahlige Eingabewert kleiner als K1, dann wird der Ausgang an LO_LIM gebunden und ein Fehler ausgegeben.

Zum umgekehrten Skalieren wird LO_LIM > HI_LIM programmiert. Beim umgekehrten Skalieren verringert sich der Ausgabewert, während der Eingabewert zunimmt.

Parameter von FC105

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	Der Signalzustand "1" am Freigabeeingang aktiviert die Box.
ENO	Ausgang	BOOL	E, A, M, D, L	Der Freigabeausgang hat den Signalzustand "1", wenn die Funktion fehlerfrei ausgeführt wird.
IN	Eingang	INT	E, A, M, D, L, P, Konstante	Eingabewert, der in einen Wert vom Datentyp REAL in physikalischen Einheiten skaliert werden soll.
HI_LIM	Eingang	REAL	E, A, M, D, L, P, Konstante	Oberer Grenzwert in physikalischen Einheiten.
LO_LIM	Eingang	REAL	E, A, M, D, L, P, Konstante	Unterer Grenzwert in physikalischen Einheiten.
BIPOLAR	Eingang	BOOL	E, A, M, D, L	Bei dem Signalzustand "1" handelt es sich um einen bipolaren Eingabewert. Bei dem Signalzustand "0" handelt es sich um einen unipolaren Eingabewert.
OUT	Ausgang	REAL	E, A, M, D, L, P	Ergebnis der Skalierung.
RET_VAL	Ausgang	WORD	E, A, M, D, L, P	Gibt den Wert W#16#0000 aus, wenn die Operation fehlerfrei ausgeführt wird. Wird ein anderer Wert ausgegeben, entnehmen Sie der Fehlerinformation nähere Informationen hierzu.

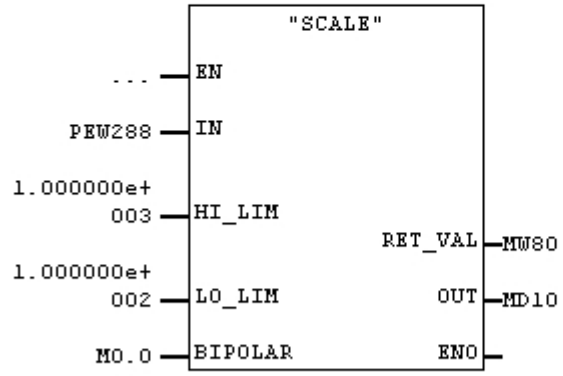
Aufgabe:

Im folgenden Beispiel wird ein Wert von 0 bis 10V mit einer Analogbaugruppe SM334 auf Steckplatz 6 eingelesen (PEW288). Er liegt anfangs als GANZZAHL(16 Bit) vor und soll auf 100 bis 1000 im Gleitpunktformat normiert und in dem Merkerdoppelwort MD10 abgespeichert werden.

Lösung mit Baustein FC105 SCALE:

```
Netzwerk 1: Wert 0 auf Merker M0.0 zuweisen
      CLR
      =     M      0.0
      NOP   0
```

```
Netzwerk 2: Baustein SCALE aufrufen
```




```
Symbolinformation:
 SCALE                    FC105              -- Scaling Values
```

Fehlerinformation Parameter RET_VAL (MW80)

Ist der ganzzahlige Eingabewert größer als 27648 (10V), dann wird der Ausgang (OUT) an HI_LIM gebunden (Wert 1000) und ein Fehler ausgegeben. Ist der ganzzahlige Eingabewert kleiner als 0 (0V), dann wird der Ausgang an LO_LIM (Wert 0) gebunden und ein Fehler ausgegeben.
Im Fehlerfall ändert sich der Signalzustand ENO von „1“ auf "0" und bei RET_VAL wird der Wert W#16#0008 ausgegeben.

5.2. ANALOGWERT NORMIEREN UND AUSGEBEN

Liegt ein normierter Wert vor und soll auf einer Analogausgabebaugruppe ausgegeben werden, so muss dieser zuvor passend normiert werden.

In STEP7- Programmen wird zur Normierung auf die Rechenoperationen zurückgegriffen.
Dies erfolgt noch in dem Datentyp REAL, damit die Rundungsfehler minimal sind. Erst dann wird dieser Wert auf einen Ganzzahlwert gerundet. Die Stellen hinter dem Komma gehen dabei verloren.

Aufgabe:

Im folgenden Beispiel liegt ein Wert von 100 bis 1000 im Gleitpunktformat abgespeichert auf Merkerdoppelwort MD20 vor und soll von 0 bis 10V normiert mit einer Analogbaugruppe SM334 ausgegeben (PAW288) werden.

Lösung in AWL:

```
L     MD20        // Wert 100 bis 1000 im Gleitpunktformat
L     1.000e+2    //
-R                // Subtraktion mit Gleitpunktzahl 100 (Offset)
L     9.000e+2    //
/R                // Division mit Gleitpunktzahl 900
L     2.7648e+4   //
*R                // Multiplikation mit Gleitpunktzahl 27648
RND               // Runden zur Ganzzahl
T     PAW 288     // 0 bis 27648 Ganzzahl (16 Bit) entspricht Analogwert ausgeben 0 bis 10 V
```

Eine weitere Möglichkeit zum Normieren bietet der Baustein **FC106 UNSCALE CONVERT**.
Sie finden den Baustein FC106 in der Bibliothek **Standard Library** unter **TI-S7 Converting Blocks**.

Werte deskalieren: FC106

Die Funktion Werte deskalieren (UNSCALE) wandelt einen Realzahlenwert (IN), der in physikalischen Einheiten zwischen einem unteren und einem oberen Grenzwert (LO_LIM und HI_LIM) skaliert ist, in einen ganzzahligen Wert um. Das Ergebnis wird in den Parameter OUT geschrieben. Die Funktion UNSCALE arbeitet mit der folgenden Gleichung:

OUT = [((IN–LO_LIM)/(HI_LIM–LO_LIM)) * (K2–K1)] + K1

Die Konstanten K1 und K2 werden unterschiedlich gesetzt, je nachdem, ob der Eingabewert BIPOLAR oder UNIPOLAR ist.

· BIPOLAR: Es wird angenommen, dass der ganzzahlige Ausgabewert zwischen -27648 und 27648 liegt, deshalb sind K1 = -27648,0 und K2 = +27648,0.

· UNIPOLAR: Es wird angenommen, dass der ganzzahlige Ausgabewert zwischen 0 und 27648 liegt, deshalb sind K1 = 0,0 und K2 = +27648,0.

Liegt der Eingabewert nicht in dem Bereich zwischen LO_LIM und HI_LIM, dann wird der Ausgang (OUT) an den nächsten Grenzwert (den oberen oder den unteren) für den angegebenen Bereich des jeweiligen Typs (BIPOLAR oder UNIPOLAR) gebunden und ein Fehler ausgegeben.

Parameter von FC106

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Eingang	BOOL	E, A, M, D, L	Der Signalzustand "1" am Freigabeeingang aktiviert die Box.
ENO	Ausgang	BOOL	E, A, M, D, L	Der Freigabeausgang hat den Signalzustand "1", wenn die Funktion fehlerfrei ausgeführt wird.
IN	Eingang	REAL	E, A, M, D, L, P, Konstante	Eingabewert, der in einen ganzzahligen Wert deskaliert werden soll.
HI_LIM	Eingang	REAL	E, A, M, D, L, P, Konstante	Oberer Grenzwert in physikalischen Einheiten.
LO_LIM	Eingang	REAL	E, A, M, D, L, P, Konstante	Unterer Grenzwert in physikalischen Einheiten.
BIPOLAR	Eingang	BOOL	E, A, M, D, L	Bei dem Signalzustand "1" handelt es sich um einen bipolaren Eingabewert. Bei dem Signalzustand "0" handelt es sich um einen unipolaren Eingabewert.
OUT	Ausgang	INT	E, A, M, D, L, P	Ergebnis der Deskalierung.
RET_VAL	Ausgang	WORD	E, A, M, D, L, P	Gibt den Wert W#16#0000 aus, wenn die Operation fehlerfrei ausgeführt wird. Wird ein anderer Wert ausgegeben, entnehmen Sie der Fehlerinformation nähere Informationen hierzu.

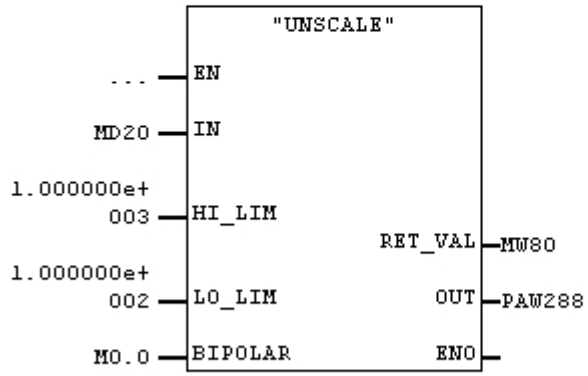
Aufgabe:

Im folgenden Beispiel liegt ein Wert von 100 bis 1000 im Gleitpunktformat abgespeichert auf Merkerdoppelwort MD20 vor und soll von 0 bis 10V normiert mit einer Analogbaugruppe SM334 ausgegeben (PAW288) werden.

Lösung Baustein FC106 UNSCALE:

```
Netzwerk 1: Wert 0 auf Merker M0.0 zuweisen
      CLR
      =     M      0.0
      NOP   0
```

```
Netzwerk 2: Baustein UNSCALE aufrufen
```




```
Symbolinformation:
 UNSCALE                FC106              -- Unscaling Values
```

Fehlerinformation Parameter RET_VAL (MW80)

Liegt der Eingabewert nicht in dem Bereich zwischen LO_LIM und HI_LIM, dann wird der Ausgang (OUT) an den nächsten Grenzwert (den oberen oder den unteren) für den angegebenen Bereich des jeweiligen Typs (BIPOLAR oder UNIPOLAR) gebunden und ein Fehler ausgegeben.
Der Signalzustand von ENO wird auf "0" und RET_VAL wird auf W#16#0008 gesetzt.