

Variable

Die Deklarationstabelle

Struktur des Programms

Wir wollen in diesem Programm nur zwei Bausteine verwenden und zwar einen Funktionsbaustein FB1 und eine Funktion FC1. Der FB1 braucht noch einen Datenbaustein DB1. Natürlich muss der Organisationsbaustein OB1 erstellt werden.

Bei der Programmierung in der **inkrementellen*** Eingabe wird die Variablendeklaration der Codebausteine in Form einer Tabelle dargestellt. Die Variablen sind nur in diesem Baustein gültig, darum nennt man sie „(baustein-) **lokale Variablen**“, d.h. der gleiche Variablenname kann in einem anderen Baustein in anderer Bedeutung verwendet werden.

Variablendeklarationstabelle

Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar
	in				
	out				
	in_out				
	stat	Nur im Funktionsbaustein vorhanden			
	temp				

FC:

The screenshot shows the 'FC' (Function Call) configuration window. On the left, a tree view shows the 'Schnittstelle' (Interface) with variables: IN, OUT, IN_OUT, TEMP, and RETURN. On the right, a table titled 'Inhalt von: 'Umgebung\Schnittstelle'' lists these variables under the 'Name' column.

Name
IN
OUT
IN_OUT
TEMP
RETURN

FB:

The screenshot shows the 'FB' (Function Block) configuration window. On the left, a tree view shows the 'Schnittstelle' (Interface) with variables: IN, OUT, IN_OUT, STAT, and TEMP. On the right, a table titled 'Inhalt von: 'Umgebung\Schnittstelle'' lists these variables under the 'Name' column.

Name
IN
OUT
IN_OUT
STAT
TEMP

Variable

Variablen- Beschreibung

Variable	Beschreibung
in	Eingangsparameter, dessen Wert vom Baustein, der den Codebaustein aufruft, bereitgestellt wird
out	Parameter, dessen Wert vom Baustein, der den Codebaustein aufruft, ausgegeben wird
in-out	Ausgangsparameter, dessen Wert vom aufrufenden Baustein bereitgestellt, vom aufgerufenen Baustein verändert und an den aufrufenden Baustein als Codebaustein zurückgegeben wird
stat	Statische Variable, die in einem Instanz-DB gespeichert wird
temp	Temporäre Variable, die in einem L-Stack gespeichert wird. Der Wert wird nach der Bearbeitung des Bausteins nicht gespeichert

In STEP7 ist die Deklaration von der Bausteinart abhängig

Deklarationstyp	Syntax in Tabelle	OB	FB	FC
Eingang	in	Unzul.	Ja	Ja
Ausgang	out	Unzul.	Ja	Ja
Durchgang	in_out	Unzul.	Ja	Ja
Statisch	stat	Unzul.	Ja	Unzul.
Temporär	temp	Ja	Ja	Ja

Unzul. = Unzulässig

Variablen, die in einem Programm verwendet werden, muss ein Datentyp zugeordnet werden. Es sind folgende elementare Datentypen zugelassen:

Datentyp	Bitgröße	Beschreibung
BOOL	1	Bit-Werte wahr oder falsch
BYTE	8	Byte-Wert 0 bis 255
WORD	16	Wort- Wert 0 bis 65535
DWORD	32	Doppelwort-Werte 0 bis $(2^{32} - 1)$
CHAR	8	ASCII-Code
INT	16	Ganzzahl (16 Bit) von -32768 bis +32767
DINT	32	Ganzzahl (32 Bit) von -2^{31} bis $(2^{31} - 1)$
REAL	32	IEEE-Gleitpunktzahl
TIME	32	IEC-Zeit in Intervallen von 1 ms
DATE	32	IEC-Datum in Intervallen von 1 Tag
TIME_OF_DAY	32	Zeit in 1ms-Raster: Std. (0-23), Min. (0-59),
TOD		Sekunde (0-59), ms (0-999)
S5TIME	32	Voreingestellte Zeitwert für Zeiten Bereich:0H_0M_0S_0MS bis 2H_46M_30S_0MS

Variable

Neben den elementaren Datentypen sind zusammengesetzte Datentypen erlaubt:

Datentyp	Beschreibung
DATE_AND_TIME DT	Der Bereich wird mit 8 Bytes definiert. In binärcodiertem Dezimalformat wird Datum und Zeit dargestellt: Jahr in Byte 0, Monat in Byte 1, Tag in Byte 2, Stunde in Byte 3, Minuten in Byte 4, Sekunden in Byte 5, Milisekunden in Byte 6 und der Hälfte von Byte 7, Wochentag in der anderen Hälfte von Byte 7
STRING	Definiert eine Sequenz von bis zu 254 Zeichen (Datentyp CHAR)
ARRAY	Dient der Gruppierung eines Datentyps; z.B. ARRAY[1..3,1..2] OF INT bedeutet ein Feld im Forma 3x2 aus Ganzzahlen
STRUCT	Dient der Gruppierung von beliebig kombinierten Datentypen

Es gibt eine definierte Schreibweise für die Werte der einzelnen Datentypen.

Schreibweise von Zeitwerten

Schreibweise	Datentyp	Beispiel
T# , Time#	TIME	T#1D_2H_10M23S0MS
D# , Date#	DATE	D#1997-3-20
TOD# , Time-of_day#	TIME_OF-DAY	TOD#13:23:32.444
S5T# , S5Time#	S5TIME	S5T#12M_22S_100MS
DT# , Date_and_time#	DATE_AND_TIME	DT#1997-3-20-17:11:3.433

Schreibweise von numerischen Werten

Schreibweise	Datentyp	Beispiel
2#	WORD,	2#0001_0110_1100_0011 -Binär 16 Bit
	DWORD	2#0001_0110_1100_0011_1111_0101_0011_1111 - Binär 32 Bits
True/False	BOOL	TRUE bzw. FALSE
B#(...), Byte#(...)	WORD,	B#(10,20) - 16 Bits
	DWORD	B#(1,15,100,112) - 32 Bits
B#16#, Byte#16#	BYTE	B#16#2A : 8 Bits Hexadezimal
W#16#, Word#16#	WORD	W#16#AA11 :16 Bits Hexadezimal
DW#16#, Dword#16#	DWORD	DW#16#08A1_AA11 :32 Bits Hexadezimal
Ganzzahl	INT	615
L#	DINT	L#44520
Realzahl	REAL	1.23e+13
c#	WORD	c#500

Variable

Der Organisationsbaustein OB1

Das Betriebssystem der S7 CPU bearbeitet den OB1 zyklisch. Ist die Bearbeitung von OB1 beendet, beginnt das Betriebssystem erneut, den OB1 zu bearbeiten. Die zyklische Bearbeitung des OB1 wird nach Beendigung des Anlaufs begonnen. Sie können in OB1 Funktionsbausteine (FBs) oder Funktionen (FCs, SFCs) aufrufen.

Beim Anlegen eines neuen Organisationsbausteins wird die Deklarationstabelle mit den Lokaldaten vorbelegt, die für den jeweiligen Organisationsbaustein vorgesehen sind. Diese 20 Bytes beinhalten eine vom Betriebssystem zur Verfügung gestellte, systemeinheitliche Startinformation. Diese Information beinhaltet sowohl Einstellungen für das Verhalten der Organisationsbausteine, als auch Informationen wie z.B. Prioritätsklasse, Nummer des OBs, Kennungen für verursachende Ereignisse usw. Diese Informationen werden zur Laufzeit des OBs eingetragen und können z.B. zu Diagnosezwecken ausgelesen werden.

Deklarationstabelle des OB1

Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar
0.0	temp	OB1_EV_CLASS	BYTE		Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
1.0	temp	OB1_SCAN_1	BYTE		1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
2.0	temp	OB1_PRIORITY	BYTE		1 (Priority of 1 is lowest)
3.0	temp	OB1_OB_NUMBR	BYTE		1 (Organization block 1, OB1)
4.0	temp	OB1_RESERVED_1	BYTE		Reserved for system
5.0	temp	OB1_RESERVED_2	BYTE		Reserved for system
6.0	temp	OB1_PREV_CYCL E	INT		Cycle time of previous OB1 scan (milliseconds)
8.0	temp	OB1_NIM_CYCLE	INT		Minimum cycle time of OB1 (milliseconds)
10.0	temp	OB1_MAX_CYCLE	INT		Maximum cycle time of OB1 (milliseconds)
12.0	temp	OB1_DATE_TIME	DATE_AND_TIME		Date and time OB1 started

Die Variablen-tabelle (VAT)

Variablen beobachten oder steuern

Das Programm ist jetzt fertig. Leider hat der Fehler-teufel zugeschlagen. Um den Fehler besser einzugrenzen, sollen ausgewählte Variablen mit definiertem Trigger beobachtet werden. Bevor Sie Variablen beobachten oder steuern können, müssen Sie eine Variablen-tabelle (VAT) erstellen und die gewünschten Variablen eintragen. Sie können diese Variablen-tabelle abspeichern, so daß sie beim erneuten Test Ihres Programms wieder zum Beobachten und Steuern verwendet werden kann.

Gehen Sie folgendermaßen vor:

1. Erzeugen Sie eine neue Variablen-tabelle, indem Sie den Menübefehl Menübefehl **Einfügen > S7-Baustein > Variablen-tabelle** wählen.
2. Bestimmen Sie die Variablen, deren Werte Sie vorgeben oder beobachten wollen und tragen Sie diese in die Variablen-tabelle ein.
Gehen Sie dabei von "außen" nach "innen" vor, d.h. wählen Sie zuerst die Eingänge und danach die Variablen, die von den Eingängen beeinflusst werden bzw. die die Ausgänge beeinflussen und zum Schluss die Ausgänge.
Wenn Sie z.B. das Eingangsbit 4.2, das Eingangsbit 4.3 und das Ausgangsbit 8.1 beobachten möchten, geben Sie in der Operandenspalte folgendes ein:

SIMATIC S7_Kurs\S7-Programm(1)\AP-off\VAT1

Operand	Symbol	Statusformat	Statuswert	Steuerwert
E 4.2	Vollmeldung 2	BOOL	false	
E 4.3	Leermeldung 2	BOOL	true	
A 8.1	Einlass-Ventil 2	BOOL	false	

3. Speichern Sie die Variablen-tabelle mit dem Menübefehl **Tabelle > Speichern**. Haben Sie mit der unbenannten Variablen-tabelle gearbeitet, müssen Sie nun eine Nummer für die Variablen-tabelle angeben, z.B. VAT1.
4. Stellen Sie über den Menübefehl **Zielsystem > Verbindung herstellen zu** eine Verbindung zur gewünschten CPU her, damit Sie die Variablen der aktiven Variablen-tabelle beobachten können.
5. Definieren Sie für das Beobachten der Variablen Triggerpunkt und Triggerbedingung über den Menübefehl **Variable > Trigger**.
6. Starten Sie das Beobachten mit dem Menübefehl **Variable > Beobachten**. Mit einem Haken wird gekennzeichnet, dass das Beobachten gestartet worden ist.
7. Beenden können Sie das Beobachten durch wiederholtes Anwählen des Menübefehls **Variable > Beobachten**. Wollen Sie einen neuen Trigger definieren, beginnen Sie bei Schritt 5.