akytec

CODESYS V3.5 MERKBLATT

KOMMENTAR

// einzeiliger Kommentar	(* mehrzeiliger Kommentar *)

VARIABLENDEKLARATION

<Name> : <Typ> := <Anfangswert>;

Benennungsregeln für Variablen

Wird kein Anfangswert angegeben, wird "Null" verwendet

Leerzeichen vor und nach ":", ":=" und ";" werden nicht gewertet

Beispiele:

```
iVar: INT := 123;
rVar: REAL := 11.22;
wVar: WORD := 16#FFAB;
// Die maximale Länge der Zeichenfolge beträgt 20 Zeichen
sVar: STRING (20) := 'hello, world';
```

BASISTYPEN VON VARIABLENDEKLARATIONSBLÖCKEN

Blocktyp	Variablen	Deklarationsstelle
VAR ... END_VAR	Lokale Variablen	In Programmen, Funktionsblöcken, Funktionen, Methoden
VAR_INPUT ... END_VAR	Eingangsvariablen	
VAR_OUTPUT ... END_VAR	Ausgangsvariablen	
VAR_IN_OUT ... END_VAR	Eingangs-/Ausgangsvariablen	
VAR CONSTANT ... END_VAR	Konstanten	
VAR_GLOBAL ... END_VAR	Globale Variablen	In der Liste der globalen Variablen
VAR RETAIN ... END_VAR	RETAIN-Variablen (nicht flüchtig)	In Programmen und Listen globaler Variablen
VAR_GLOBAL RETAIN ... END_VAR	Globale RETAIN-Variablen (nicht flüchtig)	In der Liste der globalen Variablen
VAR_GLOBAL PERSISTENT RETAIN ... END_VAR	Globale persistente Variablen (nicht flüchtig)	In der Liste der persistenten Variablen

DATENTYPEN

Boolesche und Ganzzahlen

Typ	Größe in Bytes	Gültiger Bereich
BOOL	1	TRUE / FALSE
BYTE/USINT	1	0...255
WORD/UINT	2	0...65535
DWORD/UDINT	4	$0...2^{32}-1$
LWORD/ULINT	8	$0...2^{64}-1$
SINT	1	-128...127
INT	2	-32768...32767
DINT	4	$-2^{31}...2^{31}-1$
LINT	8	$-2^{63}...2^{63}-1$

DATENTYPEN	Boolesche und Ganzzahlen

Hinweise:	1. Bitweiser Zugriff ist unterstützt:
	// TRUE in das 15. Bit der Variablen vom Typ WORD schreiben
	wVar.15 := TRUE;

DATENTYPEN	Gleitkommazahlen	
Typ	**Größe in Bytes**	**Genauigkeit** (Anzahl der signifikanten Stellen nach dem Dezimalkomma)
REAL	4	7-8
LREAL	8	15-17
Beispiele:	11.22 1.25E3 (exponentielle Notation)	

DATENTYPEN	Zeichenketten		
Typ	**Zeichengröße in Bytes**	**Kodierung**	**Beispiele für Literale** (beachten Sie die Anführungszeichen)
STRING	1	ASCII	sVar: STRING (20) := 'hello, world';
WSTRING	2	UTF-16	wsVar: WSTRING (20) := "hello, world"

Hinweise:

1. Zeichenketten in CODESYS sind null-terminiert. Wenn sie dekldeklariert werden, wird automatisch Speicher für das String-Endezeichen reserviert, d.h. STRING(20) benötigt 21 Byte Speicher, und WSTRING(20) benötigt 42 Byte Speicher.

2. Bei der Deklaration einer Zeichenkette kann deren maximale Länge angegeben werden. Hierfür gibt es keine expliziten tEinschränkungen, aber beachten Sie die Tabelle STANDARD LIBRARY und die nachstehenden Hinweise.

3. Um kyrillische Zeichen in der Visualisierung anzuzeigen, sollte nur der Typ WSTRING verwendet werden.

DATENTYPEN		Datum und Zeit		
Typ	**Größe in Bytes**	**Gültiger Bereich**	**Beschreibung**	**Einheit**
TIME	4	T#0ms...T#49d17h2m47s295ms	Zeitintervall	Millisekunden
DT	4	DT#1970-1-1-0:0:0... DT#2106-02-07-06:28:15	Datum und Zeit	Sekunden
DATE	4	DATE#1970-1-1-... DT#2106-02-07	Datum	Sekunden (die Tageszeit wird nicht verarbeitet)
TOD	4	TOD#00:00:00... TOD#23:59:59.999	Tageszeit	Milliseconds

Hinweise:

1. Gültige arithmetische Operationen

TIME + TIME = TIME	DT – TIME = DT
TIME – TIME = TIME	DT – DT = TIME
TIME x Ganzzahl = TIME	TOD + TIME = TOD
DT + TIME = DT	TOD – TIME = TOD

2. Die Typen LTIME/LDT/LDATE/LTOD unterscheiden sich in Größe (8 Byte), Bereich und Einheiten (Nanosekunden).

TYPKONVERTIERUNG

```
TO_<Typ>
// Konvertierung einer Gleitkommazahl in eine Zeichenkette
sVar := TO_STRING(rVar);
// Konvertierung einer Integer-Variablen in ein Zeitintervall (in Millisekunden)
tVar := TO_TIME(udiVar);
// andere Typen werden auf die gleiche Weise konvertiert
```

ZUSAMMENGESETZTE DATENTYPEN (siehe CODESYS-Hilfe)

ARRAY	STRUCT	UNION	ENUMERATION

KURZ ÜBER ARRAYS (als der am häufigsten verwendete zusammengesetzte Datentyp)

```
// Deklaration eines eindimensionalen Arrays aus vier Elementen mit den Anfangswerten 10, 20, 30 und
// 40 entsprechend
VAR
aiData := ARRAY [0..3] OF INT := [10, 20, 30, 40];
END_VAR
// Adressierung eines Array-Elements im Programmcode
// iVar wird der Wert 10 zugewiesen
iVar := aiData[0];
// Schreiben des Wertes 25 in das Array-Element mit dem Index 1
aiData[1] := 25;
```

BASISOPERATOREN

Mnemonik in ST-Sprache	Mnemonik in grafischen Sprachen	Beschreibung
Arithmetische Operatoren		
:=	MOVE	Zuweisung von Werten einer Variablen (oder eines Literals) an eine andere. In ST erfolgt die Zuordnung "von rechts nach links", in grafischen Sprachen "von links nach rechts".
+	ADD	Addition
-	SUB	Subtraktion
*	MUL	Multiplikation
/	DIV	Ganzzahlige Division
MOD		Nichtnegativer ganzzahliger Rest der Division
Bitweise Operatoren (anwendbar auf BOOL und alle Integer-Typen)		
NOT		Bitweises "NOT"
AND		Bitweises "AND"
OR		Bitweises "OR"
XOR		Bitweises "Exclusives OR"
Relationale Operatoren		
=	EQ	Vergleich auf Gleichheit
<>	NE	Vergleich auf Ungleichheit
>	GT	Vergleich auf "größer als"
<	LT	Vergleich auf "kleiner als"
>=	GE	Vergleich auf "größer oder gleich"
<=	LE	Vergleich auf "kleiner oder gleich"
Bitshift-Operatoren		
SHL		Verschiebung nach links, Auffüllen mit Nullen
SHR		Verschiebung nach rechts, Auffüllen mit Nullen
ROL		Zyklische Linksverschiebung (Linksrotation)
ROR		Zyklische Rechtsverschiebung (Rechtsrotation)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Auswahl-Operatoren	
SEL	Auswahl aus zwei Werten desselben Typs
MUX	Auswahl aus einer beliebigen Anzahl von Werten desselben Typs (Multiplexer)
LIMIT	Bereichseinschränkung
Mathematische Operatoren	
MIN	Auswahl eines Minimums aus einer beliebigen Anzahl von Werten desselben Typs
MAX	Auswahl eines Maximums aus einer beliebigen Anzahl von Werten desselben Typs
ABS	Absolutwert
SQRT	Quadratwurzel-Extraktion
LN	Natürlicher Logarithmus
LOG	Dezimaler Logarithmus
EXP	Exponentialfunktion
EXPT	Potenzfunktion
TRUNC	Abschneiden eines REAL-Wertes auf DINT mit Verwerfen des Bruchteils.
TRUNC INT	Abschneiden eines REAL-Wertes auf INT mit Verwerfen des Bruchteils.
Trigonometrische Operatoren	
SIN	Sinus
COS	Kosinus
TAN	Tangens
ASIN	Arkussinus
ACOS	Arkuskosinus
ATAN	Arkustangens
TIME	Seit dem Systemstart verstrichene Zeit in Millisekunden
SIXEOF, XSIZEOF	Objektgröße in Bytes
Hinweise:	1. Die Reihenfolge, in der die Operatoren in einem Ausdruck ausgeführt werden, wird durch ihre Priorität und die Anordnung der Klammern bestimmt.

STEUERANWEISUNGEN DER ST-SPRACHE

Anweisung	Beschreibung
IF...THEN...ELSIF...THEN...ELSE...END_IF	Bedingte Auswahl (Prüfung einer Reihe von Bedingungen und Ausführung der nachfolgenden Operationen für die erste, die sich als wahr erweist)
CASE...OF...:...ELSE...END_CASE	Mehrfachauswahl-Operator (Prüfung einer Integer-Variablen auf Übereinstimmung mit einer Reihe von Werten und Ausführung nachfolgender Operationen im Falle einer Übereinstimmung)
FOR...TO...BY...DO...END_FOR	Zyklus mit bestimmter Anzahl von Iterationen
WHILE...DO...END_WHILE	Zyklus mit Vorbedingung (Bedingungsprüfung vor Iteration)
REPEAT...UNTIL...END_REPEAT	Zyklus mit Nachbedingung (Bedingungsprüfung nach Iteration)
CONTINUE	Weiter zur nächsten Iteration der Schleife
EXIT	Zyklus verlassen
RETURN	POU verlassen

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

GRUNDTYPEN VON PROGRAMMOBJEKTEN (POU)

Name	Eigener Speicher und Speichern von Variablenwerten zwischen Aufrufen	Deklaration von Instanzen
Funktion (FUN)	Nein	Nein
Funktionsblock (FB)	Ja	Ja
Programm (PRG)	Ja	Nein

STANDARD-BIBLIOTHEK (Grundfunktionen und -Funktionsblöcke)

Name	Beschreibung
Zeichenketten-Funktionen	
CONCAT	Verbindet zwei Zeichenketten zu einer einzigen Zeichenkette
DELETE	Löscht ein Segment aus einer Zeichenkette
FIND	Suche nach einem Segment in einer Zeichenkette
INSERT	Fügt ein Segment in eine Zeichenkette ein
LEFT	Gibt eine bestimmte Anzahl von Zeichen einer Zeichenkette zurück, beginnend von links
LEN	Gibt die Länge einer Zeichenkette zurück
MID	Gibt eine bestimmte Anzahl von Zeichen einer Zeichenkette zurück, beginnend an einer bestimmten Position
REPLACE	Ersetzt eine bestimmte Anzahl von Zeichen einer Zeichenkette durch eine andere Zeichenkette
RIGHT	Gibt eine bestimmte Anzahl von Zeichen einer Zeichenkette zurück, beginnend von rechts

Hinweise:

1. Funktionen zur Bearbeitung von Zeichenketten des Typs WSTRING und Timern des Typs LTIME sind in der Standard64-Bibliothek verfügbar.
2. Andere häufig verwendete Blöcke (lineare Skalierung, Impulsgeber, BLINK, PID-Regler usw.) sind in der Util-Bibliothek verfügbar.
3. Zeichenketten-Funktionen können nur Zeichenketten mit einer Länge von bis zu 255 Zeichen verarbeiten. Um mit längeren Zeichenketten zu arbeiten, verwenden Sie die Funktionen der StringUtils-Bibliothek.
4. Erweiterte Funktionen für die Verarbeitung von Strings (Parsing, Formatierung usw.) sind in der Bibliothek AkytecStringUtils verfügbar.

Flanken-Detektoren	
R_TRIG	Erzeugt einen einzelnen Impuls bei der steigenden Flanke (R_TRIG) oder der fallenden Flanke (F_TRIG) der angegebenen booleschen Variablen
F_TRIG	
Triggers	
SR	Set-dominanter Trigger
RS	Reset-dominanter Trigger
Zähler	
CTU	Aufwärts-Impulszähler
CTD	Abwärts-Impulszähler
CTUD	Impulszähler mit Unterstützung beider Zählrichtungen
Zeitschaltuhren	
TP	Generator eines logischen Signals bestimmter Länge
TON	Timer mit verzögerter Einschaltung
TOF	Timer mit verzögerter Ausschaltung

ANDERE NÜTZLICHE BIBLIOTHEKEN AUS DEM CODESYS-PAKET

Bibliothek	Beschreibung	Dokumentation
CAA Memory	Speicheroperationen (Kopieren, Vergleichen, Füllen usw. von Speicherblöcken)	öffnen
CAA File	Datei-Operationen (Lesen/Schreiben)	öffnen
CAA SerialCom	Arbeiten mit COM-Port zur Implementierung nicht standardisierter Austauschprotokolle	öffnen
CAA NetBaseServices	Arbeiten mit TCP/UDP zur Implementierung nicht standardisierter Austauschprotokolle	öffnen
Visu Utils	Visualisierungs-Steuerung vom Programmcode aus (Umschalten zwischen Bildschirmen, Öffnen von Fenstern, etc.)	öffnen

CODESYS ONLINE-HILFE

CODESYS Online-Hilfe