



Dateiname: Example\_CTT3\_2.pro  
 Verzeichnis: D:\Andreas\_Schiff\Programme\CoDeSys\_Programme\Projects  
 Geändert am: 6.3.06 12:47:58 / V2.2  
 Bezeichnung: Example CTT3  
 Autor: A. Schiff, ICS GmbH  
 Version: 1.0 vom 12.10.2005  
 Beschreibung: Das Programm wickelt die Kommunikation mit einem Slave nach dem Profil S-7.A.A entsprechend der AS-Interface Spezifikation Version 3.0 ab. Es gibt ein Byte Information an den Slave aus und liest ein Byte Information vom Slave ein.

Beispielprogramm



```

0001 FUNCTION_BLOCK CTT3
0002 (* ***** *)
0003 (* für Firmware 2.0          Datum: 28.02.2006 *)
0004 (* ***** *)
0005 VAR_INPUT
0006     DI0:BOOL;
0007     DI1:BOOL;
0008     DI2:BOOL;
0009     DI3:BOOL;
0010     OutByte:BYTE;          (* Byte mit Ausgabeinformation für den Slave *)
0011 END_VAR
0012 VAR_OUTPUT
0013     DO0:BOOL;
0014     DO1:BOOL;
0015     DO2:BOOL;
0016     InByte:BYTE;          (* Byte mit Eingabeinformation vom Slave *)
0017     OK:BOOL;              (* =TRUE: Datenverkehr in Ordnung, Eingabebyte gültig
0018                             =FALSE: Datenverkehr gestört, Eingabebyte nicht gültig *)
0019 END_VAR
0020 VAR
0021     OutB: BYTE;
0022     InB:BYTE;
0023     Y: BYTE;
0024     Z1: BYTE;
0025     Z2: BYTE;
0026     Z3:BYTE;
0027     Z4:BYTE;
0028 END_VAR
0001 IF DI3=FALSE AND DI2=FALSE THEN
0002 (* Beginn der Sequenz, Ausgabebyte wird eingefroren, DO6 und DO7 werden übertragen *)
0003     OutB:=OutByte;
0004     InB:=InB AND 2#00111111;
0005     IF DI1=TRUE THEN
0006         InB:=InB OR 2#10000000;
0007     END_IF;
0008     IF DI0=TRUE THEN
0009         InB:=InB OR 2#01000000;
0010     END_IF;
0011     Y:=OutB AND 2#10000000;
0012     IF Y<>0 THEN
0013         DO1:=TRUE;
0014     ELSE
0015         DO1:=FALSE;
0016     END_IF;
0017     Y:=OutB AND 2#01000000;
0018     IF Y<>0 THEN
0019         DO0:=TRUE;
0020     ELSE
0021         DO0:=FALSE;
0022     END_IF;
0023     Z1:=Z1+1;
0024     Z2:=0;
0025     IF Z1>100 AND Z1<200 THEN
0026         OK:=FALSE;
0027         DO2:=NOT DI2;
0028         DO1:=FALSE;
0029         DO0:=FALSE;
0030         InByte:=0;
0031     ELSIF Z1>199 THEN
0032         Z1:=100;

```



```

0033     OK:=FALSE;
0034     DO2:=DI2;
0035     DO1:=FALSE;
0036     DO0:=FALSE;
0037     InByte:=0;
0038     ELSE
0039         DO2:=DI2;
0040     END_IF;
0041 END_IF;
0042 IF DI3=FALSE AND DI2=TRUE THEN
0043 (* Zweiter Schritt der Sequenz, DO4 und DO5 werden übertragen *)
0044     InB:=InB AND 2#11001111;
0045     IF DI1=TRUE THEN
0046         InB:=InB OR 2#00100000;
0047     END_IF;
0048     IF DI0=TRUE THEN
0049         InB:=InB OR 2#00010000;
0050     END_IF;
0051     Y:=OutB AND 2#00100000;
0052     IF Y<>0 THEN
0053         DO1:=TRUE;
0054     ELSE
0055         DO1:=FALSE;
0056     END_IF;
0057     Y:=OutB AND 2#00010000;
0058     IF Y<>0 THEN
0059         DO0:=TRUE;
0060     ELSE
0061         DO0:=FALSE;
0062     END_IF;
0063     Z2:=Z2+1;
0064     Z3:=0;
0065     IF Z2>200 THEN
0066         Z2:=200;
0067         OK:=FALSE;
0068         IF DI2=DO2 THEN
0069             DO2:= NOT DI2;
0070         ELSE
0071             DO2:=DI2;
0072         END_IF;
0073         DO1:=FALSE;
0074         DO0:=FALSE;
0075         InByte:=0;
0076     ELSE
0077         DO2:=DI2;
0078     END_IF;
0079 END_IF;
0080 IF DI3=TRUE AND DI2=FALSE THEN
0081 (* Dritter Schritt der Sequenz, DO2 und DO3 werden übertragen *)
0082     InB:=InB AND 2#11110011;
0083     IF DI1=TRUE THEN
0084         InB:=InB OR 2#00001000;
0085     END_IF;
0086     IF DI0=TRUE THEN
0087         InB:=InB OR 2#00000100;
0088     END_IF;
0089     Y:=OutB AND 2#00001000;
0090     IF Y<>0 THEN
0091         DO1:=TRUE;
0092     ELSE

```

```

0093     DO1:=FALSE;
0094     END_IF;
0095     Y:=OutB AND 2#00000100;
0096     IF Y<>0 THEN
0097         DO0:=TRUE;
0098     ELSE
0099         DO0:=FALSE;
0100     END_IF;
0101     Z3:=Z3+1;
0102     Z4:=0;
0103     IF Z3>200 THEN
0104         Z3:=200;
0105         OK:=FALSE;
0106         IF DI2=DO2 THEN
0107             DO2:= NOT DI2;
0108         ELSE
0109             DO2:=DI2;
0110         END_IF;
0111         DO1:=FALSE;
0112         DO0:=FALSE;
0113         InByte:=0;
0114     ELSE
0115         DO2:=DI2;
0116     END_IF;
0117 END_IF;
0118 IF DI3=TRUE AND DI2=TRUE THEN
0119 (* Vierter Schritt der Sequenz, DO0 und DO1 werden übertragen *)
0120     InB:=InB AND 2#11111100;
0121     IF DI1=TRUE THEN
0122         InB:=InB OR 2#00000010;
0123     END_IF;
0124     IF DI0=TRUE THEN
0125         InB:=InB OR 2#00000001;
0126     END_IF;
0127     Y:=OutB AND 2#00000010;
0128     IF Y<>0 THEN
0129         DO1:=TRUE;
0130     ELSE
0131         DO1:=FALSE;
0132     END_IF;
0133     Y:=OutB AND 2#00000001;
0134     IF Y<>0 THEN
0135         DO0:=TRUE;
0136     ELSE
0137         DO0:=FALSE;
0138     END_IF;
0139     Z4:=Z4+1;
0140     Z1:=0;
0141     IF Z4>200 THEN
0142         Z4:=200;
0143         OK:=FALSE;
0144         IF DI2=DO2 THEN
0145             DO2:= NOT DI2;
0146         ELSE
0147             DO2:=DI2;
0148         END_IF;
0149         DO1:=FALSE;
0150         DO0:=FALSE;
0151         InByte:=0;
0152     ELSE

```

CTT3 (FB-ST)



```

0153     DO2:=DI2;
0154     END_IF;
0155     IF Z1<10 AND Z2<10 AND Z3<10 AND Z4<10 THEN
0156         OK:=TRUE;
0157         InByte:=InB;
0158     ELSE
0159         InByte:=0;
0160     END_IF;
0161 END_IF;
    
```

PLC\_PRG (PRG-ST)

```

0001 PROGRAM PLC_PRG
0002 (*-----*)
0003 (* Testprogramm für CTT3-Datenübertragung mit Slave-Profil S-7.A.A (8E/8A-Modul) *)
0004 (* Slave muss mit der Adresse 1a im Netzwerk vorhanden sein. Der Master ist Client und reagiert auf *)
0005 (* Slave-Telegramme entsprechend der Combined Transaction Type 3 aus Spezifikation Version 3.0 *)
0006 (* Die Variable OKAY bezieht sich ausschließlich auf die CTT3-Kommunikation und bezieht den Peri- *)
0007 (* pheriefehler nicht mit ein! *)
0008 (* Dieses Beispielprogramm dient nur zur Demonstration der Funktion des PS020. Es kann keine Gewähr dafür *)
0009 (* übernommen werden, dass dieses Programm in einem realen Automatisierungsprojekt fehlerfrei läuft. *)
0010 (*Programmversion 2.0 Autor: Andreas Schiff Datum: 28.02.2006 *)
0011 (*-----*)
    
```

```

0012
0013 VAR
0014     DO0 AT %QX1.1.0:BOOL; (* Ausgang DO0 *)
0015     DO1 AT %QX1.1.1:BOOL; (* Ausgang DO1 *)
0016     DO2 AT %QX1.1.2:BOOL; (* Ausgang DO2 *)
0017     DI0 AT %IX1.1.0:BOOL; (* Eingang DI0 *)
0018     DI1 AT %IX1.1.1:BOOL; (* Eingang DI1 *)
0019     DI2 AT %IX1.1.2:BOOL; (* Eingang DI2 *)
0020     DI3 AT %IX1.1.3:BOOL; (* Eingang DI3 *)
0021     Input:BYTE; (* 8 Bit Eingangsdaten vom Slave (nur gültig, wenn OKAY=TRUE!) *)
0022     Output:BYTE; (* 8 Bit Ausgangsdaten zum Slave *)
0023     Okay:BOOL; (* =TRUE, wenn Daten korrekt übertragen worden sind
0024                =FALSE, wenn Fehler in Übertragung festgestellt wurde *)
0025     CTT3:CTT3; (* Funktionsbaustein für Kommunikation mit PS020 *)
0026     Taktgeber3:TON;
0027     Taktgeber4:TON;
0028     HB2:BOOL;
0029     Takt2:BOOL;
0030     Takt2alt: BOOL;
0031 END_VAR
    
```

```

0032
0033 (* Ende des Deklarationsteils *)
0001 (* Hauptprogramm *)
0002
0003 CTT3(DI0:=DI0, DI1:=DI1, DI2:=DI2, DI3:=DI3, OutByte:=Output);
0004 DO0:=CTT3.DO0;
0005 DO1:=CTT3.DO1;
0006 DO2:=CTT3.DO2;
0007 Input:=CTT3.InByte;
0008 Okay:=CTT3.OK;
0009
    
```

Projektinformationen  
CTT3 (FB-ST)  
PLC\_PRG (PRG-ST)

A  
1  
4

Beispielprogramm