

## ASi-Beispielprogramm

### *Inhaltsverzeichnis*

Programmfunktion .....	2
Eingesetzte Hardware.....	2
Eingesetzte Software .....	2
Mitgeltende Unterlagen .....	3
Hardwarekonfiguration.....	3
SIMATIC Step7 Bausteine .....	4
Auslesen der binären Ein- und Ausgangsdaten.....	5
Auslesen der Fehlermeldungen.....	6
Auslesen der ASi-Slave-Diagnose .....	6
Soll-Ist-Vergleich .....	10

## **Programmfunktion**

In diesem Beispielprogramm werden folgende Funktionen realisiert:

1. Auslesen der binären Ein und Ausgangsdaten der ASi-Slaves bei Vollausbau des ASi-Stranges mit 62 Teilnehmern
2. Auslesen der Fehlermeldungen
  - Erdschlusswächter
  - Doppeladressierung
  - Konfigurationsfehler
  - Peripheriefehler
3. Auslesen der ASi-Slave-Diagnose
  - LPS: Liste der projektierten Slaves
  - LDS: Liste der detektierten Slaves
  - LCS: Liste der Slaves, die seit dem letzten Auslesen der Liste einen Konfigurationsfehler ausgelöst haben
  - Deltaliste: Slaves, die im Augenblick einen Konfigurationsfehler verursachen
  - Fehlerzähler

Abbildung folgender Zustände durch einen Soll-Ist-Vergleich:

grün: alles in Ordnung

rot: Konfigurationsfehler, Slave ist projektiert, fehlt aber im ASi-Strang

grau: Slave wird detektiert, ist aber noch nicht projektiert

## **Eingesetzte Hardware**

SIMATIC S7 Netzteil	PS 407 4A
SIMATIC S7 CPU mit Profibus DP	CPU 412-2 PD
	Best.Nr.:6ES7 412-2XG00-0AB0
	Firmware Version 3.0
Pepperl+Fuchs AS-i/Profibus-Gateway	VBG-PB-K20-D
Pepperl+Fuchs AS-i Power Extender	
Pepperl+Fuchs AS-i 4E/4A-Modul	
Pepperl+Fuchs AS-i Leuchttastermodul	
Pepperl+Fuchs AS-i Drehgeber	
Pepperl+Fuchs AS-i 2E-Sicherheitsmodul	
Pepperl+Fuchs AS-i 2E/2A-Sicherheitsmodul	
Pepperl+Fuchs AS-i Sicherheitsmonitor	
Not-Aus-Schalter	
Netzteil	

## **Eingesetzte Software**

Pepperl+Fuchs GSD-File für das AS-i/Profibus Gateway	
SIMATIC Step7	Version 5.1 Service Pack 3
Programmbeispieldatei	Ausgabestand: K5.1.3.0 PF_ASi14.zip

### Mitgeltende Unterlagen

Pepperl+Fuchs AS-Interface/Profibus Gateway Bedienungsanleitung  
 SIEMENS S7-400 Dokumentation

### Hardwarekonfiguration

Aus dem Hardwarekatalog werden die unter SIMATIC 400 befindlichen Baugruppen

1. Profilschiene
2. Netzteil PS 407 4A
3. CPU CPU 412-2 DP

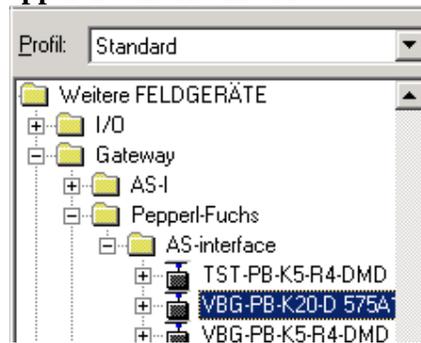
dem Projekt hinzugefügt. Bei der Auswahl der CPU Baugruppe ist auf die richtige Hardwareversion und die Firmwareversion zu achten.

Steckplatz	Baugruppe	Bestellnummer	Firmware	MPI-Adres...	E-Adresse	A-Adresse	Kommentar
1	PS 407 4A	6ES7 407-0DA01-0AA0					
2	CPU 412-2 DP	6ES7 412-2XG00-0AB0	V3.1	2			
X2	DP				4095*		
X1	MFV/DP			2	4094*		

Hardwarekonfiguration Netzteil und CPU-Baugruppe

Nachdem die SIMATIC Hardware der Hardwarekonfiguration hinzugefügt und der Profibus konfiguriert wurde, kann das Pepperl+Fuchs AS-i/Profibus-Gateway dem Projekt hinzugefügt werden.

Das AS-i/Profibus-Gateway befindet sich im Hardwarekatalog unter **Profibus/Weitere FELDGERÄTE/Gateway/Pepper-Fuchs/AS-interface**.



AS-i/Profibus-Gateway im Hardwarekatalog

Das AS-i/Profibus Gateway besitzt im Hardwarekatalog die Bezeichnung **VBG-PB-K20-D 575A1745** und kann nun per Drag and Drop dem Profibusstrang hinzugefügt werden.

Öffnet man das Gerät „VBG-PB-K20D 575A1745“ durch Betätigen des Pluszeichens im Hardwarekatalog erscheint eine Liste der möglichen Profibus Kommunikationsmodule.

Zu diesem Zeitpunkt sollte das gewünschte Profibus Kommunikationsmodul parametrieren werden. Dies erfolgt in folgenden Schritten:

1. Markieren des AS-i/Profibus-Gateway durch einen Mausklick auf das Slave Symbol. Danach wird am unteren Bildrand eine Tabelle angezeigt, welche Zeilen mit Steckplatz 0 beginnend enthält.

2. Auswahl des gewünschten Kommunikationsmoduls aus dem Hardwarekatalog. Hier „32 Byte Digital In/Out (0-31B)“.
3. Ziehen des ausgewählten Kommunikationsmoduls auf die Tabellenzeile Steckplatz 0.
4. Auswahl des gewünschten Kommunikationsmoduls „Flags + Fault Detector“ aus dem Hardwarekatalog. Diese Flags signalisieren in den einzelnen Bits den Betriebszustand des AS-i/DP Gateways und sollten im Applikationsprogramm ausgewertet werden.
5. Ziehen des ausgewählten Kommunikationsmoduls auf die Tabellenzeile Steckplatz 1.
6. Auswahl des gewünschten Kommunikationsmodul „Command Interface“ aus dem Hardwarekatalog. Hier „12 Byte Command Interface“.
7. Ziehen des ausgewählten Kommunikationsmoduls auf die Tabellenzeile Steckplatz 2.

Steckplatz	DP-Kennung	Bestellnummer / Bezeichnung	E-Adresse	A-Adresse	Kommentar
0	193	32 Byte Digital In/Out (0-31B)	0...31	0...31	
1	65	Flags + Fault Detector	32...33		
2	193	12 Byte Command Interface	512...523	512...523	
3					

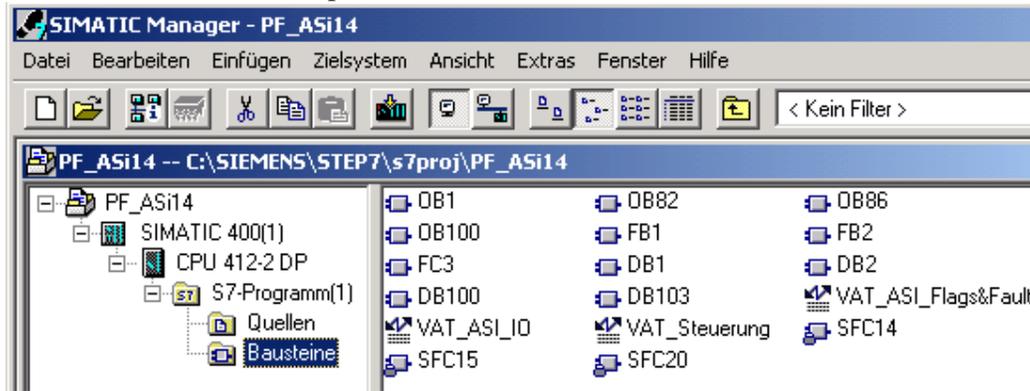
Kommunikationsmodule des AS-i/Profibus-Gateways

### **SIMATIC Step7 Bausteine**

OB1	Zyklischer Programmbaustein
OB82	Profibus Diagnosealarm. Dieser OB wird aufgerufen sobald ein Profibusteilnehmer in der Telegrammantwort das ExtDiagFlag gesetzt hat. Dieses ExtDiagFlag gibt einem Profibusteilnehmer die Möglichkeit einem Profibusmaster einen Fehlerzustand zu melden. Ist der OB82 in der CPU nicht vorhanden, so geht die CPU bei einem gesetzten ExtDiagFlag eines Profibusteilnehmers in den STOP-Zustand
OB 86	Profibus Peripheriefehler. Dieser OB wird aufgerufen, wenn der Profibusmaster den Ausfall eines Profibusteilnehmers erkennt.
OB100	Anlauf-OB. Dieser OB wird beim Anlauf der CPU einmalig ausgeführt.
FB1	FB zur Durchführung der ASi-Diagnose
FB2	FB zum Ausführen von Kommandos über die Mailbox
FC3	FC zur Visualisierung der ASi-Zustände
DB1	Instanzen-DB FB1
DB2	Instanzen-DB FB2
DB100	DB zur Anzeige der ASi-Slave-Diagnose
DB103	DB zur Visualisierung der ASi-Zustände
VAT_ASI_Flags&Fault	Variablen-tabelle zur Darstellung der Fehlermeldungen
VAT_ASI_IO	Variablen-tabelle zum Beobachten und Steuern der ASi-Eingangs- und Ausgangsbits
VAT_Steuerung	Variablen-tabelle zur Steuerung zum Auslesen der LCS-Liste und der Fehlerzähler sowie Anzeige eines Fehlers bei der Mailboxdiagnose
SFC14	Konsistente Daten eines DP-Normslaves lesen
SFC15	Daten konsistent auf einen DP-Normslave schreiben

SFC20

Kopieren des Inhaltes eines Speicherbereiches in einen anderen Speicherbereich



Step7 Programmbausteine

### Auslesen der binären Ein- und Ausgangsdaten

In der Hardwarekonfiguration wurden die 32 Byte I/O Daten des AS-i/DP Gateways auf die Eingang-/Ausgangbytes Adresse 0 bis 31 des Prozessabbildes gekoppelt.

	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1	EB 0	"IN_Flags_Slave1"	BIN	2#0000_0011	
2	EB 1	"IN_Slave2_Slave3"	BIN	2#1010_0000	
3	EB 2	"IN_Slave4_Slave5"	BIN	2#0000_0000	
4	EB 3	"IN_Slave6_Slave7"	BIN	2#0000_0000	
5	EB 4	"IN_Slave8_Slave9"	BIN	2#1100_0000	
6	EB 5	"IN_Slave10_Slave11"	BIN	2#0000_0000	
7	EB 6	"IN_Slave12_Slave13"	BIN	2#0000_0000	
8	EB 7	"IN_Slave14_Slave15"	BIN	2#0101_1101	
9	EB 8	"IN_Slave16_Slave17"	BIN	2#0110_0010	
10	EB 9	"IN_Slave18_Slave19"	BIN	2#0000_0000	
11	EB 10	"IN_Slave20_Slave21"	BIN	2#1000_0000	
12	EB 11	"IN_Slave22_Slave23"	BIN	2#0000_0000	
13	EB 12	"IN_Slave24_Slave25"	BIN	2#0000_1101	
14	EB 13	"IN_Slave26_Slave27"	BIN	2#0000_0000	
15	EB 14	"IN_Slave28_Slave29"	BIN	2#0000_0000	
16	EB 15	"IN_Slave30_Slave31"	BIN	2#0000_0000	
17	EB 16	"IN_reserved_Slave1B"	BIN	2#0000_0000	
18	EB 17	"IN_Slave2B_Slave3B"	BIN	2#0000_0000	
19	EB 18	"IN_Slave4B_Slave5B"	BIN	2#0000_0000	
20	EB 19	"IN_Slave6B_Slave7B"	BIN	2#0000_0000	
21	EB 20	"IN_Slave8B_Slave9B"	BIN	2#0000_0000	
22	EB 21	"IN_Slave10B_Slave11B"	BIN	2#0000_0000	
23	EB 22	"IN_Slave12B_Slave13B"	BIN	2#0000_0000	
24	EB 23	"IN_Slave14B_Slave15B"	BIN	2#0000_0000	
25	EB 24	"IN_Slave16B_Slave17B"	BIN	2#0000_0000	
26	EB 25	"IN_Slave18B_Slave19B"	BIN	2#0000_0000	
27	EB 26	"IN_Slave20B_Slave21B"	BIN	2#0000_0000	
28	EB 27	"IN_Slave22B_Slave23B"	BIN	2#0000_0000	
29	EB 28	"IN_Slave24B_Slave25B"	BIN	2#0000_0000	
30	EB 29	"IN_Slave26B_Slave27B"	BIN	2#0000_0000	
31	EB 30	"IN_Slave28B_Slave29B"	BIN	2#0000_0000	
32	EB 31	"IN_Slave30B_Slave31B"	BIN	2#0000_0000	
33	E 0.4	"IN_ASI_Config_Error"	BOOL	false	
34	F 0.5	"IN_ASI_Power_Fail"	BOOL		

Variablen-tabelle ASI-I/O-Daten

### **Auslesen der Fehlermeldungen**

Die direkt übertragenen AS-i Diagnose Informationen zur Fehlerauswertung sind aus den Eingangsbits des EW32 ersichtlich.

#### **Flags + Fault Detector**

- Bit 0 = Konfigurationsfehler
- Bit 1 = Slave mit Adresse NULL entdeckt
- Bit 2 = automatische Adressierung nicht möglich
- Bit 3 = automatische Adressierung verfügbar
- Bit 4 = Projektierungsmodus aktiv
- Bit 5 = nicht im Normalbetrieb
- Bit 6 = AS-i Power Fail
- Bit 7 = AS-i Master ist Offline
- Bit 8 = Peripheriefehler
- Bit 9 = reserviert
- Bit10 = reserviert
- Bit11 = reserviert
- Bit12 = Erdschluss
- Bit13 = Überspannung
- Bit14 = Störspannung
- Bit15 = Doppeladresse

	Operand	Symbol	Anzei	Statuswert
1	EW 32	"Flags_and_Fault_Detector"	BIN	2#0000_0000_0000_0000
2	E 32.0	"Peripheriefehler"	BOOL	false
3	E 32.4	"Erdschluss"	BOOL	false
4	E 32.7	"Doppeladresse"	BOOL	false
5	E 33.0	"Konfigurationsfehler"	BOOL	false

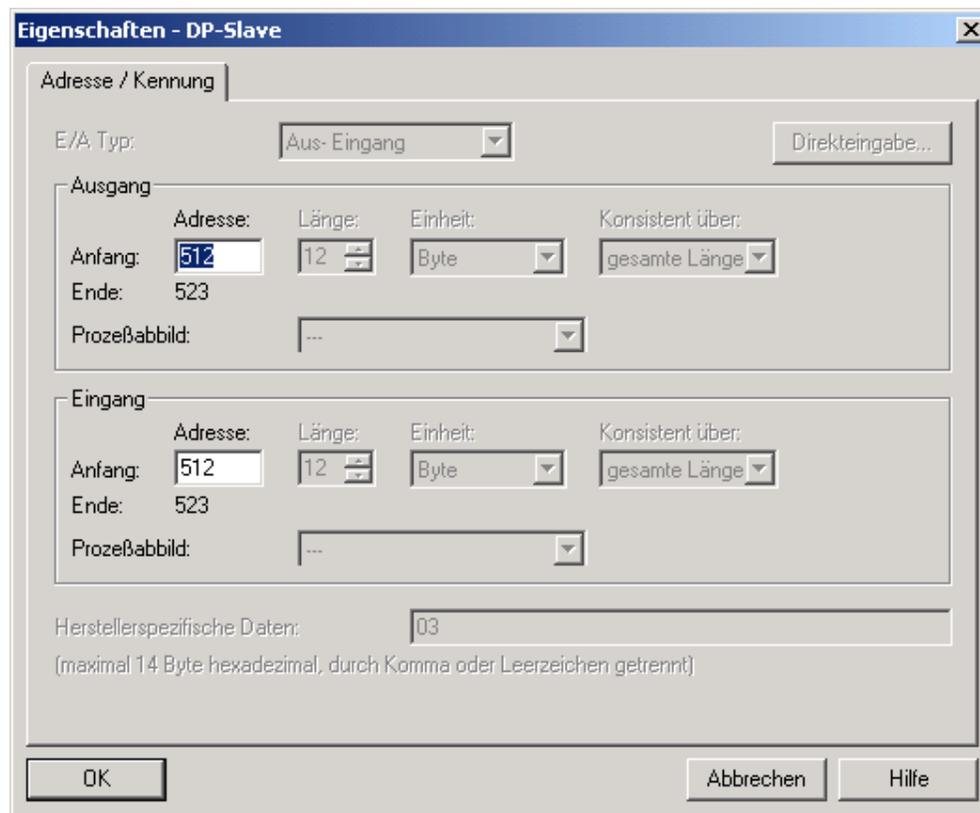
Variablentabelle Flags and Fault

### **Auslesen der ASi-Slave-Diagnose**

Zwecks Auslesen der LPS, LDS, LCS und Deltaliste sowie der Liste der Fehlerzähler ruft der OB1 den FB1 auf. Im Gegensatz zu den LPS und LDS wird die LCS und der die Liste der Fehlerzähler nur durch Setzen der Ein-/Ausgänge ReadLCS und ReadErrCounter in der Variablentabelle VAT\_Steuerung gestartet. Solange ein Konfigurationsfehler vorliegt, wird die Deltaliste ebenfalls zyklisch ausgelesen, nach dem Verschwinden des Konfigurationsfehlers wird die Deltaliste noch einmalig ausgelesen. Der FB1 ist so programmiert, dass sein Ablauf völlig unabhängig von der Hardwarekonfiguration programmiert ist und somit keine Änderungen zur Anpassung in seinem Ablauf vorgenommen werden müssen. Lediglich die Werte an den Eingängen müssen angepasst werden.

**FB1 Netzwerk1:** Der Wert des Einganges DBNr wird in die temporäre Variable DBDest kopiert. Bei der Überwachung mehrerer ASi-Gateways muss für jedes Gateway ein separater DB zur Speicherung der ASi-Gateway-Listen verwendet werden.

**FB1 Netzwerk2:** Das Auslesen der LCS und der Fehlerzähler sowie der Deltaliste, wenn ein vorliegender Konfigurationsfehler behoben wird, werden im Netzwerk 1 des FB1 behandelt. Bei einer negativen Flanke des ConfigErrBit wird ReadDelta gesetzt. Nun wird überprüft, ob der Eingang Kreis gesetzt oder nicht gesetzt ist. Dieser Eingang bestimmt, ob ein Einfachmaster angeschlossen ist und ob bei einem Doppelmaster der erste oder der zweite Kreis verwendet wird. Der entsprechende Wert wird nun nach FB2\_Kreis geladen. Ist ReadLCS, ReadErrCounter und ReadDelta nicht gesetzt, so wird das Netzwerk 2 komplett übersprungen und die Bearbeitung des Programms wird im Netzwerk 3, Sprungmarke net3, fortgesetzt. Wenn die Bearbeitung des FB2 noch nicht abgeschlossen ist, springt das Programm zur Sprungmarke FBRd, an welcher sich der Aufruf des FB2 befindet. Ansonsten wird überprüft, ob ReadLCS gesetzt ist. Ist ReadLCS gesetzt, wird 60<sub>HEX</sub> in FB2\_Befehl und 0<sub>HEX</sub> in die drei FB2-Parameter geladen. 60<sub>HEX</sub> ist der Mailboxbefehl zum Lesen der LCS (GET\_LCS). Die einzelnen Mailboxbefehl und ihre Parameter sind dem Handbuch des AS-Interface/Profibus Gateways zu entnehmen. Das Programm setzt beim Aufruf des FB2 fort. Ist ReadLCS nicht gesetzt, wird der Inhalt von ReadDelta überprüft. Ist ReadDelta gesetzt wird 57<sub>HEX</sub> nach FB2\_Befehl und 0<sub>HEX</sub> in die Parameter 1-3 geladen und es wird mit dem Aufruf des FB2 fortgefahren. Ist ReadDelta nicht gesetzt gewesen, wird ReadErrCounter überprüft. Hier wird noch zusätzlich der Zustand der Hilfsvariablen ErrCounter0\_7A bis ErrCounter24B\_31B überprüft. Durch die Benutzung der Hilfsvariablen wird die Liste der Fehlerzähler in 8 Teillisten aufgeteilt, was aufgrund der Datenmenge notwendig ist. Je nachdem in welcher Teilliste sich das Programm gerade befindet, wird die Hilfsvariable ErrCounterx\_y gesetzt. In die Variable FB2\_Befehl wird 66<sub>HEX</sub>, Mailboxbefehl für GET\_TEC\_X), geladen. In die Variable FB2\_Parameter3 wird der Wert 0<sub>HEX</sub> geladen. Die Variable FB2\_Parameter2 erhält den Wert 8<sub>HEX</sub>, da stets 8 Byte auf einmal gelesen werden. Je nachdem, welche Teilliste gelesen werden soll, wird in die Variable FB2\_Parameter1 der Wert 0, 8, 16, 24, 32, 40, 48 oder 56 geschrieben. Beim Aufruf des FB2 ASi-Mailbox wird zum Einen die Startadresse des Managementkanals (Command Interface), welche mit dem Eingang StartadresseModul übergeben wird, und zum Anderen die befehlspezifischen Parameter übergeben. Die Startadresse des Managementkanals ist einstellbar.



Startadresse des Managementkanals in der Hardwarekonfiguration

Die Variable FB2\_Start startet den FB2. FB2\_Busy zeigt die momentane Bearbeitung an und FB2\_Complete die fertige Bearbeitung. Ist ein Fehler aufgetreten, erkennt man dies anhand der Variablen FB2\_Error und kann FB2\_Status den Fehlercode entnehmen. Nach Beendigung des FB2 beginnt die Ergebnisauswertung. Vor dem Kopieren der Daten wird überprüft, ob FB2\_Complete und FB2\_Error nicht gesetzt ist. Sowie ob ReadLCS, bzw. ReadDelta oder ReadErrCounter, gesetzt ist. Bei ReadErrCounter findet wie bei der Übergabe der Mailboxbefehlsparameter noch zusätzlich die Abfrage nach der Teilliste und der Variablen AuswertungOK statt. ReadLCS, bzw. ReadDelta oder ReadErrCounter, wird zurückgesetzt und die Daten werden als Block von dem DB2, FB2 InstanzenDB, in den DBDest, beinhaltet die ASI Gateway Listen eines Masters, kopiert. Beim Auslesen der Liste der Fehlerzähler wird zusätzlich noch die Variable AuswertungOK zurückgesetzt. Ist während der Bearbeitung des FB2 ein Fehler aufgetreten und die Bearbeitung des FB2 ist abgeschlossen, wird der Ausgang Error des FB1 gesetzt und ReadLCS, ReadDelta, ReadErrCounter sowie die Hilfsvariablen aller Teillisten werden zurückgesetzt.

**FB2:** Die Fehlervariablen des FB2 werden zurückgesetzt. Durch betätigen von Reset wird der FB2 in seinen Startzustand zurückgesetzt. Die Modulstartadresse wird gespeichert und in das Pointerformat umgewandelt. Der Zeiger P\_Out\_Data wird initialisiert. Über die SFC14 werden die Daten eingelesen. Ist bei der Durchführung der SFC14 ein Fehler aufgetreten, so wird die Variable SFC14Error gesetzt und das Programm setzt mit der Fehlerauswertung, Sprungmarke Err im Netzwerk 9, fort. Ansonsten wird aus der Befehlsantwort das Togglebit bestimmt. Durch eine Word-Und-Verknüpfung von InData.Ergebnis und 80<sub>HEX</sub> wird das Togglebit herausgefiltert. Mit Hilfe eines Vergleiches der Und-Verknüpfung mit 80<sub>HEX</sub> wird die BOOL-Variable TooglebitEcho je nach Zustand des Togglebits gesetzt oder nicht. Ist der

Ausgang Busy nicht gesetzt, so wird an der Sprungmarke End2 , Ende Netzwerk 5, fortgefahren. TZeitueberwachung wird als Impuls gestartet. BefehlEcho wird mit BefehlSend und TogglebitEcho mit TogglebitSend verglichen. Stimmen die Werte überein, setzt das Programm an der Sprungmarke STA1 mit der Überprüfung des Statusbits fort. Ist dieses 0 wurde der Befehl ohne Fehler beendet, ansonsten wird StatusError gesetzt. Ist der Vergleich nicht wahr, so wird bei gesetzter Zeitueberwachung und nicht gesetzter Tzeitueberwachung TimeoutError gesetzt und mit der Fehlerauswertung im Netzwerk 9 fortgesetzt. Liefert die UND-Verknüpfung keine 1 als Ergebnis wird Zeitueberwachung gesetzt. Wurde der Befehl erfolgreich ausgeführt, StatusOK und Busy gesetzt, so werden die Variablen Zeitueberwachung, Error, Busy und Status zurückgesetzt, bzw. mit 0 geladen und die übermittelten Daten werden von InData in AsiMailboxData kopiert. Complete wird gesetzt. Ist Start gesetzt und Busy nicht, so werden die Parameter für die Profibusübertragung des Befehls zum ASi-Gateway gesetzt. Das Togglebit wird invertiert und in die Ausgangsdaten eingetragen. Start, Complete und Error werden zurückgesetzt und Status sowie AsiMailboxData werden gelöscht. Mit Hilfe der SFC15 werden die Daten von dem Quellbereich OutData konsistent zu dem ASi-Gateway übertragen. Anhand eines Vergleiches von RET\_VAL\_SFC15 und 0 wird der Variable SFC15Error gesetzt oder nicht. Beinhaltet die Variable RET\_VAL\_SFC15 den Wert 0, so wurde die SFC15 fehlerfrei ausgeführt. Im Netzwerk 9 wird gegebenenfalls der aufgetretene Fehler ausgewertet. Start, Busy und Zeitueberwachung werden gesetzt und Complete und Error zurückgesetzt. Nacheinander werden nun StatusError, SFC14Error, SFC15Error und TimeoutError abgefragt und der entsprechende Wert in Status geladen.

**FB1 Netzwerk2:** Wenn der FB2 gerade von dem Netzwerk 1 benutzt wird, wird das Netzwerk 2 komplett übersprungen. Ist der FB2 mit dem Auslesen der LDS oder der Deltaliste beschäftigt, so wird der Teil des Netzwerkes 2 zum Auslesen der LPS übersprungen und das Programm setzt an der Sprungmarke LPSe fort. Wird der FB2 nicht gerade zum Auslesen der LPS benutzt, wird LPS\_Start gesetzt. Der FB2 wird mit dem Eingang des FB1 StartadresseModul und dem Wert 44<sub>HEX</sub>, ASi-Mailboxbefehl GET\_LPS, als Wert für die Variable Befehl gestartet. Ist der FB2 abgeschlossen und es ist kein Fehler aufgetreten, so werden die Daten der LPS von dem DB2, FB2 InstanzenDB, in den DBDest, ASI Gateway Listen, kopiert. Ist während der Bearbeitung ein Fehler aufgetreten, so wird der Ausgang Error gesetzt.

Das Auslesen der LDS und der Deltaliste läuft ähnlich ab. Ist der FB2 gerade mit dem Auslesen der LPS beschäftigt, so fährt das Programm mit der Bearbeitung des Netzwerk 3 fort. Ist der FB2 mit dem Auslesen der Deltaliste beschäftigt, so setzt das Programm an der Sprungmarke RDlt fort. LDS\_Start wird gesetzt, wenn der FB2 nicht bereits mit dem Auslesen der LDS beschäftigt ist und somit LDS\_Busy nicht gesetzt ist. Anschließend folgt der Aufruf des FB2. In die Variable Befehl wird der Wert 46<sub>HEX</sub>, ASi-Mailboxbefehl GET\_LDS, geschrieben. Wenn LDS\_Complete gesetzt und während der Bearbeitung des FB2 kein Fehler aufgetreten ist, so werden die Daten der LDS von dem DB2 in den DBDest kopiert. Ist die Bearbeitung des FB2 abgeschlossen, jedoch aber ein Fehler aufgetreten, so wird der Ausgang Error auf 1 gesetzt. Wenn LDS\_Busy gesetzt ist, wird der FB1 beendet. Ist ConfigErrBit gesetzt, so wird er nicht beendet. Wenn Delta\_Busy nicht gesetzt ist, wird Delta\_Start gesetzt und der FB2 wird mit dem Befehl 57<sub>HEX</sub>, ASi-Mailboxbefehl GET\_DELTA, gestartet. Wenn Delta\_Complete gesetzt ist und während der Bearbeitung des FB2 kein Fehler aufgetreten ist, werden die Daten der Deltaliste von dem DB in den DBDest kopiert. Bei einem vorliegenden Fehler des FB2 wird der Ausgang Error gesetzt.

Adresse	Name	Typ	Anfang	Aktualwert	Kommentar
0.0	LPS_Liste.Slave7A_0A	BYTE	B#16#0	B#16#06	
1.0	LPS_Liste.Slave15A_8A	BYTE	B#16#0	B#16#C5	
2.0	LPS_Liste.Slave23A_16A	BYTE	B#16#0	B#16#13	
3.0	LPS_Liste.Slave31A_24A	BYTE	B#16#0	B#16#02	
4.0	LPS_Liste.Slave7B_0B	BYTE	B#16#0	B#16#00	
5.0	LPS_Liste.Slave15B_8B	BYTE	B#16#0	B#16#00	
6.0	LPS_Liste.Slave23B_16B	BYTE	B#16#0	B#16#00	
7.0	LPS_Liste.Slave31B_24B	BYTE	B#16#0	B#16#00	
8.0	LDS_Liste.Slave7A_0A	BYTE	B#16#0	B#16#06	
9.0	LDS_Liste.Slave15A_8A	BYTE	B#16#0	B#16#C4	
10.0	LDS_Liste.Slave23A_16A	BYTE	B#16#0	B#16#13	
11.0	LDS_Liste.Slave31A_24A	BYTE	B#16#0	B#16#02	
12.0	LDS_Liste.Slave7B_0B	BYTE	B#16#0	B#16#00	
13.0	LDS_Liste.Slave15B_8B	BYTE	B#16#0	B#16#00	
14.0	LDS_Liste.Slave23B_16B	BYTE	B#16#0	B#16#00	
15.0	LDS_Liste.Slave31B_24B	BYTE	B#16#0	B#16#00	
16.0	LCS_Liste.Slave7A_0A	BYTE	B#16#0	B#16#00	
17.0	LCS_Liste.Slave15A_8A	BYTE	B#16#0	B#16#05	
18.0	LCS_Liste.Slave23A_16A	BYTE	B#16#0	B#16#00	
19.0	LCS_Liste.Slave31A_24A	BYTE	B#16#0	B#16#00	
20.0	LCS_Liste.Slave7B_0B	BYTE	B#16#0	B#16#00	
21.0	LCS_Liste.Slave15B_8B	BYTE	B#16#0	B#16#00	
22.0	LCS_Liste.Slave23B_16B	BYTE	B#16#0	B#16#00	
23.0	LCS_Liste.Slave31B_24B	BYTE	B#16#0	B#16#00	
24.0	Deltaliste.Slave7A_0A	BYTE	B#16#0	B#16#00	
25.0	Deltaliste.Slave15A_8A	BYTE	B#16#0	B#16#01	
26.0	Deltaliste.Slave23A_16A	BYTE	B#16#0	B#16#00	
27.0	Deltaliste.Slave31A_24A	BYTE	B#16#0	B#16#00	
28.0	Deltaliste.Slave7B_0B	BYTE	B#16#0	B#16#00	

DB mit den ASI-Gateway-Listen

### Soll-Ist-Vergleich

Der Soll-Ist-Vergleich wird in der FC3 durchgeführt, welche direkt vom OB1 aufgerufen wird. Die FC3 wird die Nummer des DBs, welcher die ASI-Gateway-Listen des zuvergleichenden Kreises erhält, sowie die Nummer des DBs, in welchem die Zustände der einzelnen ASI-Slaves dargestellt werden sollen, übergeben. Die an den Eingängen anliegenden Werte werden in den temporären Variablen DB\_ListenDest und DB\_VisuDest gespeichert. Die nachfolgenden Anweisungen werden im Folgenden lediglich für den Slave mit der Adresse 1 erklärt, da sie für die 62 Slaves vollkommen identisch sind. Wenn das zu

dem Slave gehörige LPS-Bit gesetzt ist, folgt kein Sprung zu DL01, sondern das LDS-Bit wird überprüft. Beinhaltet dieses den Wert 1, so wird an der Sprungmarke ok01 Slave1A\_gruen gesetzt und die Anzeigen für rot und grau zurückgesetzt. Beinhaltet das LDS-Bit den Wert 0, so wird die Anzeige rot für den Slave 1A gesetzt und die beiden anderen, gruen und grau, zurückgesetzt. War das LPS-Bit nicht gesetzt, so wird an der Sprungmarke DL01 der Status des LDS-Bits überprüft. Ist dieses gesetzt, so wird Slave1A\_grau gesetzt und Slave1A\_gruen sowie Slave1A\_rot zurückgesetzt. Ist das LDS-Bit nicht gesetzt, ist der Slave weder detektiert noch projiziert und somit werden alle drei Zustandsanzeigen zurückgesetzt.

Adresse	Name	Typ	Anfang	Aktualwert	Kommentar
0.0	Slave1A_gruen	BOOL	FALSE	FALSE	
0.1	Slave1A_rot	BOOL	FALSE	TRUE	
0.2	Slave1A_grau	BOOL	FALSE	FALSE	
0.3	Slave2A_gruen	BOOL	FALSE	TRUE	
0.4	Slave2A_rot	BOOL	FALSE	FALSE	
0.5	Slave2A_grau	BOOL	FALSE	FALSE	
0.6	Slave3A_gruen	BOOL	FALSE	FALSE	
0.7	Slave3A_rot	BOOL	FALSE	FALSE	
1.0	Slave3A_grau	BOOL	FALSE	FALSE	
1.1	Slave4A_gruen	BOOL	FALSE	FALSE	
1.2	Slave4A_rot	BOOL	FALSE	FALSE	
1.3	Slave4A_grau	BOOL	FALSE	FALSE	
1.4	Slave5A_gruen	BOOL	FALSE	FALSE	
1.5	Slave5A_rot	BOOL	FALSE	FALSE	
1.6	Slave5A_grau	BOOL	FALSE	TRUE	
1.7	Slave6A_gruen	BOOL	FALSE	FALSE	
2.0	Slave6A_rot	BOOL	FALSE	FALSE	
2.1	Slave6A_grau	BOOL	FALSE	FALSE	
2.2	Slave7A_gruen	BOOL	FALSE	FALSE	
2.3	Slave7A_rot	BOOL	FALSE	FALSE	
2.4	Slave7A_grau	BOOL	FALSE	FALSE	
2.5	Slave8A_gruen	BOOL	FALSE	TRUE	

Zustandsvisualisierung der ASI-Slaves