

ASi-SafetySlaves

Dieses Beispielprogramm zeigt die Feststellung, welcher ASi-Sicherheitslave ausgelöst ist.

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-K5-R4-DMD
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	Ausgabestand: K5.1.3.0
Programmbeispieldatei		PF_ASi10.zip

Mitgeltende Unterlagen

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

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.
FC2	Überprüfung der Funktionalität eines ASi-Sicherheitslaves
VAT_SafetySlaves	Variablen-tabelle zum Beobachten der Ausgänge der Sicherheitslaves



Bild 1 – Step7 Programmbausteine

Programmablauf

Das Auslösen eines ASi-Sicherheitslaves wird im FC2 festgestellt. In diesem Beispiel ist der Aufruf des FC im OB1 programmiert. Der Aufruf muss für jeden ASi-Sicherheitslave extra programmiert werden.

Zuerst muss aus der übermittelten Slaveadresse sowie der Startadresse des Datenfeldes die Nummer des Eingangsbytes ermittelt werden. Die Bytenummer wird in das Pointerformat umgewandelt. Mit Hilfe einer UND-Wortverknüpfung wird nun überprüft, ob es sich um eine gerade oder ungerade Slaveadresse handelt. Bei einer geraden Slaveadressen wird das Eingangsbyte um 4 Bits nach rechts geschoben und anschließend in der Hilfsvariablen aktuelleBits gespeichert. Bei einer ungeraden Slaveadresse werden die Eingangsbits direkt in der Hilfsvariablen aktuelleBits gespeichert. Je nachdem ob es sich um einen ein- oder zweikanaligen Sicherheitslave handelt, wird der weitere Ablauf im Netzwerk 2 oder 3 ausgeführt.

Netzwerk 2 (einkanalig): Mit Hilfe einer UND-Verknüpfung werden die nicht zu der eingelesenen ASi-Slaveadresse gehörenden vier Bits auf 0 gesetzt. Das Ergebnisbyte dieser Operation wird mit 0 verglichen. Ist der Vergleich falsch, arbeitet der ASi-Sicherheitslave

richtig, das Programm setzt an der Sprungmarke ok fort und die Ausgänge Kanal1 und Kanal2 werden zurückgesetzt. Ist der Vergleich aber wahr, so ist der ASi-Sicherheitslave ausgelöst, das Programm fährt an der Sprungmarke fehl fort und die Ausgänge Kanal1 und Kanal2 werden gesetzt.

Netzwerk 3 (zweikanalig): Anhand einer UND-Verknüpfung werden die nicht zu der eingelesenen Slaveadresse gehörigen vier Bits sowie die ersten zwei zu der Slaveadresse gehörigen Bits auf 0 gesetzt. Die restlichen zwei Bits werden mit 0 verglichen, ist der Vergleich falsch, so wird die Hilfsvariable Kanal1Hilf mit 0 geladen, der Ausgang Kanal1 zurückgesetzt und das Programm fährt mit der Überprüfung des zweiten Kanals (Kan2) fort. Ist der Vergleich wahr, so wird überprüft, ob die Hilfsvariable Kanal1Hilf den Wert 1111_{BIN} entspricht. Das Hinzuziehen der Hilfsvariablen ist nötig, da laut Definition die beiden Bits des einen Kanals bei einem zweikanaligen ASi-Slave für einen Zyklus lang 00_{BIN} sein dürfen. Da die Zykluszeit des ASi-Systems nicht der des Step7 Programms entspricht, wird der Ausgang nicht sofort nach zwei aufeinanderfolgenden Zyklen gesetzt, da es auch hintereinander zu einem mehrmaligen Einlesen der Bitkombination 00_{BIN} führen kann, ohne dass der Kanal ausgelöst ist. Ein sofortiges Setzen nach zwei aufeinanderfolgenden Eingangsbits 00_{Bin} würde zu einigen kurzen Fehlsetzungen des Ausganges führen. Da es sich bei diesem Programm nicht um die Sicherheitssteuerung an sich handelt, sondern nur um eine Visualisierung der ASi-Sicherheitslaves, ist die zusätzliche Verzögerung von einigen Millisekunden nicht tragisch, zumal sie zu der Einlesezeit der SPS nicht sonderlich in das Gewicht fallen und für das menschliche Auge keines Weges wahrnehmbar ist. Für den Vergleichswert wurde 1111_{BIN} gewählt, da mit Hilfe einer zwischenzeitlich eingebundenen Hilfsvariablen festgestellt wurde, dass bei der hier vorliegenden Kombination von der Zykluszeit des ASi-Systems und des Programms bis zu 8 mal hintereinander die Kombination 00_{BIN} der beiden Eingangsbits auftreten kann, ohne dass der ASi-Sicherheitslave ausgelöst wurde.

Ist der Vergleich korrekt, so wird an der Sprungmarke K1wr der Ausgang Kanal1 gesetzt und die Hilfsvariable Kanal1Hilf mit 0 geladen. Das Programm setzt mit der Überprüfung des zweiten Kanals fort.

Beinhaltet die Hilfsvariable Kanal1Hilfe nicht den Wert 1111_{BIN} , so wird überprüft, ob der Ausgang Kanal1 bereits gesetzt ist. Besitzt dieser bereits den Wert 1, so setzt das Programm an der Sprungmarke K1wr fort. Ist der Ausgang nicht gesetzt, so wird die Hilfsvariable Kanal1Hilfe inkrementiert und das Programm setzt an der Sprungmarke Kan2 mit Überprüfung des zweiten Kanals fort. Bei der Überprüfung des zweiten Kanals werden mit Hilfe einer UND-Verknüpfung die Bits, welche nicht zu der eingelesenen Slaveadresse gehören, sowie die beiden letzten Bits der Slaveadresse auf 0 gesetzt. Der weitere Ablauf ist identisch mit dem des ersten Kanals.

Ein- und Ausgänge der FC

Eingänge:

SlaveAddr:	Die Adresse des ASi-Sicherheitslave (1...63 in Hex)
AnzahlKanaele:	Auswahl, ob ein ein- (FALSE) oder zweikanaliger (TRUE) ASi-Sicherheitslaves angeschlossen ist
StartadresseDatenfeld:	Die in der Hardwarekonfiguration eingestellte Startadresse für das Eingangsdatenfeld (in Hex)

Ein- und Ausgänge:

- Kanal1: Zustand des ersten Kanals, bei FALSE ist alles in Ordnung, bei TRUE ist der Kanal ausgelöst
- Kanal2: Zustand des zweiten Kanals, bei FALSE ist alles in Ordnung, bei TRUE ist der Kanal ausgelöst
- Kanal1Hilfe: Hilfsvariable zur Steuerung des ersten Kanals bei einem zweikanaligen Slave, ist lediglich mit einer Variablen oder einem Merker, welcher nicht verändert wird, zur Speicherung zu belegen
- Kanal2Hilfe: Hilfsvariable zur Steuerung des zweiten Kanals bei einem zweikanaligen Slave, ist lediglich mit einer Variablen oder einem Merker, welcher nicht verändert wird, zur Speicherung zu belegen

Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar
0.0	in	SlaveAddr	WORD		Adresse des ASI-Sicherheitslave
2.0	in	AnzahlKanäle	BOOL		ein(FALSE)- oder zwei(TRUE)kanalig
4.0	in	StartadresseDatenfeld	WORD		In der Hardwarekonfiguration eingestellte Startadresse
	out				
6.0	in_out	Kanal1	BOOL		belegt oder frei
6.1	in_out	Kanal2	BOOL		belegt oder frei
8.0	in_out	Kanal1Hilfe	INT		nur bei zweikanaligen Slaves benötigt
10.0	in_out	Kanal2Hilfe	INT		nur bei zweikanaligen Slaves benötigt

Bild 2 – Ein- und Ausgänge der FC2

Variablentabelle VAT_SafetySlaves

In der Variablentabelle VAT_SafetySlaves können die einzelnen Kanäle der ASi-Sicherheitslaves dargestellt werden. Bei Slaves der Sicherheitsstufe 4 kann im Gegensatz zu Slaves der Sicherheitsstufe 2 auf die Darstellung beider Kanäle verzichtet werden, da sie stets die selben Wert aufweisen.

	Operand	Symbol	Anzei	Statuswert	Steuerwert
1	M 30.0	"SafetySlave1_Kanal1"	BOOL	false	
2	M 30.1	"SafetySlave1_Kanal2"	BOOL	false	
3	M 30.2	"SafetySlave2_Kanal1"	BOOL	false	
4	M 30.3	"SafetySlave2_Kanal2"	BOOL	true	
5	M 30.4	"SafetySlave25_Kanal1"	BOOL	false	
6	M 30.5	"SafetySlave25_Kanal2"	BOOL	false	

Bild 3 – Variablentabelle ausgelöste Sicherheitslaves: Slave 2 Kanal2 ist ausgelöst