

## **ASi-SafetySlaves mit Eintrag in den Diagnosepuffer**

Das vorliegende Beispielprogramm erklärt wie die Feststellung, welcher ASi-Sicherheitslave ausgelöst ist, sowie die Speicherung der letzten ausgelösten ASi-Sicherheitslaves vorzunehmen 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_ASi11.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.
OB86	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 ASi-Sicherheitslaves
SFC52	Anwenderdefiniertes Diagnoseereignis in den Diagnosepuffer schreiben



Bild 1 – Step7 Programmabusteine

## **Programmablauf**

Das Auslösen eines ASi-Sicherheitslaves wird, zusammen mit der Eintragung in den Diagnosepuffer, in der FC2 festgestellt. In diesem Beispiel ist der Aufruf der FC2 im OB1 programmiert. Der Aufruf muss für jeden ASi-Sicherheitslave separat programmiert werden. Die Werte der Eingänge SlaveAddr und StartadresseDatenfeld werden in temporären Hilfsvariablen, zur späteren Übergabe an die SFC, gespeichert. Zuerst muss aus der übermittelten Slaveadresse und 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 ASi-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-Slave richtig, das Programm setzt an der Sprungmarke ok fort. Ist der Vergleich aber wahr, so wurde der ASI-Slave ausgelöst, das Programm fährt an der Sprungmarke fehl fort.

Sprungmarke ok: Ist die Variable Kanal1 bereits zurückgesetzt - an dieser Stelle könnte genauso gut die Variable Kanal2 überprüft werden, da bei einem zweikanaligen ASI-Sicherheitslave die beiden Ausgänge stets gleichzeitig gesetzt und zurückgesetzt werden - , wird die FC beendet. Ist die Variable gesetzt, so werden Kanal1 und Kanal2 zurückgesetzt und die SFC52 wird aufgerufen. Die Überprüfung des Zustandes der Variable Kanal1 ist notwendig, damit die SFC52 nur einmalig beim Zurücksetzen durchlaufen wird und nicht zyklisch, solange die Ausgänge Kanal1 und Kanal2 nicht gesetzt sind.

Sprungmarke fehl: Ist die Variable Kanal1 gesetzt - anstatt der Variablen Kanal1 wäre die Verwendung der Variablen Kanal2 ebenso möglich - , wird die FC an dieser Stelle beendet. Ist die Variable Kanal1 nicht gesetzt, so werden die Ausgänge Kanal1 und Kanal2 gesetzt und die SFC52 wird aufgerufen.

**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 und überprüft ob der Ausgang Kanal1 rückgesetzt. Ist dies nicht der Fall, wird Kanal1 rückgesetzt und die SFC52 wird aufgerufen, ist dies jedoch bereits geschehen, fährt das Programm mit der Überprüfung des zweiten Kanals an der Sprungmarke Kan2 fort.

Ist der Vergleich wahr, so wird überprüft, ob die Hilfsvariable Kanal1Hilf den Wert  $1111_{\text{BIN}}$  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  $00_{\text{BIN}}$  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  $00_{\text{BIN}}$  führen kann, ohne dass der Kanal ausgelöst ist. Ein sofortiges Setzen nach zwei aufeinanderfolgenden Eingangsbits  $00_{\text{BIN}}$  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  $1111_{\text{BIN}}$  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  $00_{\text{BIN}}$  der beiden Eingangsbits auftreten kann, ohne dass der ASI-Sicherheitslave ausgelöst wurde.

Ist der Vergleich korrekt, so setzt das Programm seinen Durchlauf an der Sprungmarke K1wr fort. Beinhaltet die Hilfsvariable Kanal1Hilfe nicht den Wert  $1111_{\text{BIN}}$ , so wird überprüft, ob der Ausgang Kanal1 bereits gesetzt ist. Besitzt dieser bereits den Wert 1, so setzt das Programm an der Sprungmarke Kan2 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.

Sprungmarke K1wr: Der Ausgang Kanal1 wird gesetzt, die Hilfsvariable Kanal1Hilf mit 0 geladen. Die SFC52 wird aufgerufen und das Programm fährt mit der Überprüfung des zweiten Kanals fort.

## Ein- und Ausgänge der FC2

### Eingänge:

SlaveAddr: Die Adresse des ASI-Sicherheitsslave (1...63 in Hex)  
 AnzahlKanaele: Auswahlmöglichkeit, ob ein- (FALSE) oder zweikanalig (TRUE)  
 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-Sicherheitsslave
2.0	in	AnzahlKanaele	BOOL		ein(FALSE)- oder zwei(TRUE)kanalig
4.0	in	StartadresseDatenfeld	WORD		in der Hardwarekonfiguration eingestellt
	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
n.n	return	RetAddr	WORD		

Bild 2 – Ein- und Ausgänge der FC2

## Variablentabelle VAT\_SafetySlaves

In der Variablentabelle VAT\_SafetySlaves können die einzelnen Kanäle der ASI-Sicherheitsslaves 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 – Variablen-tabelle ausgelöste Sicherheitsslaves: Slave 2 Kanal1 ausgelöst

### Ein- und Ausgänge der SFC52

- IN0: Festlegung, ob die anwenderdefinierte Diagnosemeldung an alle angemeldeten Teilnehmer gesendet werden soll
- IN1: Die Ereignis-ID muss in dem Format W#16#8xyz, W#16#9xyz, W#16#Axyz oder W#16#Bxyz eingegeben werden. Für anwenderdefinierte Diagnoseereignisse besteht die Auswahl zwischen W#16#Axyz und W#16#Bxyz. A und B gibt die Fehlerklasse an. x steht für kommendes (1) oder gehendes (0) Ereignis. yz ist der Hexadezimalwert der Meldungsnummer, welche in der Meldungsprojektierung vergeben wurde
- IN2: Die 1-wortlange Zusatzinformation kann als Begleitwert in den Meldetext integriert werden.
- IN3: Die 2-wortlange Zusatzinformation kann als Begleitwert in den Meldetext integriert werden.
- RET\_VAL: Bei korrekter Ausführung der SFC beträgt die Fehlerinformation den Wert 0, ansonsten den Fehlerwert.

Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar
0.0	in	IN0	BOOL		
2.0	in	IN1	WORD		
4.0	in	IN2	ANY		
14.0	in	IN3	ANY		
24.0	out	RET_VAL	INT		

Bild 4 – Ein- und Ausgänge der SFC52

## Meldungsprojektierung

Die Meldungsprojektierung wird im SIMATIC Manager unter „Bearbeiten“ → „Spezielle Objekteigenschaften“ → „Meldung...“ gestartet.

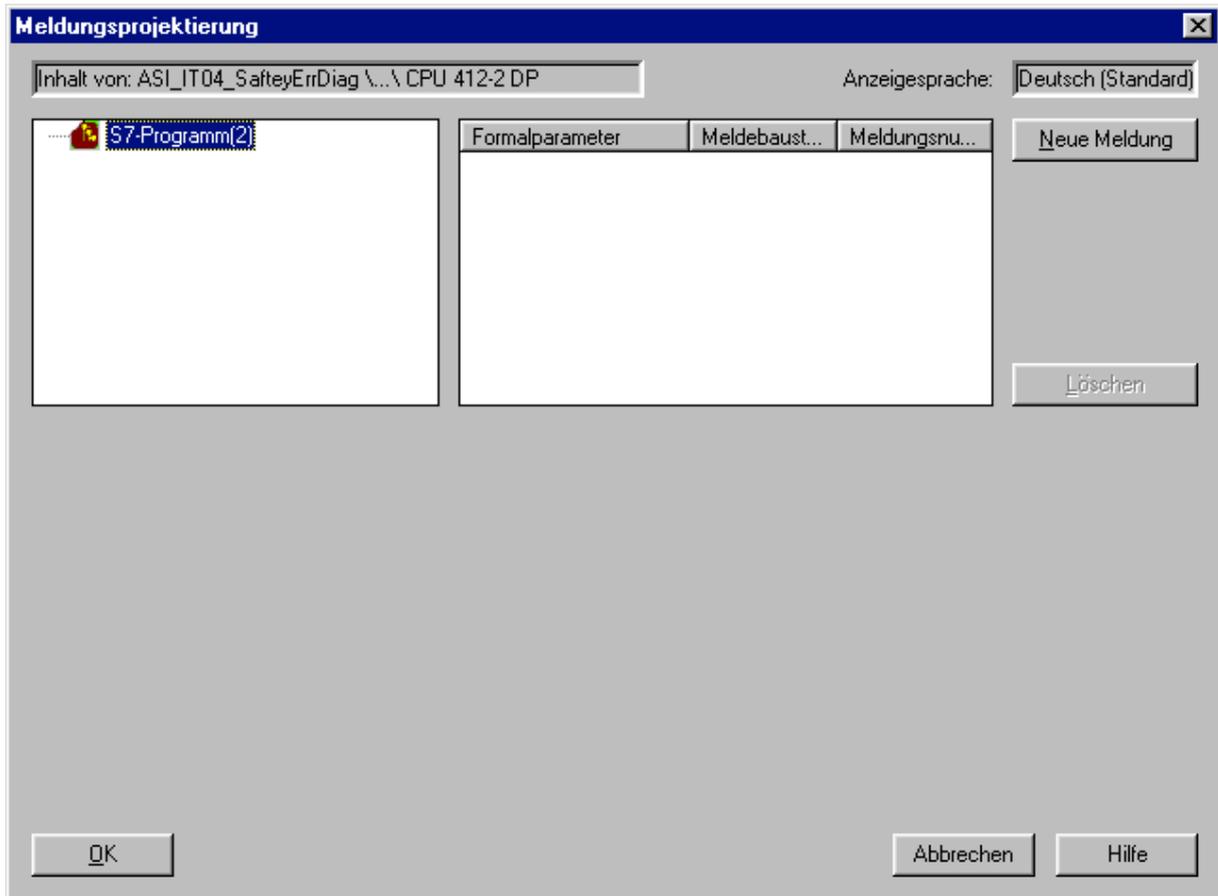


Bild 5 - Meldungsprojektierung

Durch Betätigen der Schaltfläche „Neue Meldung“ wird eine neue anwenderdefinierte Diagnosemeldung mit der Bezeichnung „WR\_USMSG <Nr.>“ eingefügt. Im Register Text, werden die Meldungstexte für das kommende und das gehende Ereignis eingetragen. Im Register Identifikation ist die vom System vorgeschlagene Meldungsnummer und Fehlerklasse bei Wunsch zu ändern und ein Meldungsname einzugeben. In den Meldungstext können auch die beiden, der SFC52 übergebenen, Begleitwerte integriert werden.

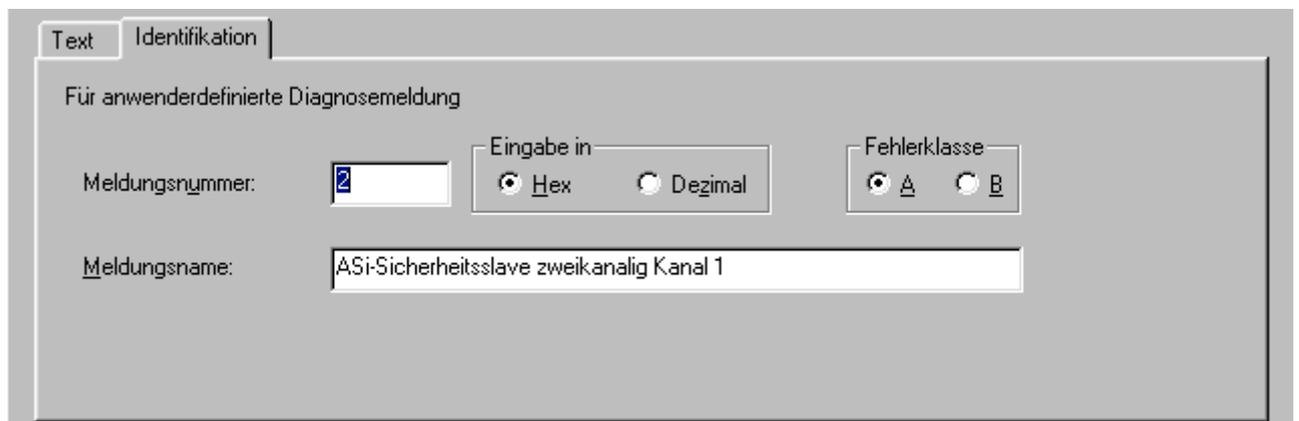
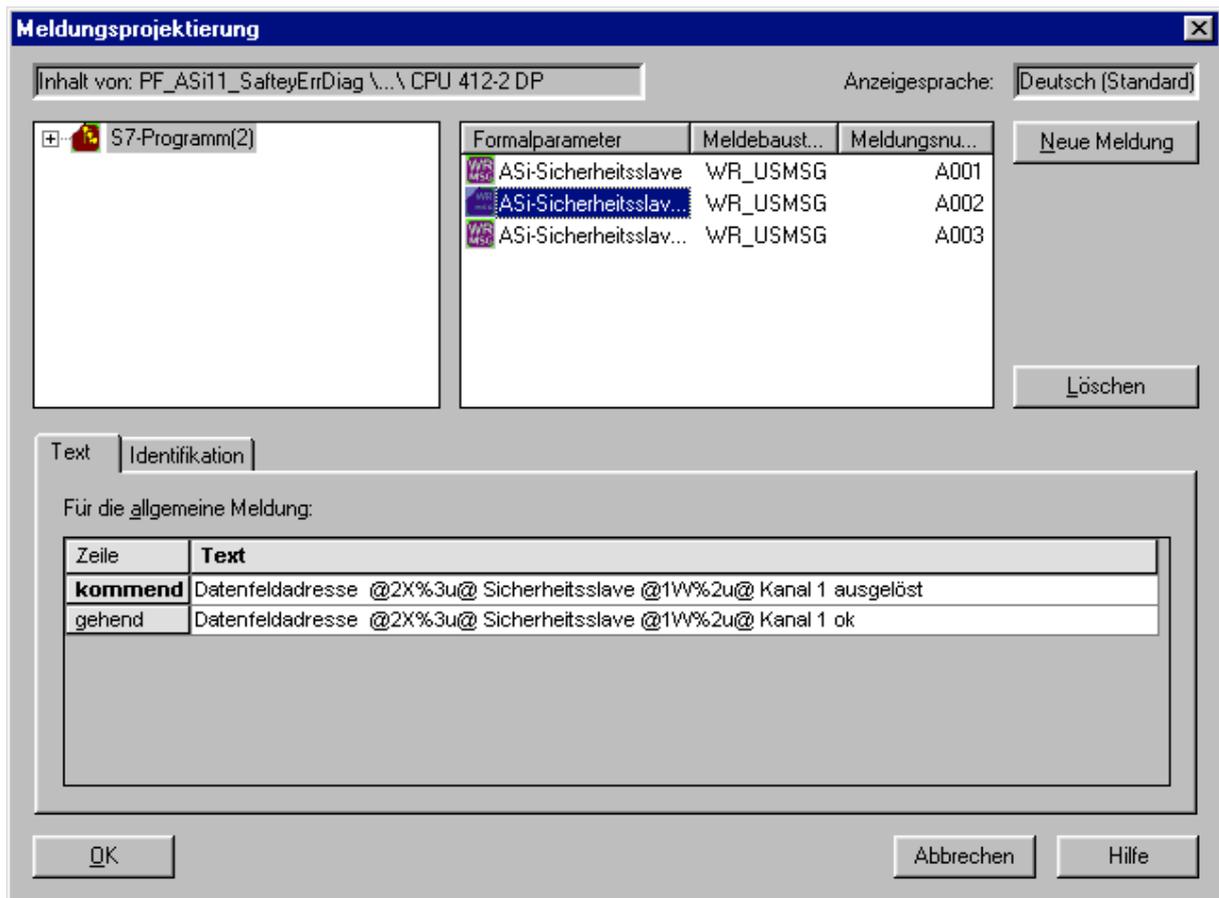


Bild 6 – angelegte Meldung

„@1W%2u@“ steht für den Wert des SFC52-Eingangs IN2, welcher die Adresse des ASi-Sicherheitslave enthält und „@2X%3u@“ für den Wert des SFC52-Eingangs IN3, welcher die Startadresse des Datenfeldes beinhaltet. Nähere Informationen zu dem Einbinden von Begleitinformationen in die anwenderdefinierte Diagnosemeldung finden Sie in den SIMATIC Software Handbüchern oder der SIMATIC Onlinehilfe.

## Anzeigen der anwenderspezifischen Meldungen

In dem Programmbeispiel werden die Meldungen in den Diagnosepuffer eingetragen, hierfür ist lediglich zu beachten, dass in den Einstellungen des Diagnosepuffers die Anzeige der “A: freie anwenderspezifische Ereignisse 1“, bzw. “B: freie anwenderspezifische Ereignisse 2“ aktiviert ist.

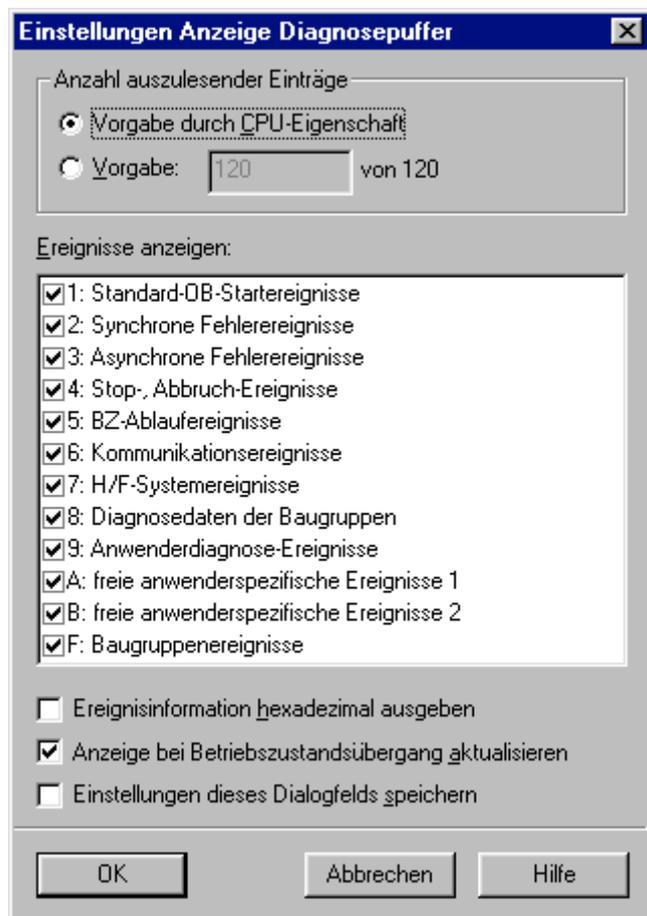


Bild 7 – Anzeigeeinstellungen des Diagnosepuffers

Die Anzahl der möglichen Eintragungen hängt von der Größe des Diagnosepuffers ab.

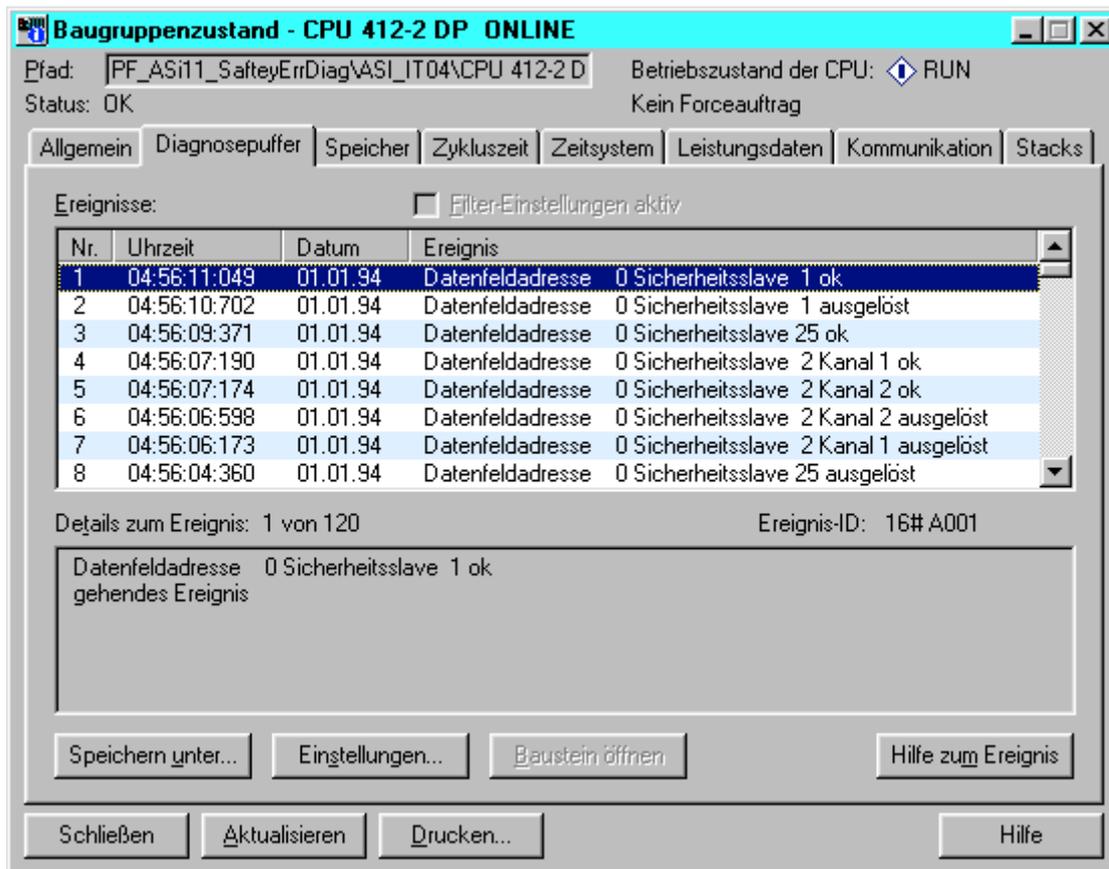


Bild 8 – anwenderdefinierte Diagnosemeldungen in dem Diagnosepuffer

Durch Setzen des Eingangswertes IN0 der SFC52 auf TRUE ist die Eintragung der Meldungen in die Liste der CPU-Meldungen leicht zu erreichen. Bei der zugehörigen Baugruppe müssen Meldungen für “W“ (Diagnoseereignisse) aktiviert sein.

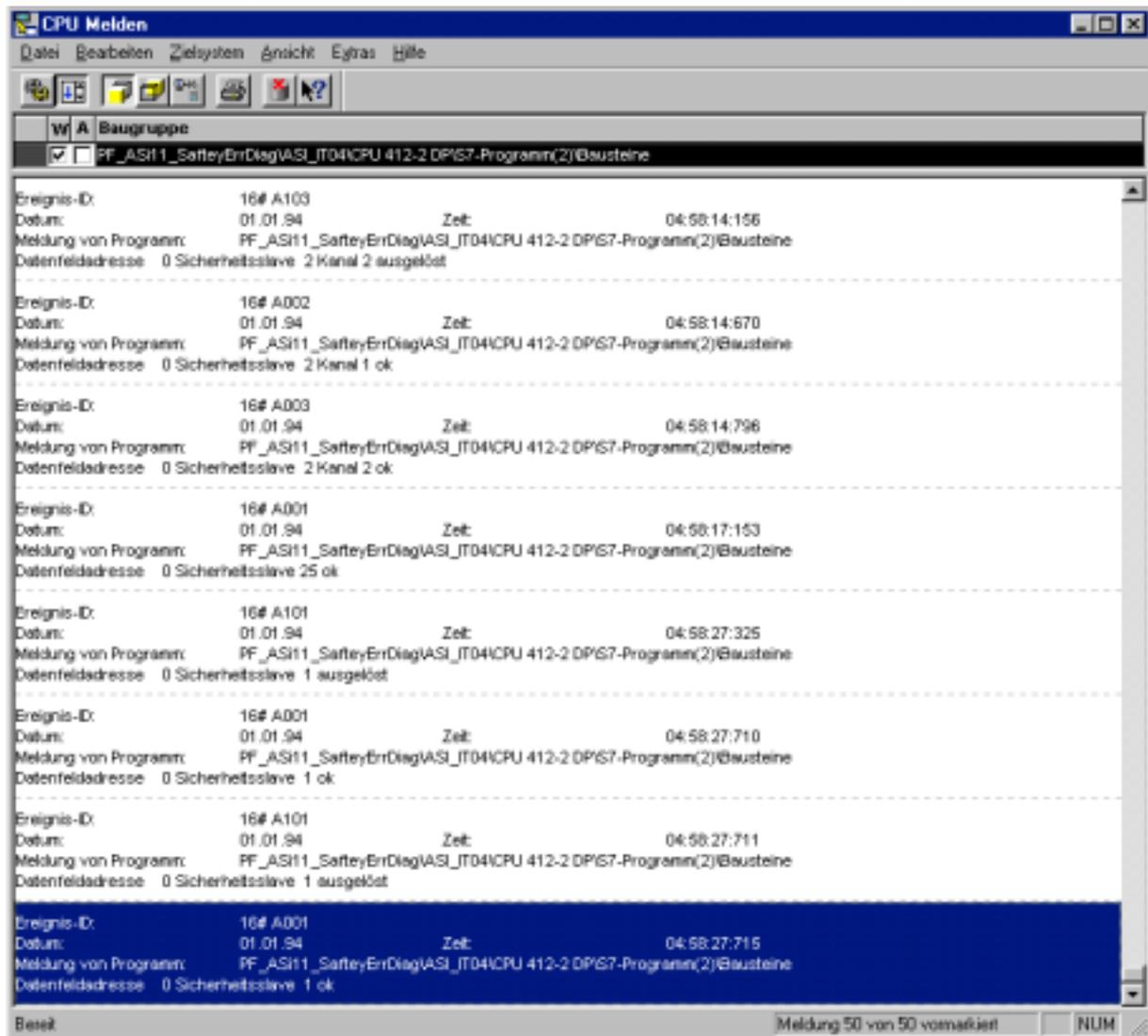


Bild 9 – anwenderdefinierte Diagnosemeldungen in dem Archiv der CPU-Meldungen

Unter „Extras“ → „Einstellungen“ kann die Größe des Archivs und somit die Anzahl der möglichen Eintragungen eingestellt werden.

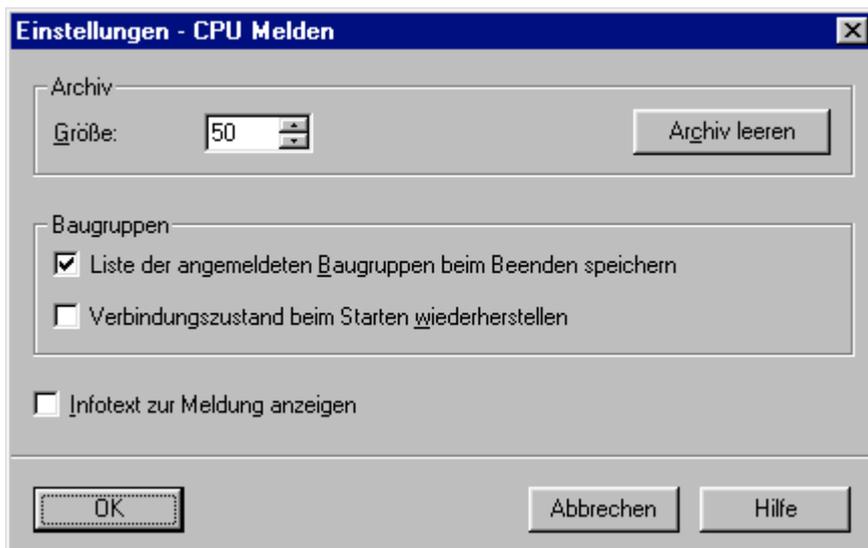


Bild 10 – Einstellungen – CPU Melden

\* y steht für 1 oder 2

\* x steht für g (gerade) bzw. u (ungerade)