



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Programmdokumentation und Funktionsbausteinbeschreibung

Name:

Sebastian Richter
Aaron Zielstorff

Matrikelnummer:

572906
567183

Fachbereich: FB1

Studiengang: M. Elektrotechnik

Fachsemester: 2. FS

Fach: VA2 Hochverfügbare und sichere Systeme

Dozent: Prof. Dr.-Ing. Stephan Schäfer

Abgabe am: 23. September 2022

Inhaltsverzeichnis

1	Hauptprogramm und eigene Funktionsbausteine	4
1.1	Main-File	4
1.2	Funktionsbaustein der Zustände	8
1.3	Blinker-Funktionsbaustein	9
2	Safety-Programm	10
2.1	Grundlagen sichere Programme	10
2.2	Struktur von Sicherheitsprogrammen	10
2.3	Datenaustausch: Anwenderprogramm - Sicherheitsprogramm	12
2.4	Variablen der F-Peripherie-DBs	13
2.5	Main-Safety-File	19
3	Visualisierung	25
	Literaturverzeichnis	26

Abbildungsverzeichnis

1.1	Starten der Anlage	4
1.2	Stoppen der Anlage	5
1.3	Abfrage der Zustandsbedingungen	6
1.4	Beschreibung des Fehlerzustands	7
1.5	Setzen der Rückmeldungen der Schütze	7
1.6	Funktionsbaustein Zustände_DB()	8
1.7	Funktionsbaustein Blinker()	9
2.1	Zykluszeit Sicherheitsprogramm	10
2.2	Aufbau Sicherheitsprogramm	11
2.3	Not-Halt FB	19
2.4	Fehlerleuchtmelder	20
2.5	Motorschütz Förderschnecke	20
2.6	Motorschütz Förderband	21
2.7	Merker Endlage Förderschnecke	21
2.8	Merker Endlage Förderband	21
2.9	Diskrepanzauswertung Schütz Förderschnecke	22
2.10	Diskrepanzauswertung Schütz Förderband	23
2.11	Globale Diskrepanzauswertung	23
2.12	Vereinigung Quittieraufforderungen	24
2.13	FB zum globalen Quittieren	24
3.1	Visualisierung mit SIMATIC HMI	25

Tabellenverzeichnis

2.1	Datenaustausch zwischen Sicherheits- und Standard-Anwenderprogramm	12
2.2	Zugriff auf Prozessabbild der Standardperipherie und F-Peripherie	12
2.3	Variablen der F-Peripherie-DBs	13
2.4	Wiedereingliederung nach Kanalpassivierung	16
2.5	Aufbau von DIAG	18

1 Hauptprogramm und eigene Funktionsbausteine

Das Hauptprogramm stellt den Einstiegspunkt der Software dar. Aus diesem werden sämtliche Funktionen bzw. Funktionsbausteine aufgerufen, welche für das Verhalten der Anlage zuständig sind. Ausnahme ist lediglich das Programm mit den sicherheitsrelevanten Funktionen und Funktionsbausteinen.

Wichtig ist anzumerken, dass in den Bildern gezeigtes Programm für eine Simulation eingesetzt wurde, und nicht mit einer realen Anlage genutzt werden darf. Dafür müssten sämtliche Variablen mit dem Suffix „_HMI“ ersetzt werden durch die jeweilige Variable ohne den Suffix. Weitere Anpassungen für die Nutzung des Programms an einer realen Anlage wurden an den jeweiligen Funktionsbausteinen getätigt.

1.1 Main-File

Das erste Netzwerk (Abbildung 1.1) setzt die Funktion des START-Drucktasters (S1) um. Über einen rücksetzdominanten FlipFlop wird die Betätigung des Tasters gespeichert, bis die Rücksetzbedingung (NOT S0) erfüllt ist. Ausgang des Netzwerkes ist eine Merkervariable (xM_S1) mit der booleschen Information, dass **START** vom Nutzer gedrückt wurde.

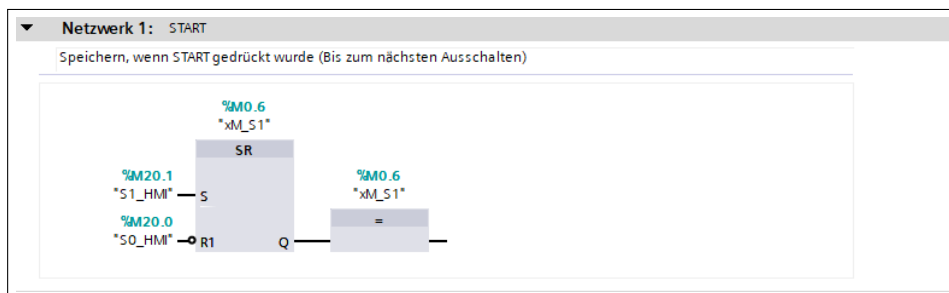


Abb. 1.1: Starten der Anlage

Im zweiten Netzwerk (Abbildung 1.2) wird der 5-Sekunden-Nachlauf des Förderbandes nach der Förderschnecke beim Ausschalten programmiert. Durch die Abfrage des Zustands des STOP-Leuchtdrucktasters (S0) wird das Schütz der Förderschnecke (K3) ausgeschaltet. Der STOP-Taster wurde als Öffner vorgesehen, folglich wird auf die negative Flanke getriggert. Sobald das Schütz (K3) ausgeschaltet ist, wird die Ausschaltverzögerung (TOF) gestartet. Nach Ablauf der 5-Sekunden wird das Schütz des Förderbandes (K4) ebenfalls ausgeschaltet.

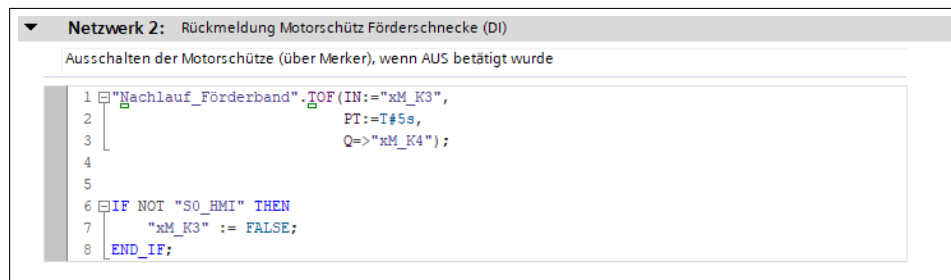


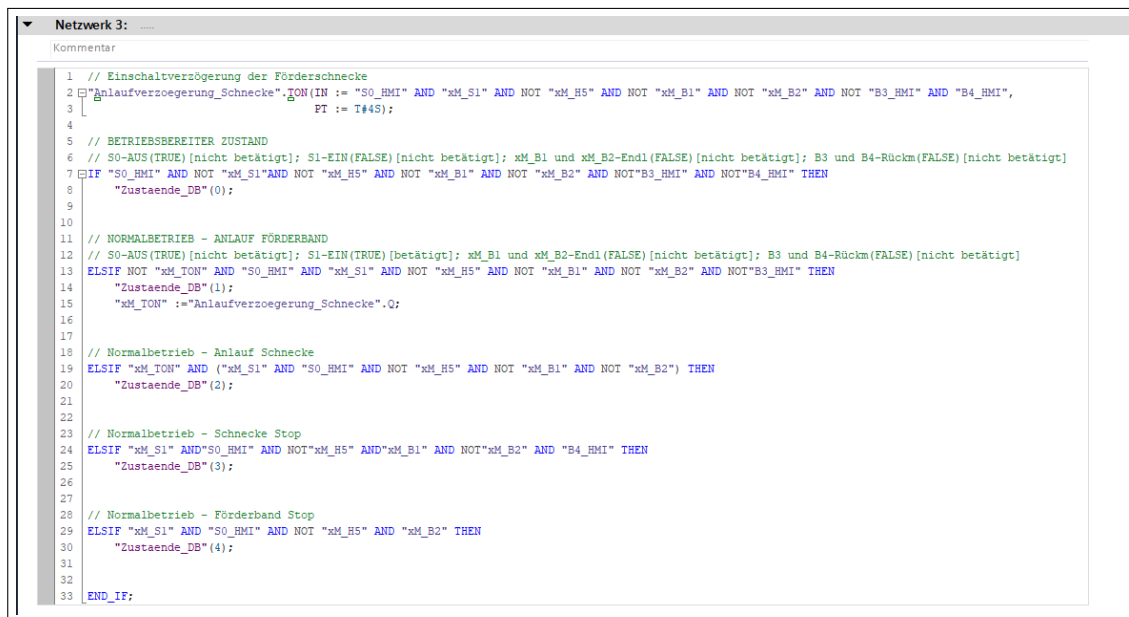
Abb. 1.2: Stoppen der Anlage

Im dritten Netzwerk (Abbildung 1.3) wurden Abfragen zu den Bedingungen für die Zustände der Anlage implementiert. Jede *IF* bzw. *ELSIF*-Bedingung fragt die Eintrittsvoraussetzungen eines Zustands ab. Diese können aus den Zustandsgraphen abgelesen werden. Sind die Bedingungen für einen Zustand erfüllt, wird der Funktionsbaustein Zustände_DB() mit einer Nummer aufgerufen. Die Nummer gibt an, um welchen Zustand es sich handelt. Folgende Zustände wurden umgesetzt:

- 0 - Betriebsbereiter Zustand
- 1 - Anlauf Förderband
- 2 - Anlauf Schnecke
- 3 - Schnecke Stop
- 4 - Förderband Stop

Der Nachlauf des Förderbandes wurde in keinem separaten Zustand umgesetzt, sondern ist in Netzwerk 2 (Abbildung 1.2) beim Stoppen der Anlage mit implementiert.

Der Anlauf der Schnecke erfolgt ebenfalls zeitverzögert (TON) und ist von Zeile 1 bis 3 im Netzwerk 3 (Abbildung 1.3) umgesetzt. Der Eingang der Einschaltverzögerung beinhaltet die gleichen Bedingungen wie der Zustand „Anlauf Förderband“.



```

Netzwerk 3: .....
Kommentar
1 // Einschaltverzögerung der Förderschnecke
2 "Anlaufverzögerung_Schnecke".TON(IN := "S0_RMI" AND "xM_S1" AND NOT "xM_H5" AND NOT "xM_B1" AND NOT "xM_B2" AND NOT "B3_RMI" AND "B4_RMI",
3   PT := T#4S);
4
5 // BETRIEBSBEREITER ZUSTAND
6 // S0-AUS(TRUE) [nicht betätigt]; S1-EIN(FALSE) [nicht betätigt]; xM_B1 und xM_B2-End1(FALSE) [nicht betätigt]; B3 und B4-Rückm(FALSE) [nicht betätigt]
7 IF "S0_RMI" AND NOT "xM_S1" AND NOT "xM_H5" AND NOT "xM_B1" AND NOT "xM_B2" AND NOT "B3_RMI" AND NOT "B4_RMI" THEN
8   "Zustaeende_DB"(0);
9
10
11 // NORMALBETRIEB - ANLAUF FÖRDERBAND
12 // S0-AUS(TRUE) [nicht betätigt]; S1-EIN(TRUE) [betätigt]; xM_B1 und xM_B2-End1(FALSE) [nicht betätigt]; B3 und B4-Rückm(FALSE) [nicht betätigt]
13 ELIF NOT "xM_TON" AND "S0_RMI" AND "xM_S1" AND NOT "xM_H5" AND NOT "xM_B1" AND NOT "xM_B2" AND NOT "B3_RMI" THEN
14   "Zustaeende_DB"(1);
15   "xM_TON" := "Anlaufverzögerung_Schnecke".Q;
16
17
18 // Normalbetrieb - Anlauf Schnecke
19 ELIF "xM_TON" AND ("xM_S1" AND "S0_RMI" AND NOT "xM_H5" AND NOT "xM_B1" AND NOT "xM_B2") THEN
20   "Zustaeende_DB"(2);
21
22
23 // Normalbetrieb - Schnecke Stop
24 ELIF "xM_S1" AND "S0_RMI" AND NOT "xM_H5" AND "xM_B1" AND NOT "xM_B2" AND "B4_RMI" THEN
25   "Zustaeende_DB"(3);
26
27
28 // Normalbetrieb - Förderband Stop
29 ELIF "xM_S1" AND "S0_RMI" AND NOT "xM_H5" AND "xM_B2" THEN
30   "Zustaeende_DB"(4);
31
32
33 END_IF;

```

Abb. 1.3: Abfrage der Zustandsbedingungen

Das vierte Netzwerk (Abbildung 1.4) umfasst das Verhalten der Anlage im Fehlerfall. Dabei wird in drei Phasen unterschieden:

- Fehler wird detektiert
- Fehler wird behoben
- Fehler wird quittiert

Im der ersten Phase blinkt der Leuchtmelder (H5) mit einer Frequenz von $f = 1\text{Hz}$. Dazu wird der Funktionsbaustein `Blinktakt()` aufgerufen. Sobald der Fehler behoben wurde, wird dies durch das dauerhafte Leuchten von H5 signalisiert. Gleichzeitig erfolgt das Blinken des QUITTIER-Leuchtdrucktaster (H2) mit selbiger Frequenz des Leuchtmelders H5 aus Phase 1. Sobald der QUITTIER-Leuchtdrucktaster (S2) gedrückt wurde, ist die Phase zwei beendet und die Anlage wird in Phase drei wieder in den betriebsbereiten Zustand versetzt.

```

Netzwerk 4: Fehlerbehandlung
Fehlerzustand

1 // Wenn Fehler Anliegt (noch nicht beseitigt)
2 IF "xM_H5" AND NOT "xM_ACK_REQ" THEN
3   "Blinktakt_H5"(t_ON := T#500ms,
4                 t_OFF := T#500ms,
5                 xStart := TRUE,
6                 xM_Blink => "H5_HMI");
7
8
9 // Wenn Fehler behoben wurde und quittiert werden muss
10 ELSEIF "xM_H5" AND "xM_ACK_REQ" THEN
11   "Blinktakt_H2"(t_ON:=T#500ms,
12                 t_OFF:=T#500ms,
13                 xStart:=TRUE,
14                 xM_Blink=>"H2_HMI");
15   "H5_HMI" := TRUE;
16   "xM_Quittieren" := TRUE;
17
18
19 END_IF;
20
21 // Wenn Fehler quittiert wurde wieder zurück zu Betriebsbereit
22 IF "xM_Quittieren" AND "S2_HMI" THEN
23   "xM_K3" := "xM_K4" := "xM_S1" := "H5_HMI" := FALSE;
24
25
26 END_IF;

```

Abb. 1.4: Beschreibung des Fehlerzustands

Das Netzwerk Fünf (Abbildung 1.5) ist lediglich im simulierten Anlagenzustand anzuwenden und ermöglicht die Rückmeldung der geschalteten Schütze (K3 und K4) durch die Variablen (B3 und B4). Im realen Anlagenbetrieb werden die Rückmeldungen automatisch durchgeführt, folglich ist dieses Netzwerk zu entfernen.

```

Netzwerk 5: Automatisches Schalten der Schütz-Rückmeldungen im Simulationsbetrieb
Unbedingt ENTFERNEN, wenn auf realer Anlage ausgeführt!

1 IF "xM_K3" THEN
2   "B3_HMI" := TRUE;
3 ELSE
4   "B3_HMI" := FALSE;
5 END_IF;
6
7 IF "xM_K4" THEN
8   "B4_HMI" := TRUE;
9 ELSE
10  "B4_HMI" := FALSE;
11 END_IF;

```

Abb. 1.5: Setzen der Rückmeldungen der Schütze

1.2 Funktionsbaustein der Zustände

Der Funktionsbaustein Zustände_DB() (Abbildung 1.6) umfasst die Implementierung der Zustände gemäß der Zustandsgraphen. Im Main-File werden entsprechend Integer-Werte von Null bis Vier vergeben. Anhand dieser Werte wird durch eine CASE-Anweisung der richtige Zustand ermittelt und die jeweiligen notwendigen Variablen auf TRUE oder FALSE gesetzt. Sofern die Anlage keinen dieser Zustände zugeordnet werden kann, wird über die ELSE-Abfrage eine Textnachricht im Terminal der SPS zurückgegeben. Die Anlage wird durch die Safety-Baugruppen ausgeschaltet und in einen sicheren Zustand versetzt.

```

1 CASE #iZustand OF
2   0:
3     #sZustand := 'Betriebsbereit';
4
5     "H0_HMI" := "H2_HMI" := "xM_K3" := "xM_K4" := "xM_TON" := FALSE; // H5 FALSE
6     // "xM_Zustand_2" := FALSE;
7     // Blinken des START-Tasters
8     #Blinktakt(t_ON := T#500ms,
9               t_OFF := T#500ms,
10              xStart := TRUE,
11              xM_Blink => "H1_HMI");
12
13
14   1:
15     #sZustand := 'Anlauf Foerderband';
16
17     "H2_HMI" := "xM_K3" := "xM_TON" := FALSE; // H5 FALSE
18     "H0_HMI" := "xM_K4" := "H1_HMI" := TRUE;
19
20
21   2:
22     #sZustand := 'Anlauf Schnecke';
23
24     "H2_HMI" := FALSE; // H5 FALSE
25     "H0_HMI" := "H1_HMI" := "xM_K3" := "xM_K4" := TRUE;
26
27
28   3:
29     #sZustand := 'Schnecke Stop';
30
31     // "xM_Zustand_2" := FALSE;
32     "H2_HMI" := "xM_K3" := FALSE; // H5 FALSE
33     "H0_HMI" := "H1_HMI" := "xM_K4" := TRUE;
34
35
36   4:
37     #sZustand := 'Foerderband Stop';
38
39     // "xM_Zustand_2" := FALSE;
40     "H2_HMI" := "xM_K3" := "xM_K4" := "xM_TON" := FALSE; // H5 FALSE
41     "H0_HMI" := "H1_HMI" := TRUE;
42
43
44   ELSE
45     #sZustand := 'Undefiniert';
46
47     "H1_HMI" := "H2_HMI" := "xM_K3" := "xM_K4" := FALSE; // H5 FALSE
48
49 END_CASE;

```

Abb. 1.6: Funktionsbaustein Zustände_DB()

1.3 Blinker-Funktionsbaustein

Zuletzt wurde ein Funktionsbaustein für eine Blinker-Funktionalität umgesetzt (Abbildung 1.7). Über diesen ist es möglich per Aufruf von `Blinktakt()` ein Blinksignal zu generieren. Dem Funktionsbaustein kann eine Ein-Zeit (`t_ON`) und eine Aus-Zeit (`t_OFF`) mitgegeben werden, um den Blinktakt zu setzen. Über `xStart` wird der Blinker aktiviert. Die Ausgangsvariable (`xM_Blink`) liefert das generierte Blinksignal.

Eingesetzt wird der Funktionsbaustein für den **START-Leuchtdrucktaster**, den **Fehler-leuchtmelder** und den **QUITTIER-Leuchtdrucktaster**.

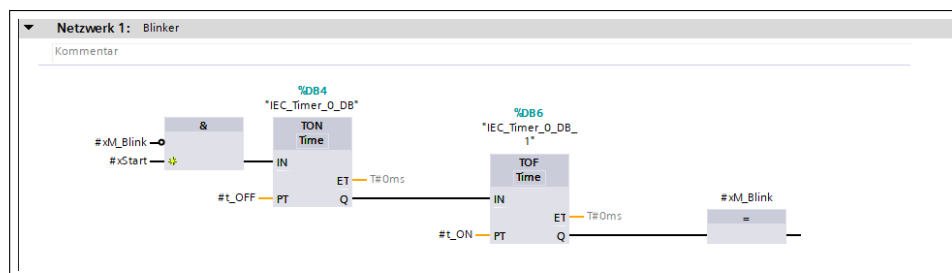


Abb. 1.7: Funktionsbaustein Blinker()

2 Safety-Programm

2.1 Grundlagen sichere Programme

Das Sicherheitsprogramm wird parallel zum Hauptprogramm auf einer sogenannten F-CPU (fehlersichere CPU) ausgeführt. Es besitzt meist eine kürzere Zykluszeit und kann über Interrupts in das Hauptprogramm eingreifen, falls dies erforderlich ist (siehe Abbildung 2.1).

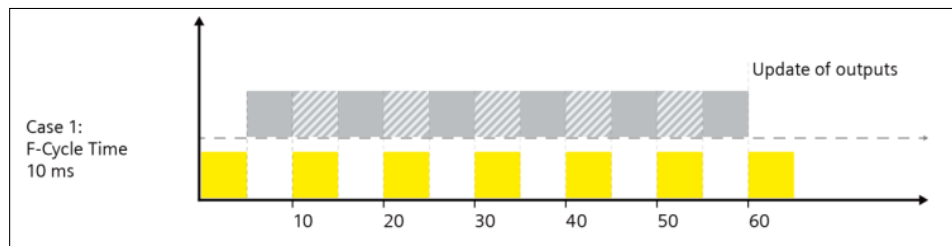


Abb. 2.1: Einfluss der Zykluszeit des Sicherheitsprogramms auf das Standard- Anwenderprogramm

2.2 Struktur von Sicherheitsprogrammen

Ein Sicherheitsprogramm besteht zur Strukturierung aus einer oder zwei F-Ablaufgruppen. Jede F-Ablaufgruppe enthält:

- F-Bausteine, die von Ihnen mit FUP oder KOP erstellt werden oder aus der Projektbibliothek oder globalen Bibliotheken eingefügt werden
- F-Bausteine, die automatisch ergänzt werden (F-Systembausteine F-SBs, automatisch generierte F-Bausteine, F-Ablaufgruppeninfo-DB und F-Peripherie-DBs)

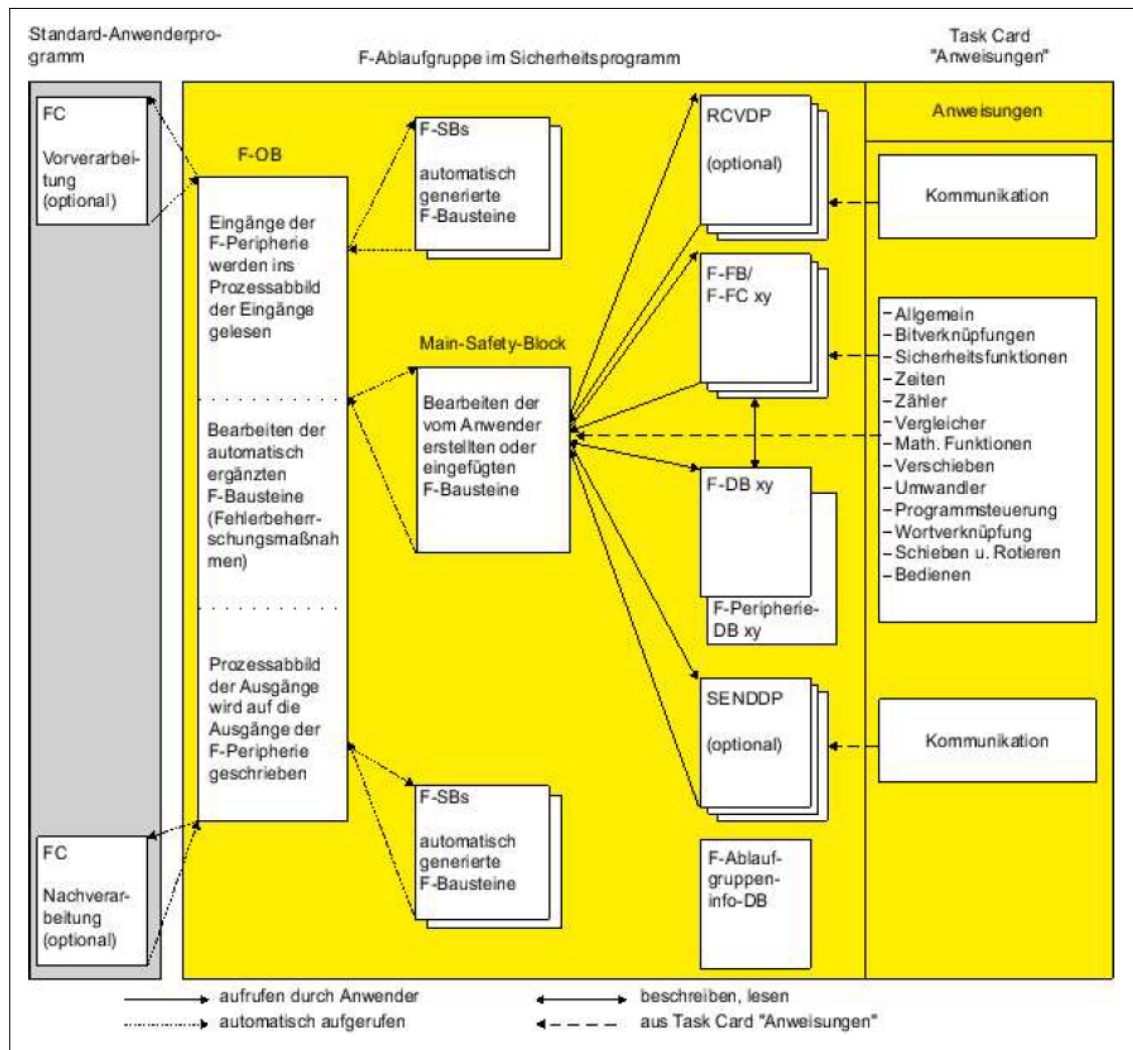


Abb. 2.2: Schematischen Aufbau eines Sicherheitsprogramms bzw. einer F-Ablaufgruppe für eine F-CPU S7-1200/1500

2.3 Datenaustausch: Anwenderprogramm - Sicherheitsprogramm

Es besteht die Möglichkeit, Daten zwischen dem Sicherheits- und Standard- Anwenderprogramm auszutauschen. Dazu können Variablen aus DBs, F-DBs sowie Merker verwendet werden:

	Vom Standard-Anwenderprogramm aus		Vom Sicherheitsprogramm aus	
	lesend	schreibend	lesend	schreibend
Variable aus DB	zulässig	zulässig	entweder lesend <u>oder</u> schreibend auf eine Variable aus dem DB	
Variable aus F-DB	zulässig	nicht zulässig	zulässig	zulässig
Merker	zulässig	zulässig	entweder lesend <u>oder</u> schreibend auf einen Merker	

Tab. 2.1: Datenaustausch zwischen Sicherheits- und Standard-Anwenderprogramm

Außerdem besteht die Möglichkeit, auf das Prozessabbild der Standard- und F-Peripherie zuzugreifen:

		Vom Standard-Anwenderprogramm aus		Vom Sicherheitsprogramm aus	
		lesend	schreibend	lesend	schreibend
Prozessabbild Standardperipherie	PAE	zulässig	zulässig	zulässig	nicht zulässig
	PAA	zulässig	zulässig	nicht zulässig	zulässig
Prozessabbild F-Peripherie	PAE	zulässig	nicht zulässig	zulässig	nicht zulässig
	PAA	zulässig	nicht zulässig	nicht zulässig	zulässig

Tab. 2.2: Zugriff auf Prozessabbild der Standardperipherie und F-Peripherie

Zur Entkopplung des Anwender- vom Sicherheitsprogramm wird empfohlen, für den Datenaustausch Übergabe-Datenbausteine zu definieren.

2.4 Variablen der F-Peripherie-DBs

	Variable	Datentyp	Funktion	Startwert
Variablen, die Sie beschreiben können/müssen	PASS_ON	BOOL	1 = Passivierung aktivieren	0
	ACK_NEC	BOOL	1 = Quittierung für Wiedereingliederung erforderlich bei F-Peripherie-/Kanalfehlern	1
	ACK_REI	BOOL	1 = Quittierung für Wiedereingliederung	0
	IPAR_EN	BOOL	Variable für Umparametrierung fehlersicherer DP-Normslaves/IO-Normdevices bzw. bei SM 336; F-AI 6x0/4 ... 20 mA HART zur Freigabe der HART-Kommunikation	0
	DISABLE*	BOOL	1 = F-Peripherie deaktivieren	0
Variablen, die Sie auswerten können	PASS_OUT	BOOL	Passivierungsausgang	1
	QBAD	BOOL	1 = Ersatzwerte werden ausgegeben	1
	ACK_REQ	BOOL	1 = Quittierungsanforderung für Wiedereingliederung	0
	IPAR_OK	BOOL	Variable für Umparametrierung fehlersicherer DP-Normslaves/IO-Normdevices bzw. bei SM 336; F-AI 6x0/4 ... 20 mA HART zur Freigabe der HART-Kommunikation	0
	DIAG	BYTE	Nicht fehlersichere Serviceinformation	0
	DISABLED*	BOOL	1 = F-Peripherie ist deaktiviert	0
	QBAD_I_xx	BOOL	1 = Ersatzwerte werden ausgegeben auf Eingangskanal xx (S7-300/400)	1
	QBAD_O_xx	BOOL	1 = Ersatzwerte werden ausgegeben auf Ausgangskanal xx (S7-300/400)	1

* ab Safety-System-Version V2.1 für S7-1200/1500

Tab. 2.3: Variablen der F-Peripherie-DBs

PASS_ON:

Mit der Variable PASS_ON können Sie eine Passivierung einer F-Peripherie, z. B. abhängig von bestimmten Zuständen in Ihrem Sicherheitsprogramm, aktivieren.

Sie können über die Variable PASS_ON im F-Peripherie-DB nur die gesamte F-Peripherie passivieren, kanalgranulare Passivierung ist nicht möglich.

Solange PASS_ON = 1 ist, erfolgt eine **Passivierung** der zugehörigen F-Peripherie.

ACK_NEC:

Wenn von der F-Peripherie ein F-Peripheriefehler erkannt wird, erfolgt eine **Passivierung** der betroffenen F-Peripherie. Wenn Kanalfehler erkannt werden, erfolgt bei projektierter kanalgranularer Passivierung eine Passivierung der betroffenen Kanäle, bei Passivierung der gesamten F-Peripherie eine Passivierung aller Kanäle der betroffenen F-Peripherie. Nach Behebung des F-Peripherie-/Kanalfehlers erfolgt die **Wiedereingliederung** der betroffenen F-Peripherie abhängig von ACK_NEC:

- Mit ACK_NEC = 0 können Sie eine **automatische Wiedereingliederung** parametrieren.
- Mit ACK_NEC = 1 können Sie eine **Wiedereingliederung** durch eine **Anwenderquittierung** parametrieren.

ACK_REI:

Wenn vom F-System für eine F-Peripherie ein Kommunikationsfehler oder ein F-Peripheriefehler erkannt wird, erfolgt eine Passivierung der betroffenen F-Peripherie. Wenn Kanalfehler erkannt werden, erfolgt bei projektierter kanalgranularer Passivierung eine Passivierung der betroffenen Kanäle, bei Passivierung der gesamten F-Peripherie eine Passivierung aller Kanäle der betroffenen F-Peripherie. Für eine **Wiedereingliederung** der F-Peripherie/Kanäle der F-Peripherie nach Behebung der Fehler ist eine **Anwenderquittierung** mit positiver Flanke an der Variablen ACK_REI des F-Peripherie-DBs erforderlich:

- nach Kommunikationsfehlern immer
- nach F-Peripherie-/Kanalfehlern nur bei Parametrierung "Kanalfehler Quittierung = Manuell" bzw. ACK_NEC = 1

Bei einer Wiedereingliederung nach Kanalfehlern werden alle Kanäle, deren Fehler beseitigt wurden, wiedereingegliedert.

Eine Quittierung ist erst möglich, wenn die Variable ACK_REQ = 1 ist.

IPAR_EN:

Die Variable IPAR_EN entspricht der Variablen iPAR_EN_C im Busprofil PROFIsafe, ab PROFIsafe Specification V1.20.

Fehlersichere DP-Normslaves/IO-Normdevices

Wann Sie diese Variable bei einer Umparametrierung von fehlersicheren DP-Normslaves/IO-Normdevices setzen/rücksetzen müssen, entnehmen Sie der PROFIsafe Specification ab V1.20 bzw. der Dokumentation zum fehlersicheren DP-Normslave/IO-Normdevice.

Beachten Sie, dass durch IPAR_EN = 1 keine Passivierung der betroffenen F-Peripherie

ausgelöst wird.

Soll bei $IPAR_EN = 1$ passiviert werden, müssen Sie die Variable $PASS_ON = 1$ setzen.

DISABLE:

Mit der Variable DISABLE können Sie eine F-Peripherie deaktivieren.

Solange $DISABLE = 1$ ist, erfolgt eine **Passivierung** der zugehörigen F-Peripherie.

In den Diagnosepuffer der F-CPU werden zu dieser F-Peripherie keine Diagnoseeinträge des Sicherheitsprogramms (z. B. wegen Kommunikationsfehler) mehr eingetragen.

Bereits vorhandene Diagnoseeinträge werden als gehend gekennzeichnet.

QBAD:

Bei einer Passivierung geht die F-Peripherie in den fehlersicheren Zustand. Nach Fehlerbehebung kann diese wieder eingegliedert werden.

Für die Wiedereingliederung existieren verschiedene Möglichkeiten. Dabei ist zu berücksichtigen, ob die F-Peripherie das PROFIsafe-Profil RIOforFA-Safety unterstützt.

Die Dezentrale Peripherie ET 200SP, hier: F-DI8x24 V DC verfügt über KEIN RIOforFA-Safety Profil.

Kommt es an einer F-Peripherie zu einem Fehler (z. B. zu einem Kanalfehler) geht die F-Peripherie (bzw. der betroffene Kanal) in den sicheren Zustand. In diesem Zustand der „Passivierung“ werden statt der Prozesswerte automatisch Ersatzwerte ausgegeben.

Nach Beseitigung des Fehlers, der zur Passivierung führte, kann die Umschaltung von Ersatzwerte auf Prozesswerte erfolgen. Die Umschaltung kann automatisch oder nach einer Anwenderquittierung im Sicherheitsprogramm erfolgen.

Begriffe für die Umschaltung sind „Wiedereingliederung“ oder auch „Reintegration“.

Ersatzwertausgabe nach:	F-Peripherie ohne Profil "RIOforFA-Safety" mit FCPUs S7-1200/1500
Anlauf des F-Systems	QBAD und PASS_OUT = 1, DISABLED unverändert , für alle Kanäle gilt: - Kanalwert = Ersatzwert (0) - Wertstatus = 0*
Kommunikationsfehlern	
F-Peripheriefehlern	
Kanalfehlern bei Projektierung Passivierung der gesamten F-Peripherie	
Kanalfehlern bei Projektierung kanalgranulare Passivierung	QBAD und PASS_OUT = 1, DISABLED unverändert , für betroffene Kanäle gilt: - Kanalwert = Ersatzwert (0) - Wertstatus = 0*
solange im F-Peripherie-DB mit PASS_ON = 1 eine Passivierung der F-Peripherie aktiviert ist	QBAD = 1, PASS_OUT und DISABLED unverändert , Für alle Kanäle gilt: - Kanalwert = Ersatzwert (0) - Wertstatus = 0*
solange im F-Peripherie-DB mit DISABLE = 1 die F-Peripherie deaktiviert ist	QBAD, PASS_OUT und DISABLED = 1, für alle Kanäle gilt: - Kanalwert = Ersatzwert (0) - Wertstatus = 0*

Tab. 2.4: Wiedereingliederung nach Kanalpassivierung

ACK_REQ:

Wenn vom F-System für eine F-Peripherie ein Kommunikationsfehler oder ein F-Peripherie-/Kanalfehler erkannt wird, erfolgt eine Passivierung der betroffenen F-Peripherie bzw. einzelner Kanäle der F-Peripherie.

Durch ACK_REQ = 1 wird signalisiert, dass für eine Wiedereingliederung der betroffenen F-Peripherie/der Kanäle der F-Peripherie eine **Anwenderquittierung** erforderlich ist.

Das F-System setzt ACK_REQ = 1, sobald der Fehler behoben ist und eine Anwenderquittierung möglich ist. Bei kanalgranularer Passivierung setzt das F-System ACK_REQ = 1, sobald ein Kanalfehler behoben ist. Für diesen Fehler ist eine Anwenderquittierung möglich. Nach erfolgter Quittierung wird ACK_REQ vom F-System auf 0 zurückgesetzt.

IPAR_OK:

Die Variable IPAR_OK entspricht der Variablen iPar_OK_S im Busprofil PROFIsafe, ab PROFIsafe Specification V1.20.

Fehlersichere DP-Normslaves/IO-Normdevices

Wie Sie diese Variable bei einer Umparametrierung von fehlersicheren DP-Normslaves/IO-Normdevices auswerten können, entnehmen Sie der PROFIsafe Specification ab V1.20 bzw. der Dokumentation zum fehlersicheren DP-Normslave/IO-Normdevice

DIAG:

Über die Variable DIAG wird eine nicht fehlersichere Information (1 Byte) über aufgetretene Fehler für Servicezwecke zur Verfügung gestellt.

Sie können diese über Bedien- und Beobachtungssysteme auslesen oder ggf. in Ihrem Standard-Anwenderprogramm auswerten. Die DIAG-Bits bleiben gespeichert, bis Sie an der Variablen ACK_REI eine Quittierung durchführen oder bis eine automatische Wiedereingliederung erfolgt.

Bit Nr.	Belegung	Mögliche Fehlerursachen	Abhilfemaßnahmen
Bit 0	Timeout von F-Peripherie erkannt	Die PROFIBUS/PROFINET-Verbindung zwischen F-CPU und F-Peripherie ist gestört. Der Wert für die F-Überwachungszeit der F-Peripherie ist zu gering eingestellt die F-Peripherie erhält ungültige Parametrierungsdaten oder	<ul style="list-style-type: none"> - Überprüfung Sie die PROFIBUS/PROFINET-Verbindung und stellen Sie sicher, dass keine externen Störquellen vorhanden sind. - Überprüfen Sie die Parametrierung der F-Peripherie. Stellen Sie ggf. einen höheren Wert für die Überwachungszeit ein. Übersetzen Sie die Hardware-Konfiguration erneut und laden Sie diese in die F-CPU. Übersetzen Sie das Sicherheitsprogramm erneut. - Überprüfen Sie den Diagnosepuffer der F-Peripherie. - Schalten Sie die Spannung der F-Peripherie aus und wieder ein.
		interne Fehler der F-Peripherie oder	F-Peripherie tauschen
		interne Fehler der F-CPU	F-CPU tauschen
Bit 1	F-Peripherie-/Kanalfehler von F-Peripherie erkannt ¹	siehe Handbücher zur F-Peripherie	siehe Handbücher der F-Peripherie
Bit 2	CRC-/Sequenznummernfehler von F-Peripherie erkannt	siehe Beschreibung für Bit 0	siehe Beschreibung für Bit 0
Bit 3	Reserve	-	-
Bit 4	Timeout von F-System erkannt	siehe Beschreibung für Bit 0	siehe Beschreibung für Bit 0
Bit 5	Sequenznummernfehler von F-System erkannt ²	siehe Beschreibung für Bit 0	siehe Beschreibung für Bit 0
Bit 6	CRC-Fehler von F-System erkannt	siehe Beschreibung für Bit 0	siehe Beschreibung für Bit 0
Bit 7	Adressierungsfehler ³	-	Wenden Sie sich an Service & Support

¹ Nicht bei F-Peripherie, die das Profil „RIOforFA-Safety“ unterstützt

² nur bei F-CPU S7-300/400

³ nur bei F-CPU S7-1200/1500

Tab. 2.5: Aufbau von DIAG

2.5 Main-Safety-File

Netzwerk 1 (Abbildung 2.3) zeigt den Funktionsbaustein für die NOT-HALT Funktionalität (Engl. E-Stop). Der Baustein besitzt für uns drei relevante Eingänge. Dem Eingang **E_STOP** wird die dem Not-Halt zugehörige Variable (S5) zugeordnet. Grundsätzlich besitzt der Not-Halt zwei Adressen, da er zweikanalig ist. Der Öffnerkontakt wird jedoch der Variable S5 zugeordnet. Somit ist das Signal am E_STOP-Eingang im Normalfall TRUE und im ausgelösten Zustand FALSE. Der zweite wichtige Eingang ist **ACK_REQ**. Hier sollte TRUE eingesetzt werden, damit nach Auslösung eines Not-Halts der Fehler erst quittiert werden muss, damit die Anlage wieder in den Normalbetrieb übergeht. Dem letzten relevanten Eingang **ACK** wird die Variable des Quittiert-Drucktasters (S2) zugeordnet. Nun kann ein Not-Halt-Fehler quittiert werden.

In unserem Beispiel sind zwei Ausgänge des E_STOP Funktionsbausteins von Bedeutung. Zum einen **Q**, welches zurück gibt, ob sich die Anlage im Not-Halt befindet. TRUE würde bedeuten, dass die Anlage Normal operiert und FALSE, dass ein Not-Halt vorliegt. Dem Ausgang ist die Merkervariable *xM_E_Stop* zugewiesen, über welche im Netzwerk 2 (siehe Abbildung 2.4) die Merkervariable für den Fehlerleuchtmelder geschaltet wird. Über **ACK_REQ** gibt der Baustein zurück, ob ein Quittieren erforderlich ist. Dies ist genau dann der Fall, wenn ein Not-Halt ausgelöst wurde und die Anlage wieder in den Normalbetrieb überführt werden kann. Auch hier wird eine Merkervariable verwendet (*xM_Ack_Req1*), die im Netzwerk 10 (Abbildung 2.12) mit den anderen Merkervariablen für *Ack_Req* verodert wird.

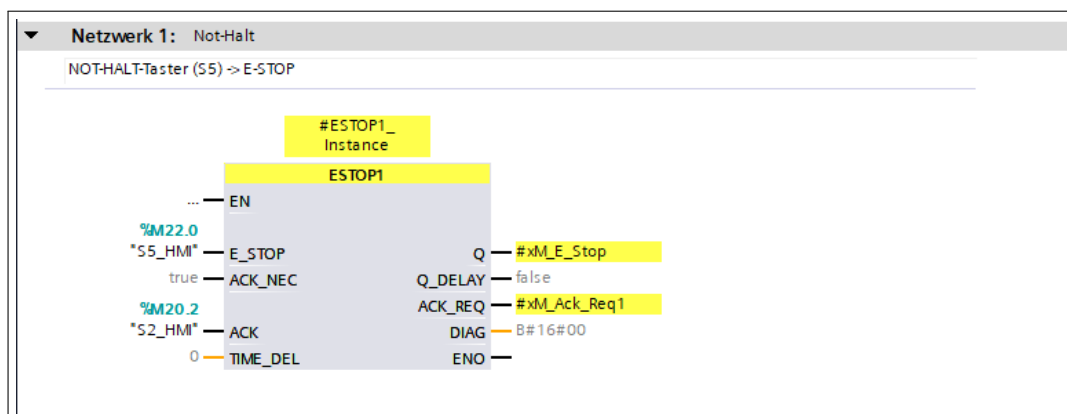


Abb. 2.3: Sicherer Funktionsbaustein für Not-Halt-Funktionalität

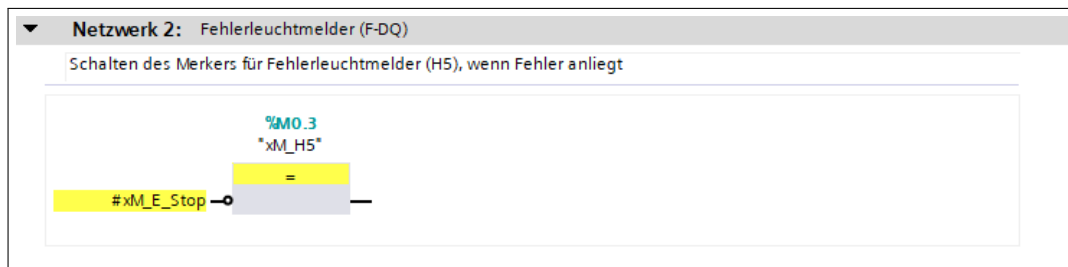


Abb. 2.4: Netzwerk zur Implementierung des Fehlerleuchtmelders

Netzwerk 3 und 4 (Abbildung 2.5 und Abbildung 2.6) zeigen die Beschaltung der Schütze für die Schnecke (K3) und das Förderband (K4) sowie deren zugehörigen Leuchtmelder (H3 und H4). Grundsätzlich gilt, dass wenn ein Not-Halt (xM_E_Stop) ausgelöst wurde, dass beide Schütze und deren zugehörige Betriebsmittel per FALSE-Signal abgeschaltet werden müssen. Im Falle der Förderschnecke gilt weiterhin, dass diese deaktiviert wird, wenn der Endlagenschalter der Schnecke (B1) oder der Endlagenschalter des Förderbandes (B2) oder beide gleichzeitig ausgelöst sind.

Das Förderband kann lediglich zusätzlich zum Not-Halt über die Endlage des Förderbandes (B2) deaktiviert werden.

Die Ansteuerung im Normalbetrieb erfolgt über die Merkervariablen xM_K3 bzw. xM_K4, die im Standard- Anwenderprogramm gesetzt werden.

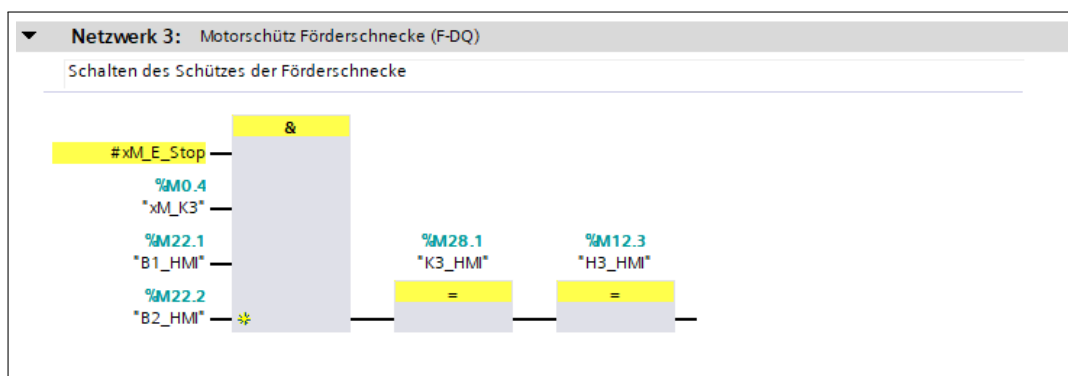


Abb. 2.5: Netzwerk zur Ansteuerung des Motorschützes der Förderschnecke

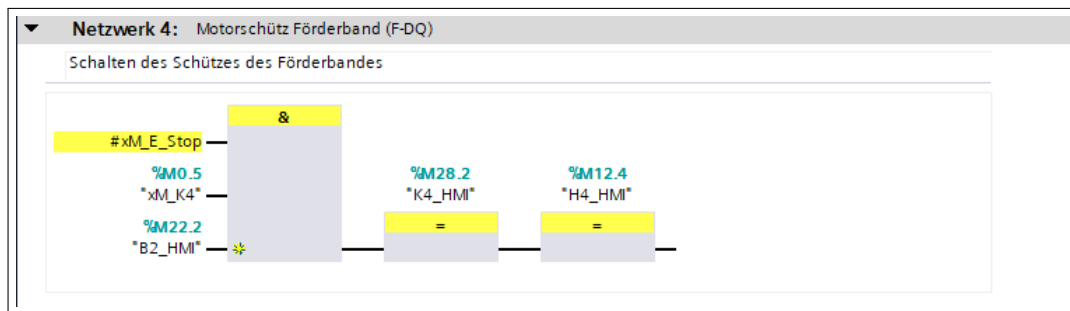


Abb. 2.6: Netzwerk zur Ansteuerung des Motorschützes des Förderbandes

Die Netzwerke 5 und 6 (siehe Abbildung 2.7 und Abbildung 2.8) beinhalten jeweils eine einfache Zuweisung der beiden F-Variablen der Endlagenschalter (B1 und B2) zu Merkervariablen (xM_B1 und xM_B2), die im Standard- Anwenderprogramm nun weiterverwendet werden können. Auch die Endlagen sind zweikanalig ausgeführt und jeweils der Öffner-Kontakt ist mit der Variablen verknüpft, weshalb hier noch Negationen hinzugefügt wurden, um das Anwenderprogramm nicht unnötig zu verkomplizieren.

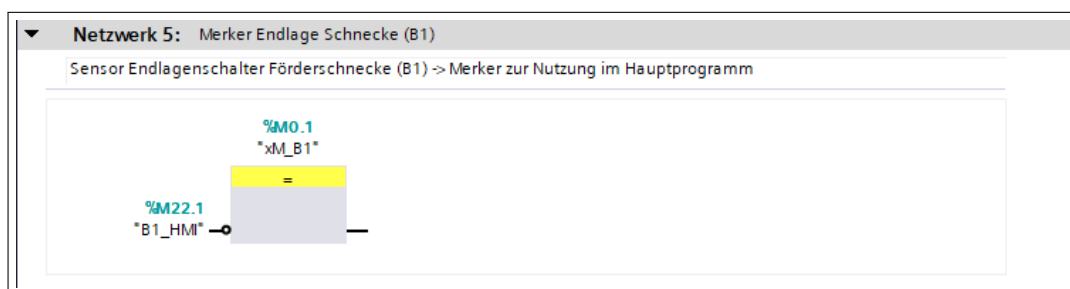


Abb. 2.7: Zuweisung des Sensorsignals der Förderschnecke zu einer Merkervariable

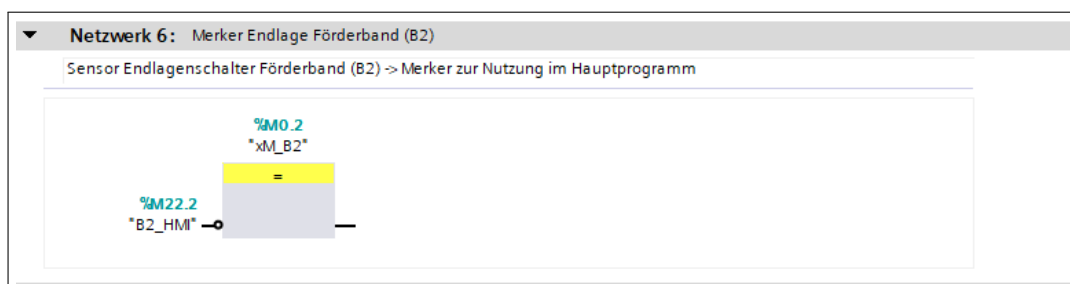


Abb. 2.8: Zuweisung des Sensorsignals des Förderbandes zu einer Merkervariable

In den Netzwerken 7 und 8 (Abbildung 2.9 und Abbildung 2.10) findet eine Diskrepanzanalyse zwischen jeweils den Schützen und den Rückmeldungen der Schütze statt (K3 und B3 bzw. K4 und B4). Es soll folglich ermittelt werden, ob die Werte der jeweils zusammengehörigen booleschen Variablen voneinander abweichen, um zu ermitteln, ob ein Anlagenfehler vorliegt. Dazu wird der Funktionsbaustein **EV1oo2DI** verwendet. Der Baustein besitzt für uns fünf relevante Eingänge. Zunächst **IN1** und **IN2**, welche die beiden Eingangsvariablen (z. B. K3 und B3) entgegennehmen, die auf eine Diskrepanz überwacht werden sollen. Über **DISCTIME** kann eingestellt werden, nach welcher Zeit eine Diskrepanz zwischen den beiden Eingängen zum Auslösen eines Fehlers vergehen muss. Hier wurden konkret zunächst 500 ms eingesetzt, um zu berücksichtigen, dass die Schütze bei einer realen Anlage eine gewisse Zeit benötigen, um zu Schalten. Wie auch schon beim Not-Halt wird hier der Eingang **ACK_NEC** auf TRUE gesetzt, so dass ein Fehler zunächst eine Quittierung erfordert. Am Eingang **ACK** wird erneut der Quittier-Drucktaster (S2) eingesetzt, um einen Diskrepanzfehler quittieren zu können.

Von den Ausgängen des Funktionsbausteins werden lediglich zwei benötigt. Zum Einen für das Quittieren der Ausgang **ACK_REQ**, wenn nach einem Diskrepanzfehler dieser quittiert werden kann. Hier wird auch wieder eine Merkervariable eingesetzt (xM_Ack_Req2 bzw. xM_Ack_Req3), welche in Netzwerk 10 (siehe Abbildung 2.12) mit dem anderen Quittier-Merker verodert werden. Zum Anderen der Ausgang **DISC_FLT**, welcher ein TRUE-Signal ausgibt, wenn eine Diskrepanz zwischen den beiden Eingängen **IN1** und **IN2** vorliegt und ein FALSE-Signal, wenn die Eingänge identische Signalwerte besitzen.

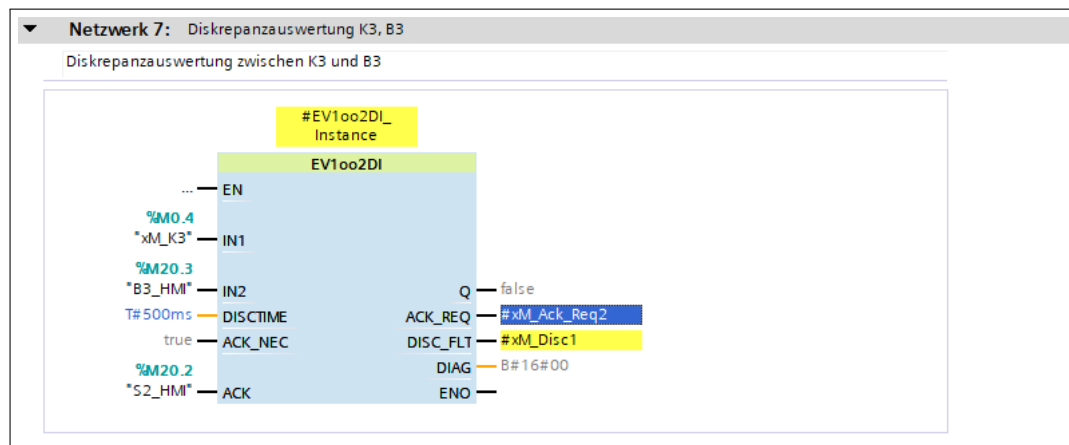


Abb. 2.9: Funktionsbaustein zur Diskrepanzauswertung des Schütz-Schaltzustandes der Förderschnecke mit dem Rückmeldesignal des Schützes

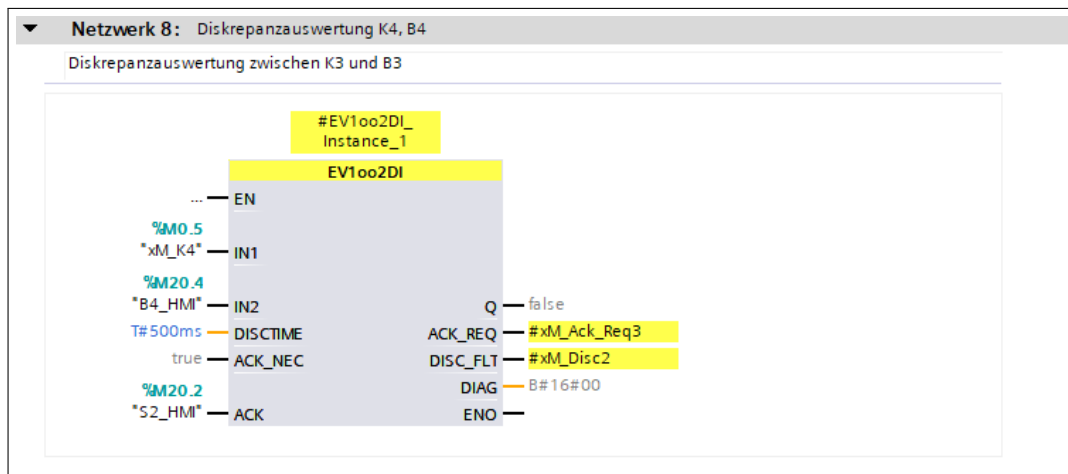


Abb. 2.10: Funktionsbaustein zur Diskrepanzauswertung des Schütz-Schaltzustandes des Förderbandes mit dem Rückmeldesignal des Schützes

In Netzwerk 9 (siehe Abbildung 2.11) werden die Merkervariablen (xM_Disc1 und xM_Disc2) der Ausgänge der Diskrepanzanalyse (DISC_FLT) miteinander verodert. Liegt folglich mindestens eine Abweichung vor zwischen Schütz und Rückmeldung, so wird die Merkervariable für den Not-Halt (xM_E_Stop) gesetzt.

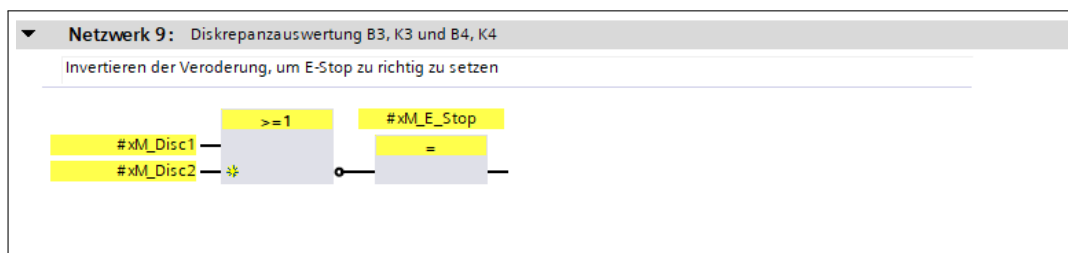


Abb. 2.11: Vereinigung der Diskrepanzauswertungen zu einer Globalen Diskrepanzauswertung der Eingangssignale

Netzwerk 10 (Abbildung 2.12) zeigt wie bereits erwähnt die Veroderung der Quittier-Aufforderungen (Ack_Req). Ist mindestens ein Signal auf TRUE, wird der Nutzer aufgefordert den jeweils aufgetretenen Fehler zu quittieren.

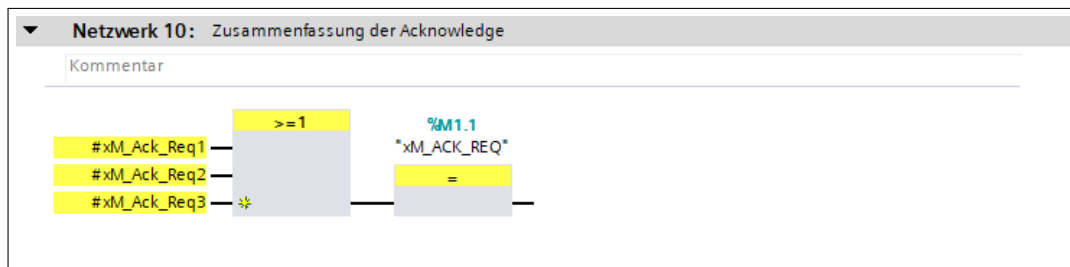


Abb. 2.12: Vereinigung der Signale zur Quittieraufforderung

Das letzte Netzwerk des Sicherheitsprogramms (siehe Abbildung 2.13) zeigt den Funktionsbaustein zum globalen Quittieren von Fehlern. Über diesen können alle F-Peripherie und F-Ablaufgruppen wieder eingegliedert werden nach z. B. einem Kanalfehler mit anschließender Passivierung. Der Baustein besitzt lediglich einen Eingang (**ACK_GLOB**). Auch an diesem wird der Quittier-Taster (S2) eingesetzt.

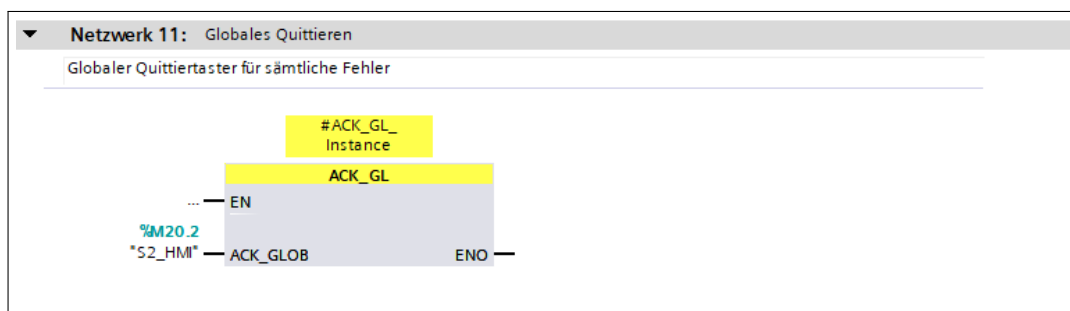


Abb. 2.13: Sicherer Funktionsbaustein zur Umsetzung einer globalen Quittierfunktionalität

3 Visualisierung

Da zum Zeitpunkt der Dokumentationserstellung keine reale Anlage zur Verfügung stand, wurde mit der SIMATIC HMI eine Visualisierung angefertigt (Abbildung 3.1). Die Visualisierung spiegelt den realen Aufbau wieder. Zur Überprüfung der Schütze (K3, K4), wurden zwei weitere Leuchtmelder eingefügt. Diese sind in der realen Anlage nicht vorhanden. Im Gegensatz zur realen Anlage können durch die Visualisierung nicht alle Sachverhalte korrekt dargestellt werden. Somit sind die Öffner-Taster mit dem Kommentar „Toggle“ versehen, da in der Simulation keine öffnenden Taster eingefügt werden können. Bei der Bedienung ist darauf zu achten, dass ein Klicken das jeweilige Bit nur invertiert! Die Endlagentaster des Förderbands und der Förderschnecke sind durch B1 und B2 dargestellt und ebenfalls mit einem Kommentar versehen worden.

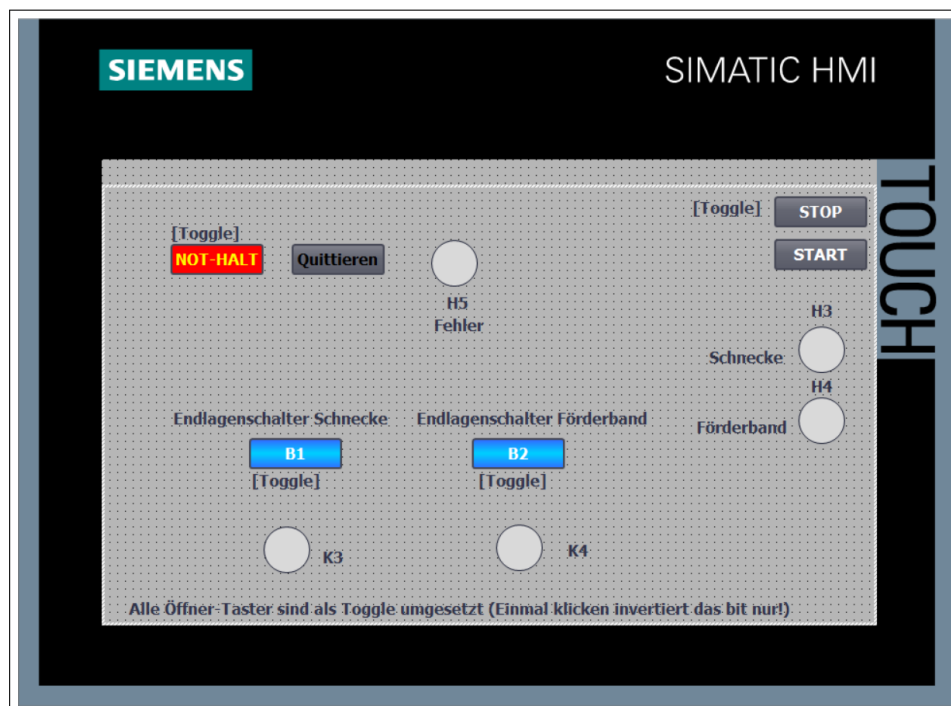


Abb. 3.1: Visualisierung mit SIMATIC HMI unter Nutzung der Software WinCC

Literaturverzeichnis

- [1] HTW-Logo auf dem Deckblatt
https://de.wikipedia.org/wiki/Datei:Logo_HTW_Berlin.svg
Stand: 17.08.2018 um 14:49 Uhr
- [2] HTW-Logo in der Kopfzeile
<http://tonkollektiv-htw.de/>
Stand: 17.08.2018 um 14:53 Uhr
- [3] Informationssystem des TIA Portals V17
<https://support.industry.siemens.com/cs/document/65601780/tia-portal-ein-%C3%BCberblick-der-wichtigsten-dokumente-und-links-steuerung?dti=0&lc=de-DE>
Stand: 17.09.2022 um 09:40 Uhr