

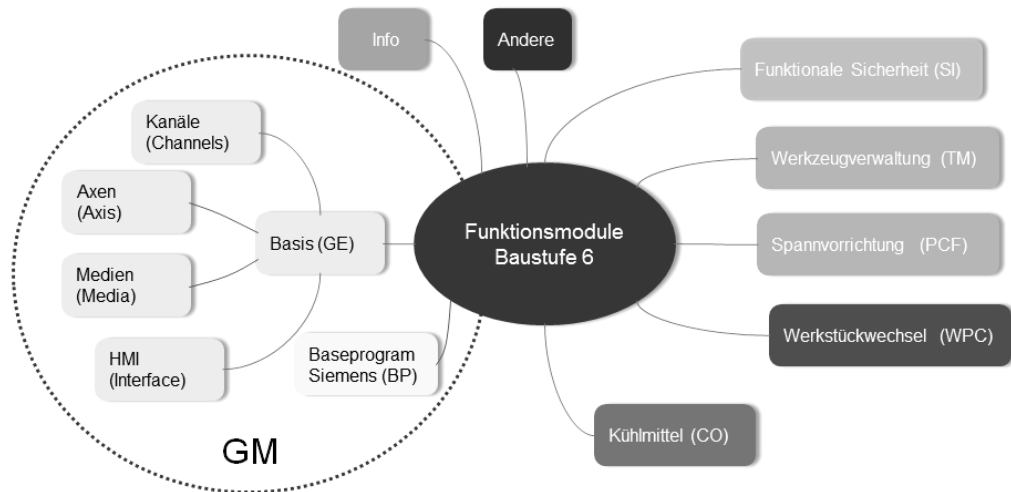
Baustufe 6 Neuheiten

1	Funktionsmodule	2
2	Funktionale Planung	4
3	Symbolische Adressierung	5
4	PLC-Interface	7
4.1	PLC-Interface	7
5	Einrichten am Magazin (HMI)	10
6	Funktionsorientierte HMI Bilder	11
7	Handling Alarmer, Meldungen, Merker und Timer	14
7.1	Alarmer und Meldungen in Baustufe 6	14
7.2	Timer	17
7.3	Merker	19
8	Erweiterung schneller Datenkanal	21
9	M-Funktionsarten überarbeitet	23
9.1	M-Funktionsarten im DB165	23
10	Achs-Handling überarbeitet	27
11	Safety neu strukturiert	30
12	SCL-Bausteine	36
13	Multiinstanzen	39
13.1	Instanz und Multiinstanz	39
13.2	Testbetrieb aktivieren	43
13.3	Bausteine beobachten	44
14	Kanalstruktur Baustufe 6	45
15	Grundstellungskonzept	47
16	Kundenspezifische Netzwerke (SPEC)	49
17	Sonstige Neuerungen	50
18	Projektvorlage	51
18.1	Projektvorlage erstellen	51
18.2	Maschine erstellen	53

1 Funktionsmodule

In Baustufe 6 wurden sämtliche Bausteine in Modulen geordnet. Alle ein Modul betreffenden Bausteine befinden sich im Modul. Die Kommunikation zwischen Funktionsmodulen erfolgt über definierte, standardisierte Schnittstellen. In Baustufe 5 mögliche Überschneidungen zwischen Funktionsmodulen und Mehrfachverwendungen von Funktionsbausteinen wurden damit aufgelöst.

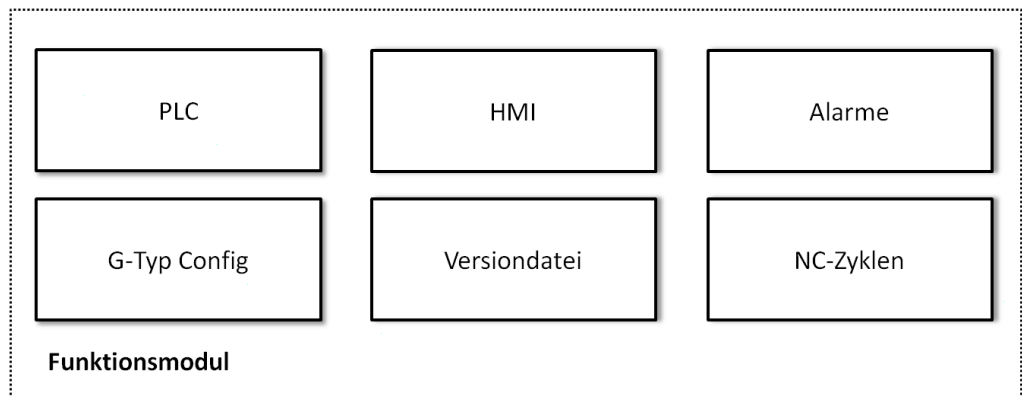
Module



Funktionsmodule organisieren in Baustufe 6 Funktionsbausteine (FB), Funktionsaufrufe (FC) und andere Bausteine modular.

Funktionsmodule sind einerseits Basis-Funktionsmodule (GM) wie das Baseprogramm und die Basis (GE) mit Kanälen, Achsen, Medien und dem Interface (HMI) und andererseits Funktionsmodule für Sicherheit (SI), Werkzeugverwaltung (TM), Werkzeugwechsel (TM) und andere Funktionsgruppen der Maschinen. Diese Module werden je nach Maschine verwendet oder angepasst.

Funktionsmodul

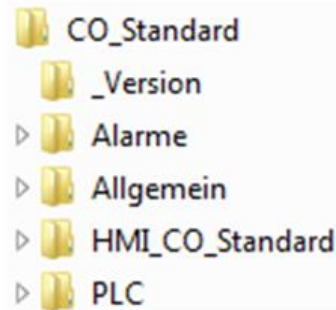


Einzelne Funktionsmodule beinhalten sämtliche benötigten Dateien wie z.B. Alarme, HMI-Daten, die „G-Typ Config“, die Versionsdatei, NC-Zyklen und NC-Definitionen oder die benötigten PLC-Bausteine (FCs, FBs und DBs). Die Dateien werden beim Erstellen der Maschine zusammengeführt.



In Baustufe 6 gibt es ein gemeinsames Basiskonzept für Universal- und Systemmaschinen!

Beispiel Im Funktionsmodul Kühlmittel (CO) sind PLC-Bausteine wie z.B. Funktionsbausteine, Alarmer, HMI-Dateien, die Versionsdatei und die Datei „G-Typ Config“ enthalten. Die Datei „G-Typ Config“ ist im Verzeichnis „Allgemein“ gespeichert. NC-Zyklen und Definitionen werden im CO-Modul nicht verwendet.



Timer Timer, Merker und Alarmer sind in Baustufe 6 umstrukturiert. Funktionsbausteine für Timer, neu organisierte Alarmer und Datenbausteine für Merker übernehmen ihre Funktion.

Timer und Merker werden nur noch eingeschränkt eingesetzt.

Baustufe 5 In Baustufe 5 waren Funktionsgruppen auf viele Bausteine verteilt. Durch diesen Aufbau ergaben sich folgende Nachteile bei der Programmierung:

- In der PLC waren Timer nur eingeschränkt verfügbar.
- Überschneidungen zwischen Funktionsmodulen führten zu Schwierigkeiten bei der Programmierung von Alarmen.
- Die beschränkte Zahl von Merkern in der PLC führte zu Problemen bei der Verwendung vieler Merker.

Die Modularisierung und Anpassung der Funktionsgruppen war insgesamt relativ kompliziert. Änderungen bedingten Anpassungen aller Bausteine. Dies betraf vor allem Universalmaschinen. Sowohl Verwaltung als auch Versionierbarkeit wurden dadurch eingeschränkt.

2 Funktionale Planung



Alle Symbole müssen in der Elementarfunktionsliste erstellt oder geändert werden. Verantwortlich ist dabei die Konstruktion.



Im Funktionsplan sind nur nicht-elektrische Betätigungen aufgeführt! Elektrische Aktoren oder Sensoren sind im Funktionsplan nicht angeben.

Funktionsplan

In Baustufe 6 werden zu Funktionsgruppen gehörende Funktionen, Elementarfunktionen und deren symbolische Adressen in einem Funktionsplan beschrieben und definiert. Nicht-elektrische Betätigungen für Hardware bzw. deren symbolische Adressen sind ebenfalls im Funktionsplan aufgeführt.

Struktur

Auf der obersten Ebene werden Funktionsgruppen wie z.B. die Funktionsgruppe für die Maschinenverkleidung (1) benannt. Zu der Funktionsgruppe gehören Funktionen wie z.B. „Arbeitsraum mit Schutzeinrichtung vom Bediener trennen“.

a) Funktionsgruppe c) Elementarfunktion
b) Funktion d) Elementarfunktions-
 variante e) Symbolische
 Adresse f) Betätigung
 Kommentar

GROB Funktionsplan					
Funktionsgruppe	11000	MEC	Maschinenverkleidung		
Funktion	11010	SW	Arbeitsraum mit Schutzeinrichtung vom Bediener / Beladeeinrichtung trennen		
Funktionsvariante	201		manuelle Arbeitsraumbeladetür		
Funktion Ausführungsvariante	01		Schutztür / SES119		
Version	00				
Elementarfunktion	E120		Schutztür Arbeitsraum Beladen		
Elementarfunktionsvariante	ML		manuell		
Symbolische Adresse	Position	Text	Anschluss	Betätigung	
mMEC_SW_E120_ML_F	F	Schutztür Arbeitsraum Beladen öffnen	-	manuell	
mMEC_SW_E120_ML_H	H	Schutztür Arbeitsraum Beladen schließen	-	manuell	
Elementarfunktion	E121		Schutztür Arbeitsraum Beladen geschlossen überwachen und zuhalten		
Elementarfunktionsvariante	FN		Sicherheitsendschalter		
Symbolische Adresse	Position	Text	Anschluss	Betätigung	
iMEC_SW_E121_FN_H	H	Schutztür Arbeitsraum Beladen geschlossen und zugehalten	-	-	
mMEC_SW_E121_FN_H	H	Schutztür Arbeitsraum Beladen Zuhaltung betätigen	-	Feder	
				Ausführung	

Elementarfunktionen wie z.B. E120 Schutztür Arbeitsraum beladen stehen unter den Funktionen. Für ihre Programmierung werden symbolischen Adressen wie z.B. „iMEC_SW_E121_FN_H“ verwendet. Dadurch besteht keine statische Zuweisung von Symbolen und Ein- und Ausgängen mehr. Manuelle und mechanische Betätigungen werden rechts angegeben.

3 Symbolische Adressierung

Grund Durch die symbolische Adressierung wird der Umstieg auf TIA-Portal vorbereitet. Im Nachfolger von Step7 **muss** symbolisch programmiert werden.

Symbole In Baustufe 6 werden Hardwareadressen indirekt adressiert. Statt Adressen werden symbolische Adressen (Symbole) verwendet. Symbole werden durchgängig verwendet und in der mechanischen Planung über ein System aus Buchstaben und Zahlen festgelegt. Sie werden später nicht verändert.

Aufbau Symbole bestehen aus Zeichenfolgen aus sechs einzelnen Abschnitten mit 23 Buchstaben:

Einzelne Abschnitte bezeichnen den jeweiligen Aktor/Sensor, Zuordnung zu Funktionsmodul, Funktion und Elementarfunktion, und die Position. Unterstriche sind Platzhalter.

Aktor/Sensor Am Anfang des Symbols steht ein klein geschriebener Buchstabe für den Aktor/Sensor, wie z.B. ein Eingang (Input):

Aktor/Sensor	
i	= Input (IEC 61131-3)
q	= Output (IEC 61131-3)
m	= mechanical, manual (z.B. manuelle Türe oder Motorbremse)
n	= NC - relevant (z.B. Vorschubantrieb oder Motorgeber)

Funktionen Für die Bezeichnung von Funktionsgruppe, Funktion, Elementarfunktion und Elementarfunktionsvariante wird immer ein bestimmtes, Grob-spezifisches Kürzel verwendet (siehe Elementarfunktionsliste).

Position Danach folgt die Positionsangabe mit einem Buchstaben. Die Buchstaben bezeichnen sowohl Positionen wie z.B. Endstellungen als auch andere Werte oder Zustände, z.B. Werte von Temperatursensoren oder Zustände wie „OK“, „Not OK“ oder die Position eines Gebers:

Position	
A	= Analogwert (z.B. Temperatursensor)
F	= Final position 1 (Endstellung / eingeschaltet / bestätigt)
F2	= Final position 2
H	= Home position 1 (Grundstellung / ausgeschaltet / n. betätigt)
H2	= Home position 2
M	= rein mechanisch
N	= Not OK (Nicht in Ordnung)
O	= OK (in Ordnung)
P	= Position (z.B. Geber oder Motor)
R	= Reserve
V	= View 1 (reine Anzeige: z.B. LED Türschalter)
V2	= View 2

Beispiel Über die symbolische Adresse „iMEC-SW-E121_FN_H“ (1) wird in der Programmierung erfragt, ob die Schutztür des Arbeitsraumes geschlossen und zugehalten ist:

Elementarfunktion ①	E121	Schutztür Arbeitsraum Beladen geschlossen überwachen und zuhalten		
Elementarfunktionsvariante	FN	Sicherheitsendschalter		
Symbolische Adresse	Position	Text	Anschluss	Betätigung
iMEC_SW_E121_FN_H	H	Schutztür Arbeitsraum Beladen geschlossen und zugehalten	-	-
mMEC_SW_E121_FN_H	H	Schutztür Arbeitsraum Beladen Zuhaltung betätigen	-	Feder
qMEC_SW_E121_FN_F	F	Schutztür Arbeitsraum Beladen Zuhaltung lösen	-	-

Am Anfang steht die Angabe „i“ (Eingang als Aktor/Sensor) gefolgt von der Bezeichnung „MEC“ (Funktionsgruppe Maschinenverkleidung).

iMEC-SW-E121_FN_H

Darauf folgt die Funktionsbezeichnung „SW“ (Arbeitsraum mit Schutzeinrichtung trennen), „E121“ für die Elementarfunktion (Schutztür Arbeitsraum Beladen), „FN“ als Bezeichnung für die Elementarfunktionsvariante (Sicherheitsendschalter) und die Positionsangabe „H“ (Grundstellungsposition).

Liste Elementarfunktionen

Die gepflegte Liste mit Elementarfunktionen befindet sich unter folgender Adresse:
http://grob-portal/Abteilungen/KO-FE/UM/Uebersichten_Gxx6/Elementarfunktionen_Uebersicht.xlsm

Mit einer Schaltfläche des GROB-Tools „Schnellstart“ kann die Liste ebenfalls geöffnet werden. Es muss dazu konfiguriert sein.

SW-Erstellung

In Baustufe 6 kann die SW-Erstellung bereits vor Erstellung der Zuordnungsliste (ZULI) erfolgen, da Symbole unabhängig vom Projekt gleich bleiben. Die ZULI wird automatisch aus EPLAN generiert.

Baustufe 5

In Baustufe 5 war zum Verschieben von Hardware-Adressen immer eine Software-Anpassung erforderlich. Der Aufwand zum Verändern der Software war relativ hoch und parallele Entwicklung von Hardware und Software erschwert.

4 PLC-Interface

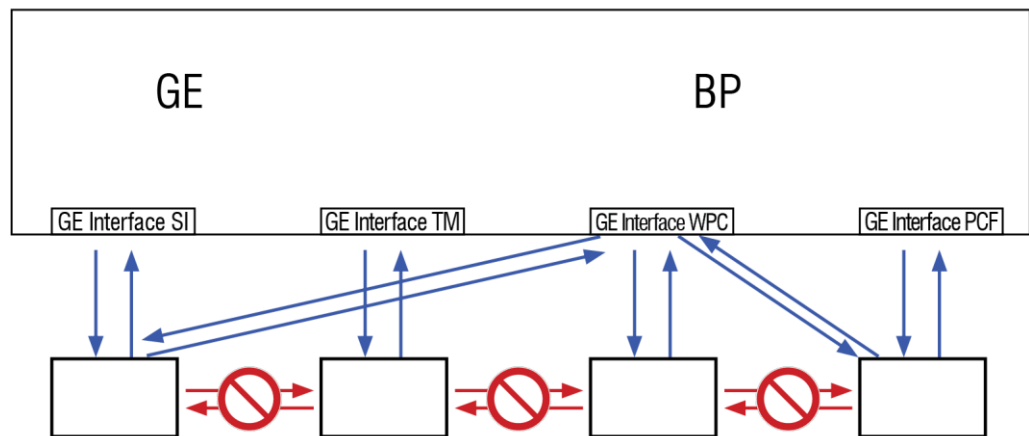
4.1 PLC-Interface



In Baustufe 6 erfolgt Datenaustausch zwischen den Funktionsmodulen nur über die Schnittstelle GE-Interface! Der direkte Datenaustausch zwischen Funktionsmodulen ist nicht erlaubt.

In Baustufe 6 wurde für den Datenaustausch zwischen den Datenbausteinen das GE-Interface eingerichtet. Das GE-Interface ist Teil der Basis (GE). Es kann aus allen Funktionsmodulen verwendet werden.

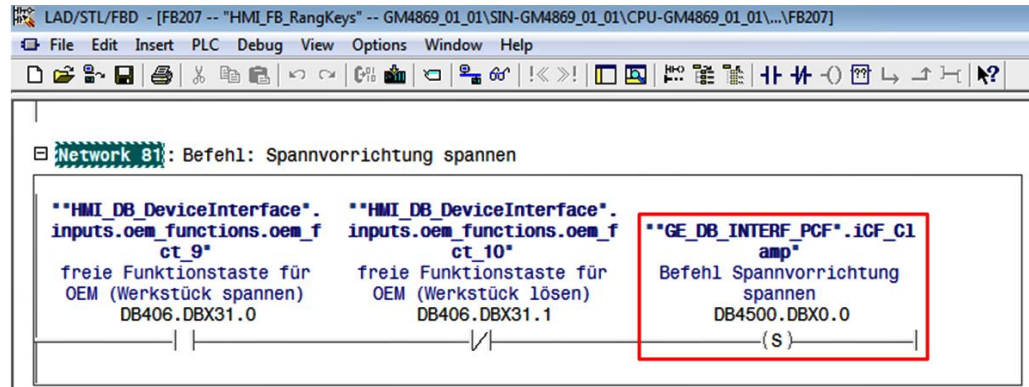
GE Interface Der Datenaustausch zwischen den Funktionsmodulen erfolgt in Baustufe 6 nur über das GE-Interface. Über das GE-Interface können die Datenbausteine Funktionen steuern und Rückmeldungen erhalten. Die abgekürzten Funktionsaufrufe wie z.B. TM für „Tool Management“ sind dabei Teil des Funktionsaufrufs.



Bits Die im Interface mit „i“ gekennzeichneten Bits sind globale Eingänge um Funktionen zu steuern. Im Interface mit „q“ gekennzeichnete Bits sind Rückmeldungen des Moduls, wie z.B. „Achse ist vorhanden“.

Beispiel Für das Spannen und Lösen eines Werkstückes werden die Bits über das GE-Interface übertragen. Für das Spannen wird das Bit iC_fClamp übertragen. Die Rückmeldung erfolgt über das Bit qCF_Clamped.

Nach Betätigen einer Funktionstaste im HMI wird der Befehl „Spannvorrichtung spannen“ über das GE-Interface an die Spannvorrichtung (PCF) gesendet:



Die Spannvorrichtung wird dabei auch mit dem FB207 und dem Modul WPC (FB 5100) angesteuert:

Address:

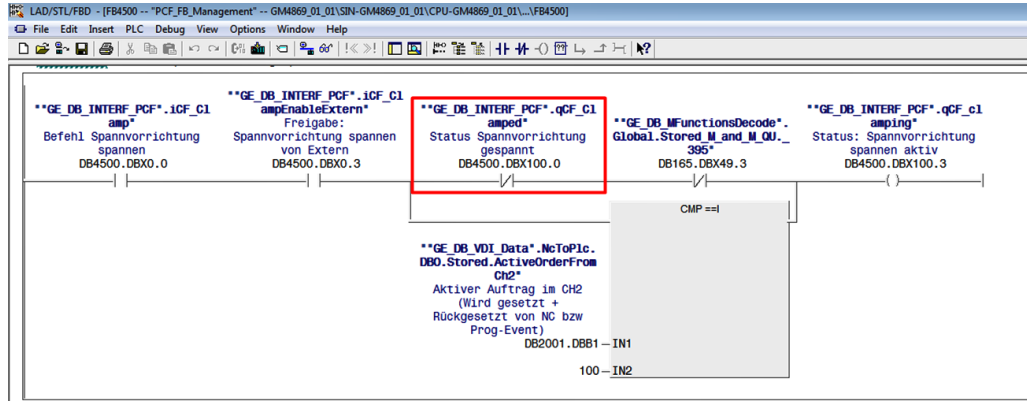
Block	Block symbol	Details	Typ
FB207	HMI_FB_RangKeys	NW 81 /S	W
FB4500	PCF_FB_Management	NW 5 Sta 12 /S	W
FB4500	PCF_FB_Management	NW 15 Sta 5 /S	W
FB4500	PCF_FB_Management	NW 232 Sta 3 /R	W
FB5100	WPC_FB_ManualFunctions	NW 122 /S	W
FB4500	PCF_FB_Management	NW 16 Sta 1 /A	R

Die verwendete symbolische Adresse bzw. das Bit ist „iCF_Clamp“:

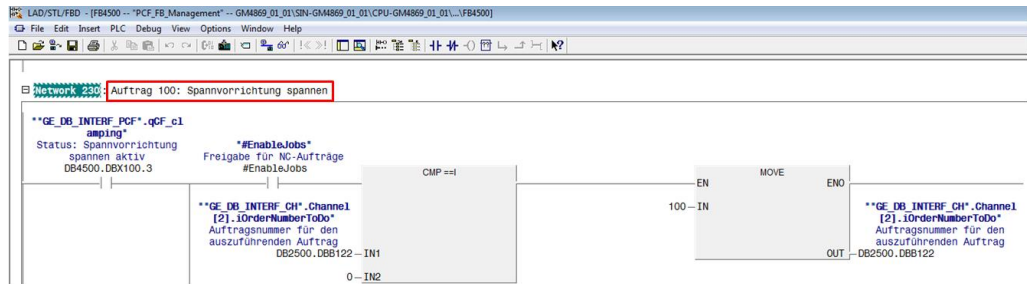
Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	iCF_Clamp	BOOL	FALSE	Befehl Spannvorrichtung spannen
+0.1	iCF_Unclamp	BOOL	FALSE	Befehl Spannvorrichtung lösen
+0.2	iEnablePressure	BOOL	FALSE	Freigabe die Kupplungen zu beaufschlagen
+0.3	iCF_ClampEnableExtern	BOOL	FALSE	Freigabe: Spannvorrichtung spannen von Extern
+0.4	iCF_UnclampEnableExtern	BOOL	FALSE	Freigabe: Spannvorrichtung lösen von Extern

Nach Senden der Daten über das Interface wird die Spannvorrichtung gespannt. Kanal 2 erhält den Auftrag zur Spannung der Spannvorrichtung (Auftrag 100) und gibt über das GE-Interface Rückmeldung.

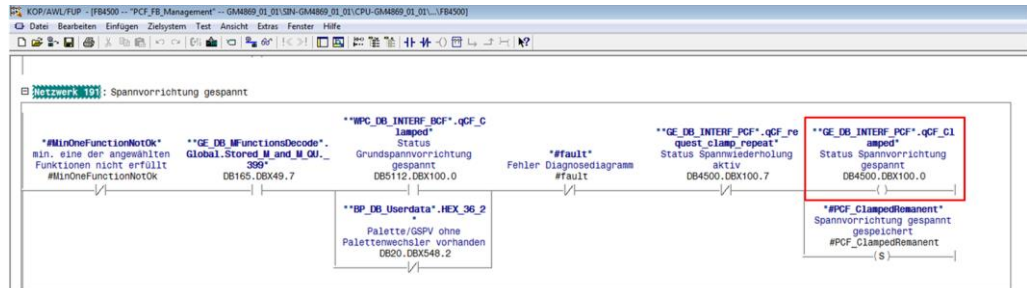
Rev. 1.0



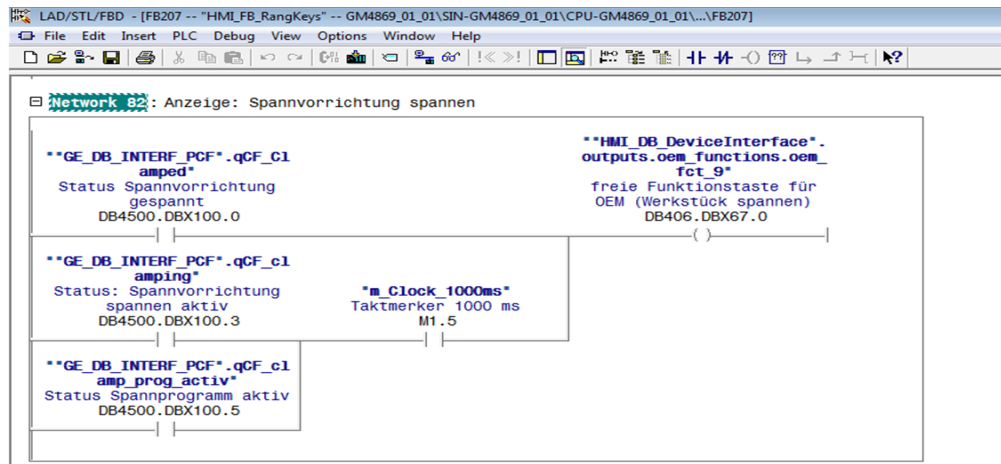
Kanal 2 arbeitet jetzt den Auftrag 100 ab. Sobald die Freigaben gesetzt sind, wird die Spannvorrichtung gespannt.



Spannen Nach dem Spannvorgang wird „Status Spannvorrichtung gespannt“ (qCF_Clamped) über das GE-Interface zurück gemeldet.



Die Rückmeldung für die Anzeige wird im FB207 verwendet:

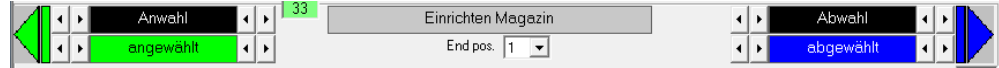


baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid

5 Einrichten am Magazin (HMI)

In Baustufe 6 sind Jog-Bewegungen der Magazinachsen geschützt. Werkzeugwechsel-Achsen können im Einrichtbetrieb nur über „Einrichtbetriebsart Magazin“ verfahren werden (ohne Verriegelungen).



- JOG**
 - Bewusstes Freischalten der JOG-Funktionen (Ohne Anwahl ist keine Bewegung möglich)
 - Begrenzte Verriegelungen
- Baustufe 5**
 - Freigabe für das Fahren im JOG war sehr aufwendig (z.B. viele SW-Nocken, komplexe Verriegelungsnetzwerke)
 - Automatikbetrieb: die Vorschubsperrn sind nur vom NC-Programm aktiv
 - Einrichten: im WZW sind alle Vorschubsperrn ausgeschaltet (SERVICE-Mode)
 - Absolute Verriegelung war nicht möglich (z.B. Interpolation)

Rev.

baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid

6 Funktionsorientierte HMI Bilder

In Baustufe 6 sind Funktionen nicht mehr auf mehrere HMI-Bilder verteilt. Settingbilder werden über variable Layouts konfiguriert. Die Schnittstelle zu HMI bildet der DB 159 zusammen mit dem DB2300.

Funktionen Funktionen sind in Baustufe 6 nicht mehr auf mehrere HMI-Bilder verteilt, sondern komplett in einem Bild vorhanden:

The screenshot shows the 'Spindel' (Spindle) control interface. It includes the following elements:

- Spindellaufzeit:** 0 [h] 0 [m]
- Anzahl Hübe des Spannsatzes:** 0
- Spindel bei offener Türe drehbar:** nein
- Istwert Spindelzange:** 0.00 [U]
- Freigabe Wertänderung:** (Green indicator)
- gelöst:** (Green indicator)
- gespannt mit Werkzeug:** (Green indicator)
- gespannt ohne Werkzeug:** (Green indicator)
- Einstellwert:** 0.00 [U], 0.10 [U], 0.10 [U]
- + Toleranz:** 0.10 [U], 0.10 [U], 0.05 [U]
- Toleranz:** 1.00 [U], 1.55 [U], 1.55 [U]

Settingbilder Settingbilder werden in Baustufe 6 über variable Layouts konfiguriert. Mit variablen Layouts können beliebige Zustände und Werte über frei gewählte Elemente visualisiert und verändert werden.

Datenbausteine Schnittstelle In Baustufe 6 ist der Schnittstellen-Datenbaustein für das HMI der DB159 (1). Er wird allerdings nicht mehr direkt verwendet, sondern in der PLC von dem Datenbaustein DB2300 beschaltet (2).

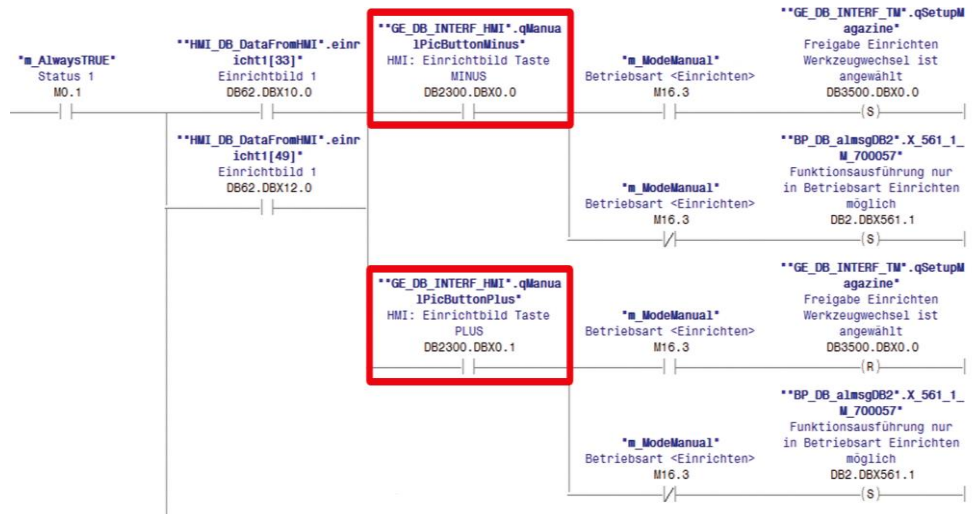
The screenshot shows the configuration software interface with the following settings:

- Presettings:**
 - Touch activation
 - Press 2nd?
 - Direct key byte: 28
 - Width of the move on the touch area in %: 20
 - Without direct key module or MPP/MCP
 - Move gesture active
 - NetNames active
 - Permanent control assignment according to NetNames for header
 - Select screen via PLC
 - Alternative address
 - Change the standard data block (DB 159)
- System menu:**
 - Language change hotkey
 - Password input hotkey
 - Lamp test hotkey
- Softkey:**
 - One-row
 - Two-row
- Network active:**
 - Fetch IP from DHCP
 - Assign IP address
 - Computer name: []
 - IP address: 1.1.1.0
 - Subnet mask: 255.255.255.0
 - Gateway: []
- Bit monitoring, language change (DB59.DBX96.3):**
 - Bit monitoring, language change (DB59.DBX96.3)
- Set control address at project installation:**
 - Set control address at project installation
 - IP address of the control: []
- Softkey matrix via PLC:**
 - Softkey matrix via PLC (DB 2300, Word 50)

DB159 Der DB159 ist der Nachfolger des DB59. Die Verschiebung auf den DB159 ermöglicht die Nutzung aller 31 möglichen Achsen von Siemens. Er wird in der Logik nicht verwendet, bleibt aber als Schnittstelle zu Siemens bestehen.

Einrichtzeilen In Baustufe 6 können Einrichtzeilen variabel angepasst werden. Dafür wird der DB2300 verwendet. Er bildet die Schnittstelle zu HMI und beschaltet den DB159.

Beispiel:



Einrichtbilder Im DB2300 verwendet Einrichtbild 1 die Zeile 1 bis 256 (1). Einrichtbild 2 verwendet die Zeile 1 bis 256 (2).

DB2300 DB_GE_INTERF_HMI		
+288.0	Screen1_JogLine	ARRAY[1..256] ①
*2.0		"UDT_HMI_JogLine"
+800.0	Screen2_JogLine1	ARRAY[1..256] ②
*2.0		"UDT_HMI_JogLine"

Rev.

baustufe 6 neuigkeiten 1,1.docx

Hubert Schmid

Bits Die Einrichtzeilen werden mit dem „User Data Type“ 165 (UDT 165) programmiert. Dabei stehen folgende Bits pro Zeile zu Verfügung:

UDT165 UDT_HMI_JogLine

+0.0	iLineInUse	Einrichtzeile wird verwendet	Wenn Zeile benutzt wird auf TRUE setzen
+0.1	iChangeLeftRightinCase	Tausche Links Rechts wenn Kundenwunsch	Zeile tauschen wenn Grundstellung links
+0.2	iDoNotShowThisLine	Einrichtzeile Ausblenden	Bei 1 wird Einrichtzeile dynamisch ausgeblendet
+0.3	iMovingActiveLeft	Bewegung läuft Links	Bewegung links aktiv wird angezeigt
+0.4	iMovingActiveRight	Bewegung läuft Rechts	Bewegung rechts aktiv wird angezeigt
+0.5	iEnableLeft	Bewegungsfreigabe Links	Bedingungen für Bewegung links vorhanden
+0.6	iEnableRight	Bewegungsfreigabe Rechts	Bedingungen für Bewegung rechts vorhanden
+0.7	iInFinalPositionLeft	Endlage Links erreicht	Endlage links erreicht
+1.0	iInFinalPositionRight	Endlage Rechts erreicht	Endlage rechts erreicht
+1.1	qOrderLeft	Befehl: Bediener hat links gedrückt	Taste links gedrückt
+1.2	qOrdnerRight	Befehl: Bediener hat rechts gedrückt	Taste rechts gedrückt

Das Bit „iLineInUse“ (1) bestimmt dabei, ob die Einrichtzeile verwendet wird.

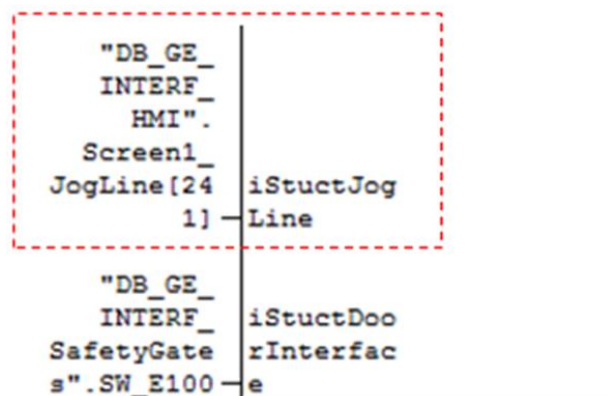
Wird die Verwendung einer Einrichtzeile nicht zyklisch auf „1“ gesetzt, wird der DB2300 nicht zyklisch auf den DB159 umgeladen. Die GROB-Diagramme können dann die Einrichtzeile verwenden.



Hinweis

Einrichtzeile Mit dem Any Pointer iStructJogLine kann einem Baustein eine komplette Einrichtzeile übergeben werden.

FB4022 FB_SI_EuchnerLookASI



Baustufe 5 In Baustufe 5 war eine Funktion auf mehrere HMI-Bilder verteilt (z.B. Einrichten, Setting, VarLayout). Für Settingbilder wurden keine variablen Layouts verwendet. Damit war das HMI unflexibel und zum Teil kompliziert.

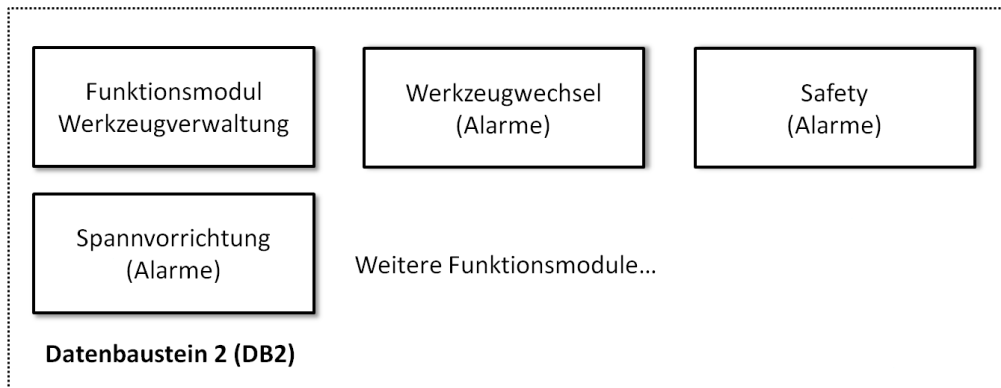
7 Handling Alarmer, Meldungen, Merker und Timer

7.1 Alarmer und Meldungen in Baustufe 6

Alarmer sind in Baustufe 6 einzelnen Funktionsmodulen zugeordnet. Die Struktur von zugeordneten Datenbausteinen wurde ebenfalls verändert. Als Ergebnis ist die Verwaltung vereinfacht und Alarmer lassen sich häufiger und flexibler einsetzen.

Organisation In Baustufe 6 ist die feste Verkettung von Alarmen zu Alarmmeldungen (Reaktionen) aufgehoben. Reaktionen werden jetzt direkt mit dem Alarm programmiert.

DB2



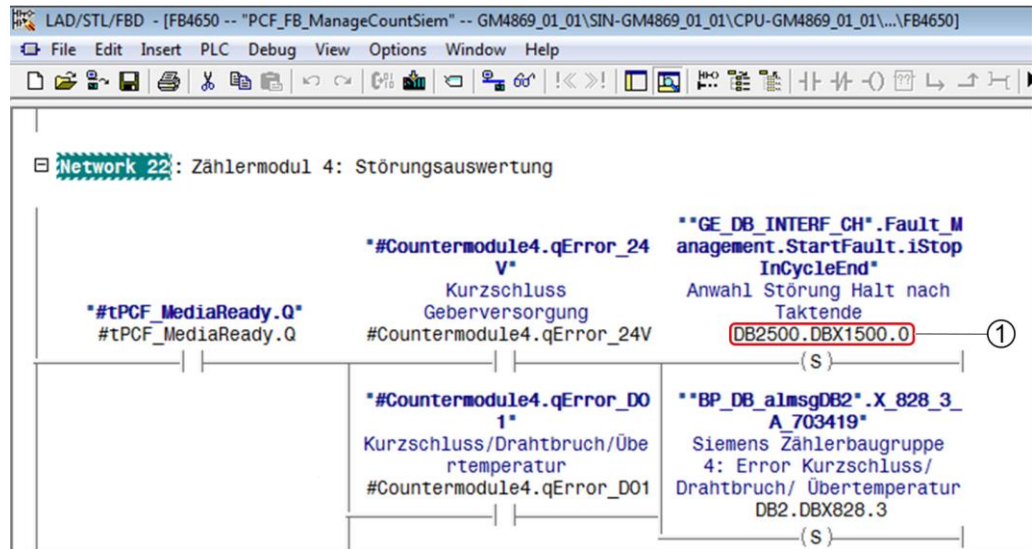
Jedes Funktionsmodul hat innerhalb des Datenbausteins 2 (DB2) einen eigenen Bereich für seine Alarmer. Blockweise Pflege ist dadurch möglich. Der Datenbaustein 126 (DB126) ist in Baustufe 6 ausschließlich für Diagramme reserviert.

Reaktionen Alarmreaktionen sind in Baustufe 6 von Alarmen gelöst. Sie werden direkt mit dem Alarm programmiert und lassen sich deshalb mehrfach verwenden. Sie werden vorwiegend über den Datenbaustein 2500 (DB2500) programmiert.

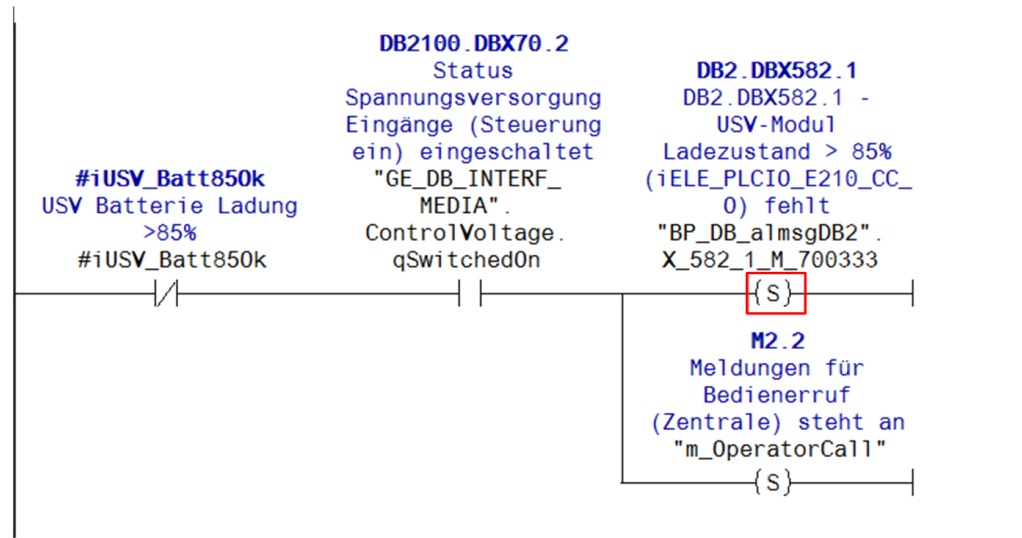
Beispiel Eine Alarmreaktion für den Soforthalt auf allen Kanälen wird über "Fault Management.StartFault.iStopDirectAllChannels" (1) aufgerufen.

1500.0	Fault_Management.StartFault.iStopInCycleEnd	BOOL	FALSE
1500.1	Fault_Management.StartFault.iStopToolNotInWorkpice	BOOL	FALSE
1500.2	Fault_Management.StartFault.iStopDuringToolChange	BOOL	FALSE
1500.3	Fault_Management.StartFault.iStopAfterToolChange	BOOL	FALSE
1500.4	Fault_Management.StartFault.iStopDirectAllChannels	BOOL	FALSE
1500.5	Fault_Management.StartFault.iStopDirectAndMachineOff	BOOL	FALSE
1500.6	Fault_Management.StartFault.iFaultWhithoutReaction	BOOL	FALSE
1502.0	Fault_Management.StartFault.iStopDirectChannel[1]	BOOL	FALSE
1502.1	Fault_Management.StartFault.iStopDirectChannel[2]	BOOL	FALSE
1502.2	Fault_Management.StartFault.iStopDirectChannel[3]	BOOL	FALSE
1502.3	Fault_Management.StartFault.iStopDirectChannel[4]	BOOL	FALSE

In Kontaktplan (KOP) wird die Reaktion über den Datenbaustein 2500 (DB2500) programmiert (1):



Meldungen Meldungen werden in Baustufe 6 auch **gesetzt (S)**. Somit ist ein mehrfaches Programmieren einzelner Meldungen möglich.



Die folgenden beiden Merker behalten ihre Funktion:

- M2.2 Bedieneruff
- M2.3 Alarmserver DB126

Rev. 1.0

baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid

Sperrern auslösen Auch Vorschubsperrern (FeedDisable), Einleeseperrern (ReadInDisable) und Startsperrern (StartDisable) für **Kanäle** werden über den Datenbaustein 2500 (DB2500) programmiert:

Adresse	Name
0.0	
+0.0	Channel
*0.0	
+0.0	iOrderFeedDisable
+0.1	iOrderReadInDisable
+0.2	iOrderStartDisable

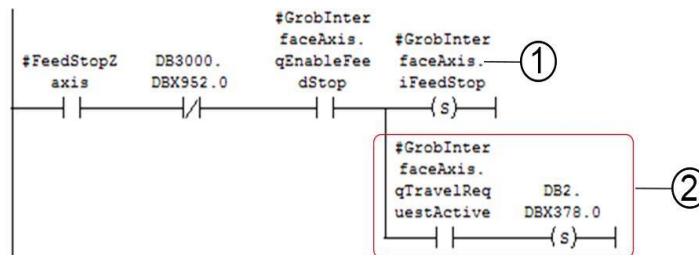
Der Offset beträgt dabei 100 Byte pro Kanal.

Vorschubsperrern Vorschubsperrern für **Achsen** werden jetzt über die Schnittstellen-Datenbausteine für die Achsen gesetzt. Sie werden innerhalb der Module über „Axis[x].iFeedStop“ und den DB3000 programmiert. Der Offset beträgt dabei pro Achse 50 Byte.

Adresse	Name	Kommentar
0.0	Axis [1] .iFeedStop	Vorschubsperrern auslösen (S)

Beispiel Die Vorschubsperrern einer Z-Achse wird mit „#GrobInterfaceAxis.iFeedStop“ und dem DB3000 programmiert (1). Die Anzeige erfolgt dabei nur, wenn der Fahrbefehl ansteht. Die Funktion „qTravelRequestActive“ gibt dann Rückmeldung über einen anstehenden Fahrbefehl (2).

☐ Netzwerk 16 : Z-Achse Vorschubsperrern: ZM-Achse ist nicht im Freigabebereich



7.2 Timer

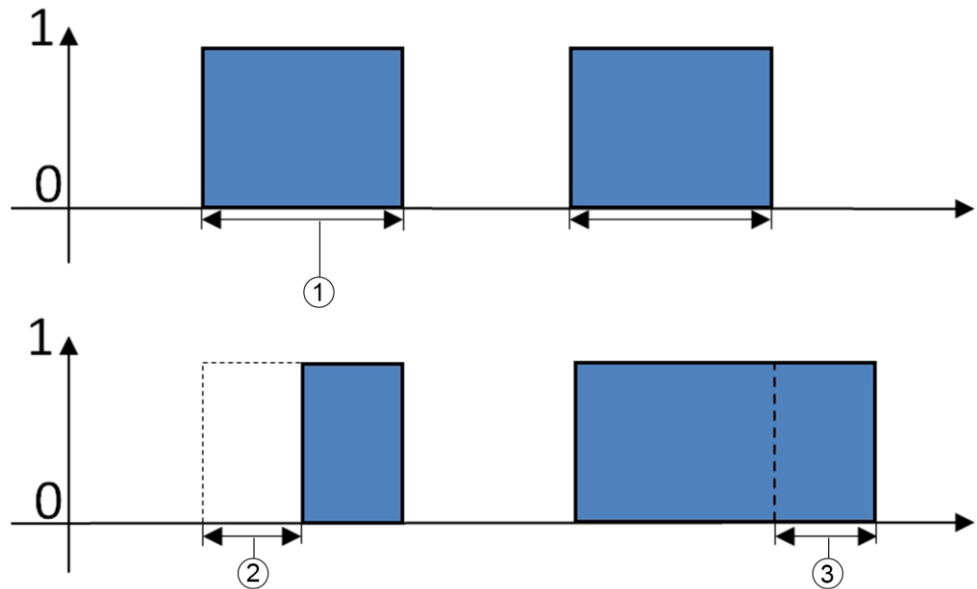


Hinweis

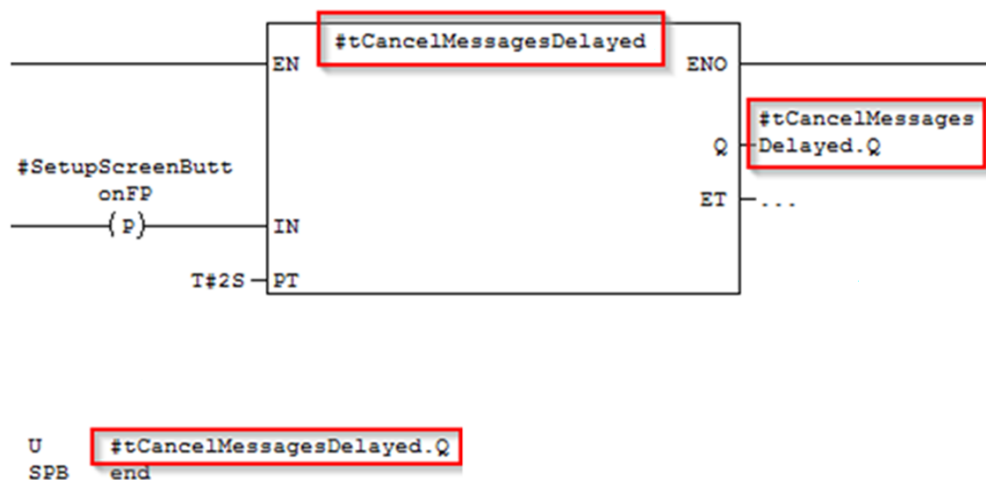
Timer werden in Baustufe 6 über die Siemens-Funktionsbausteine SFB 3-5 programmiert!

Timer (Zeitfunktionen) werden in Baustufe 6 meist über Siemens-Funktionsbausteine programmiert. Sie sind dadurch beliebig verfügbar. Die aus Baustufe 5 bekannten Timer werden nur im Grundprogramm weiter verwendet.

Programmierung Mit den Funktionsbausteinen SFB3 (1), SFB4 (2) und SFB5 (3) werden in Baustufe 6 die Timer programmiert. Pulse (kurze Zeitfenster) verwenden den Funktionsbaustein SFB3, Einschaltverzögerungen den Funktionsbaustein SFB 4 und Ausschaltverzögerungen den Funktionsbaustein SFB 5.



Beispiel Der Timer „#tCancelMessagesDelayed“ für die verzögerte Darstellung von „Abbrechen“-Meldungen (Cancel Messages) verwendet den SFB 4 für die Einschaltverzögerung.



Der Ausgang heißt dabei wie der Timer, wird aber durch ein „Q“ am Ende ergänzt. Im Programm kann daher die Signalquelle leichter gefunden werden.

Grundprogramm Im Grundprogramm bleiben einzelne Timer wie z.B. „t_MachineOn“ (Maschine eingeschaltet) bestehen.

t_MachineOn	T	12	TIMER	Maschine eingeschaltet
t_MachineOnPulse	T	13	TIMER	Maschine eingeschaltet positive Flanke
t_arbeit_ventile_ein	T	14	TIMER	Ventile im Bereich Arbeitsraum eingeschaltet verzögert
t_TechnicalFault	T	25	TIMER	Technische Störung verzögert
t_AlarmNcPresent	T	27	TIMER	NC-Alarm steht an verzögert

Baustufe 5 In Baustufe 5 wurden die Standard-Timer von Siemens verwendet. Diese sind allerdings begrenzt. Für Baustufe 6 wäre es schwierig, sie aufzuteilen und die Symbolik mitzuführen. Durch Verwendung der Timerbausteine muss die Symboltabelle nicht mehr angepasst werden.

Rev.

baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid

7.3 Merker

Merker (Zwischenspeicher) werden in Baustufe 6 durch Datenbausteine ersetzt. Sie sind dadurch beliebig verfügbar. Im Basiskonzept dienen sie als Platzhalter für reale Ein- und Ausgänge aus der Zuordnungsliste (ZULI). Im Grundprogramm werden Merker wie bisher verwendet.

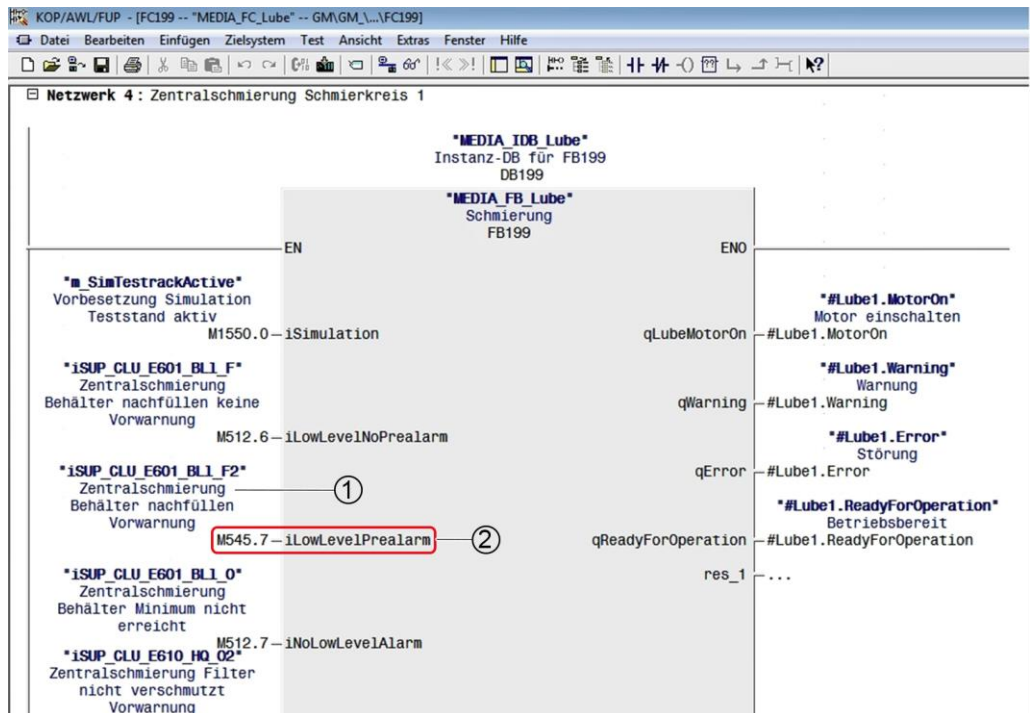
Grundprogramm Merker werden in Baustufe 6 nur noch im Grundprogramm oder als Platzhalter eingesetzt. Im remanenten Merkerbereich M0.0 – M36.7 sind einige Merker erhalten geblieben.

Beispiel Die Adresse „M 16.3“ steht für den Merker „mModeManual“ (Betriebsart Einrichten):

990	m_wkst entladen_aus	M 14.1	BOOL	Werkstück entladen (Taktart Test) ausgeschaltet
991	m_ModeAuto	M 16.0	BOOL	Betriebsart <Automatik> (Automatik Dauertakt)
992	m_ModeSingleCycle	M 16.1	BOOL	Betriebsart <Einzelbetrieb> (MDA oder Automatik Einzeltakt)
993	m_ModeSingleStep	M 16.2	BOOL	Betriebsart <Einzelschritt> (MDA oder Automatik Einzeltakt)
994	m_ModeManual	M 16.3	BOOL	Betriebsart <Einrichten>
995	m_AcknowledgeFault	M 16.6	BOOL	<Störung quittieren>
996	m_taktzeitueberschreitg	M 25.0	BOOL	Zentralfunktion: Taktzeitüberschreitung
997	m_PreselectionRunOut	M 28.0	BOOL	Zentralfunktion: Vorwahl Leerfahren an LIP
998	m_RunoutOffpHm	M 28.6	BOOL	Zentralfunktion Taktart Leerfahren Aus FP HM
999	m_CycleType1_StandAl...	M 30.0	BOOL	Taktart 1 Stand-Alone Betrieb
1000	m_CycleType2_RunOut	M 30.1	BOOL	Taktart 2 <Maschine Leerfahren>

Basiskonzept „Merker“ (Platzhalter) werden im Basiskonzept überall dort verwendet, wo der Maschine später ein Eingang oder Ausgang zugewiesen werden kann. Sie können somit in jedem FC oder FB auftauchen. In der Zuordnungsliste (ZULI) werden Merker mit größeren Nummern als 500 als Platzhalter eingesetzt.

Beispiel In der Basis sind in einer Zentralschmierung Eingänge wie z.B. der „Behälter nachfüllen Vorwarnung“ (1) über Merker wie z.B. M545.7 (2) für die Projektierung vorbereitet.

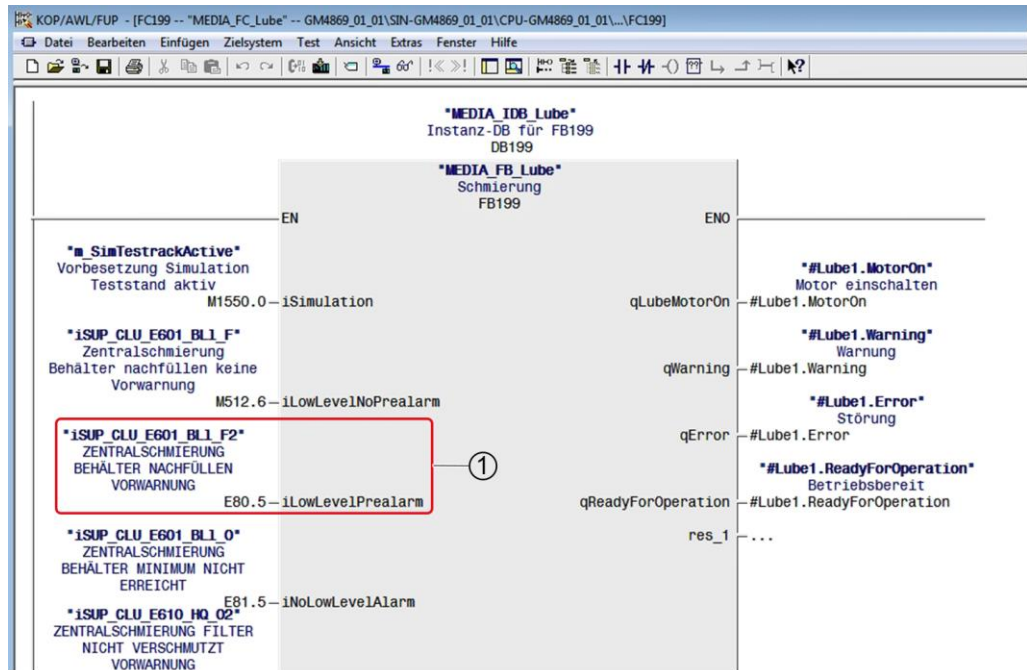


Rev. 1.0

baustufe 6 neuigkeiten 1.1.docx

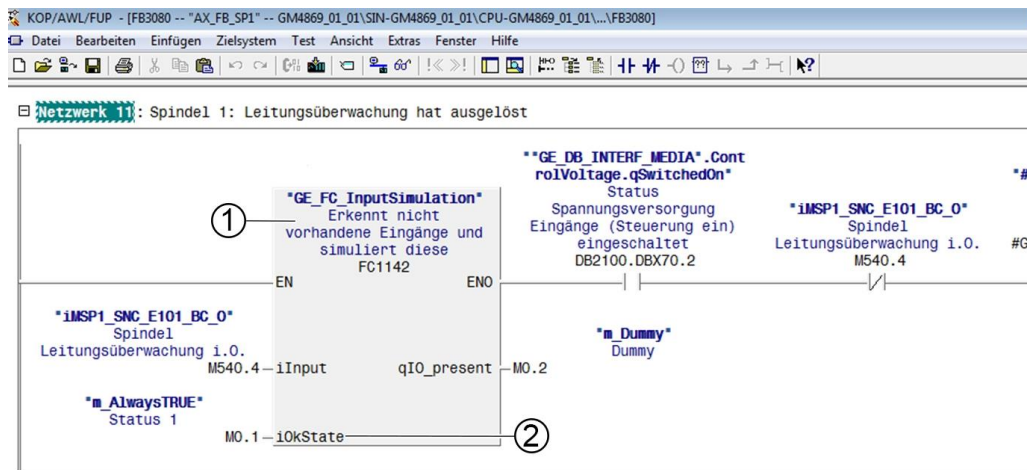
Hubert Schmid

Nach Erstellung der Maschine sind die Merker vorwiegend durch reale Ein- und Ausgänge ersetzt. Der Eingang „Behälter nachfüllen Vorwarnung“ ist z.B. dem Eingang „iLowLevelPrealarm“ (E80.5) zugeordnet (1).



Rev.

FC1142 Sind reale Ein- und Ausgänge im Modul für Zuordnungen nicht verfügbar, verbleibt der Merker als „Dummy“ bzw. Platzhalter. Zur Simulation nicht vorhandener Eingänge kann der Funktionsbaustein 1142 (FC1142) verwendet werden:



baustufe 6 neuigkeiten 1.1.docx

Der FC1142 (1) erkennt, ob es sich um einen realen Eingang oder den Dummy Merker handelt. Falls am Eingang der Dummy Merker programmiert ist, wird dieser automatisch mit "iOKState" beschrieben (2).

Baustufe 5 Merker wurden in Baustufe 5 viel verwendet. Ihr Umfang war begrenzt, und bei Erweiterungen gab es häufiger Schwierigkeiten. Oft waren keine Reserven vorhanden. Die Symbolik musste immer angepasst werden, und viele Merker waren durch Optionen belegt.

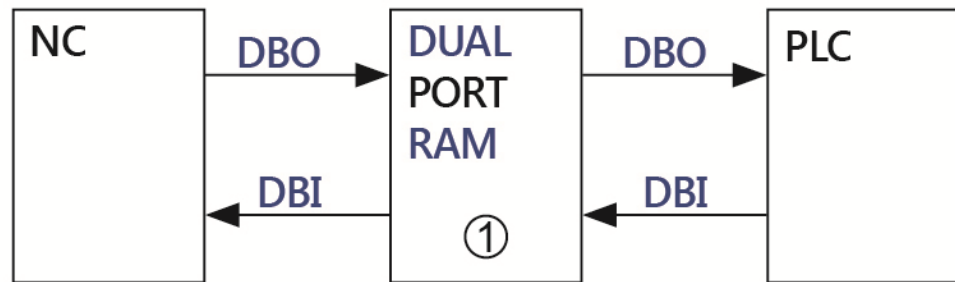
Hubert Schmid

8 Erweiterung schneller Datenkanal

In Baustufe 6 wurde der schnelle Datenkanal auf 4000 Byte erweitert. Die Übertragung zwischen NC und PLC erfolgt über 2 konsistente Blöcke in zwei Bausteinen mit zwischengeschaltetem Pufferdatenbaustein (Dualport RAM). Es gibt eine modulbezogene Aufteilung wie in den maschinenspezifischen Makros (MMAC). Diese Neustrukturierung erlaubt eine unabhängige Pflege und klare Bereichseinteilung der Funktionsmodule.

Schneller Datenkanal

Als schnellen Datenkanal (VDI-Schnittstelle) bezeichnet man die Schnittstelle zwischen der PLC und der NC. In Baustufe 6 ermöglicht sie Senden und Empfangen mit je 2000 Byte. Mit einem puffernden Dualport RAM (1) mit 4096 Byte werden die auszutauschenden Informationen, Daten und Signale zwischen NC und PLC übertragen:

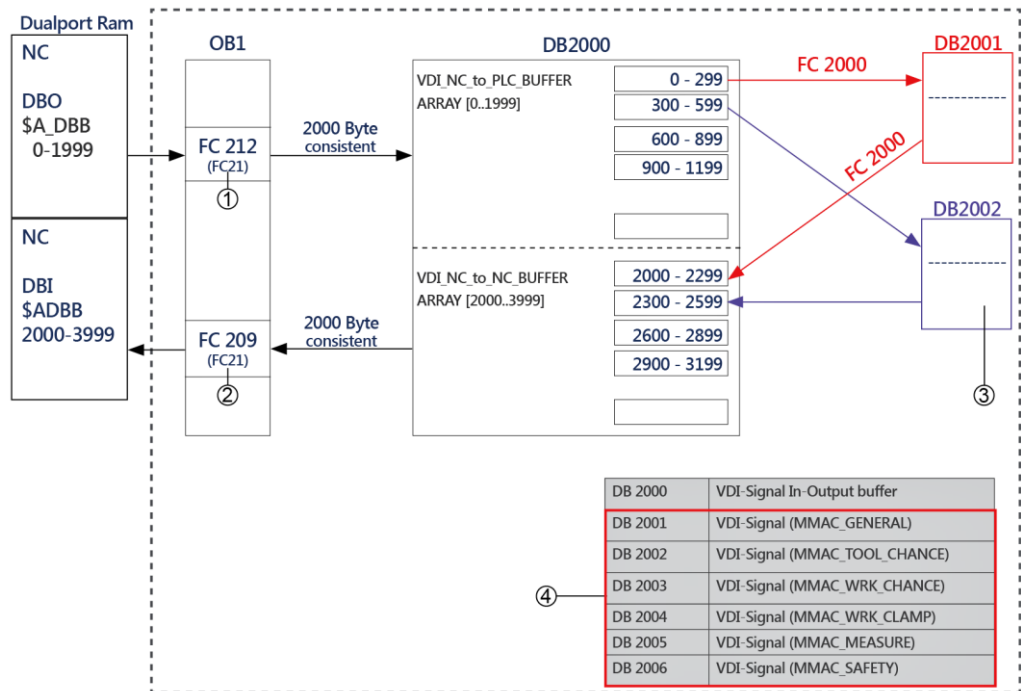


Die NC und die PLC greifen auf die gleiche Speicherzelle im Dualport RAM, den DB 2000, zu. Die PLC greift dabei direkt auf den DB 2000 zu. Die NC greift über die Systemvariable, \$A_DBB[...], zu.

DB2000

Der **DB2000** wird in Baustufe 6 als Pufferdatenbaustein verwendet. Er ermöglicht konsistentes Lesen und Schreiben der Datenblöcke. Symbolische Adressierung ist für ihn nicht möglich.

Schaltbild Folgende Abbildung zeigt ein Prinzipschaltbild einer VDI-Schnittstelle mit Dualport RAM:



Der FC212 (3) im Dualport RAM überträgt NC-Signale (DBO) und gibt sie konsistent zur PLC weiter. Der FC209 (2) im Dualport RAM empfängt konsistente PLC-Signale und gibt sie zur NC weiter (DBI). Der DB 2000 arbeitet als Pufferdatenspeicher.

Der Datenaustausch zwischen PLC und Dualport RAM findet mit Schnittstellen-DBs (1) statt, die die Daten direkt zum DB 2000 übertragen oder empfangen. Die Schnittstellen-DBs übertragen die Daten entsprechend der im Prinzipschaltbild enthaltenen Tabelle der Schnittstellen-DBs zu den einzelnen Modulen in der PLC. Damit hat jedes Modul einen eigenen Schnittstellen-DB (4).



Die Verwaltung vom DB2000 (Pufferdatenbaustein ohne Symbolik) befindet sich im FC2000!

Rev.

baustufe 6 neuigkeiten 1,1.docx

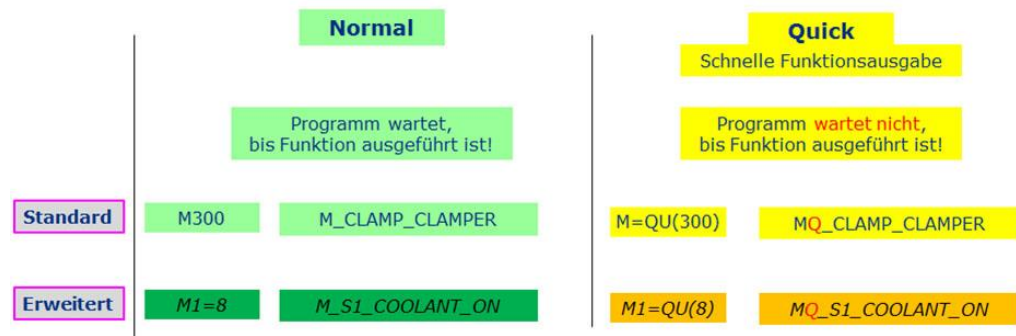
Hubert Schmid

9 M-Funktionsarten überarbeitet

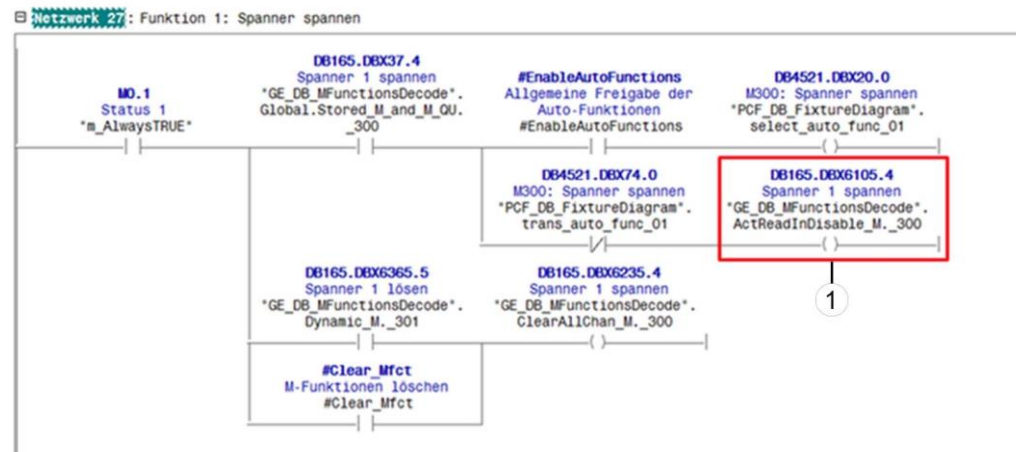
Ab Baustufe 6 wird zwischen „Normal“ und „Quick“ M-Funktionen unterschieden. M-Funktionen können als Standardfunktion oder als erweiterte Funktion verwendet werden:

- Normal: Eine „Normal“ M-Funktion hat die Eigenschaft, dass das Programm wartet bis die Funktion vollständig ausgeführt ist. Es besteht eine Einlesesperre.
- Quick: Eine „Quick“ M-Funktion hat die Eigenschaft, dass das Programm **nicht** wartet bis die Funktion vollständig ausgeführt ist. Es besteht **keine** Einlesesperre.

Übersicht



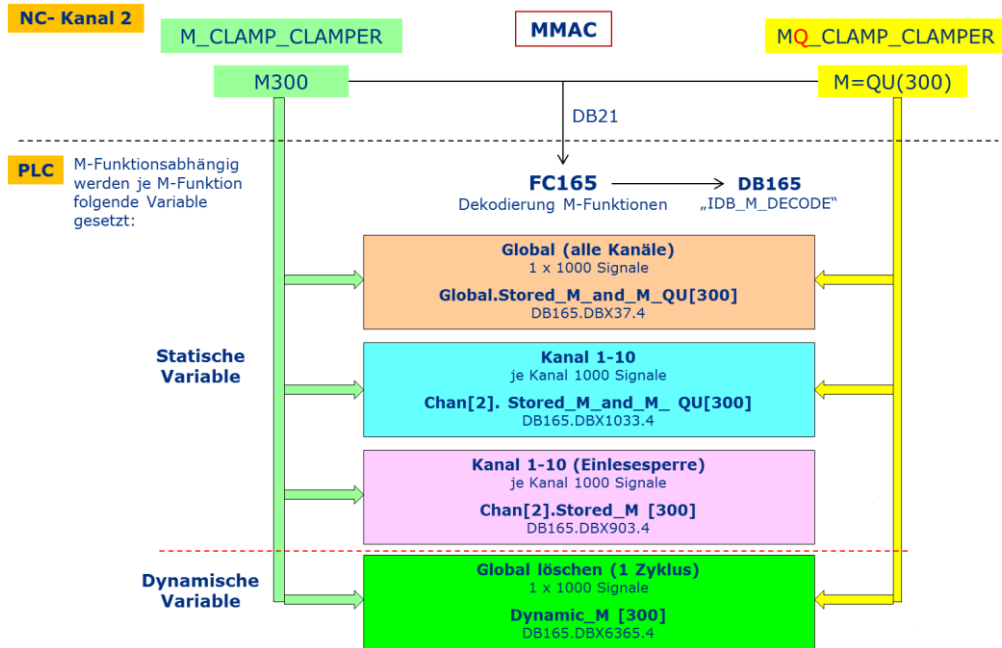
Beispiel mit Einlesesperre (1)



9.1 M-Funktionsarten im DB165

In der NC wird bereits durch die festgelegte M-Funktionsart unterschieden, welche Variable im DB 165 gesetzt werden und welche nicht. Mit dem Schnittstellen-DB21 werden die Signale dann von der NC an die PLC übertragen und im FC165 automatisch decodiert.

Übersicht



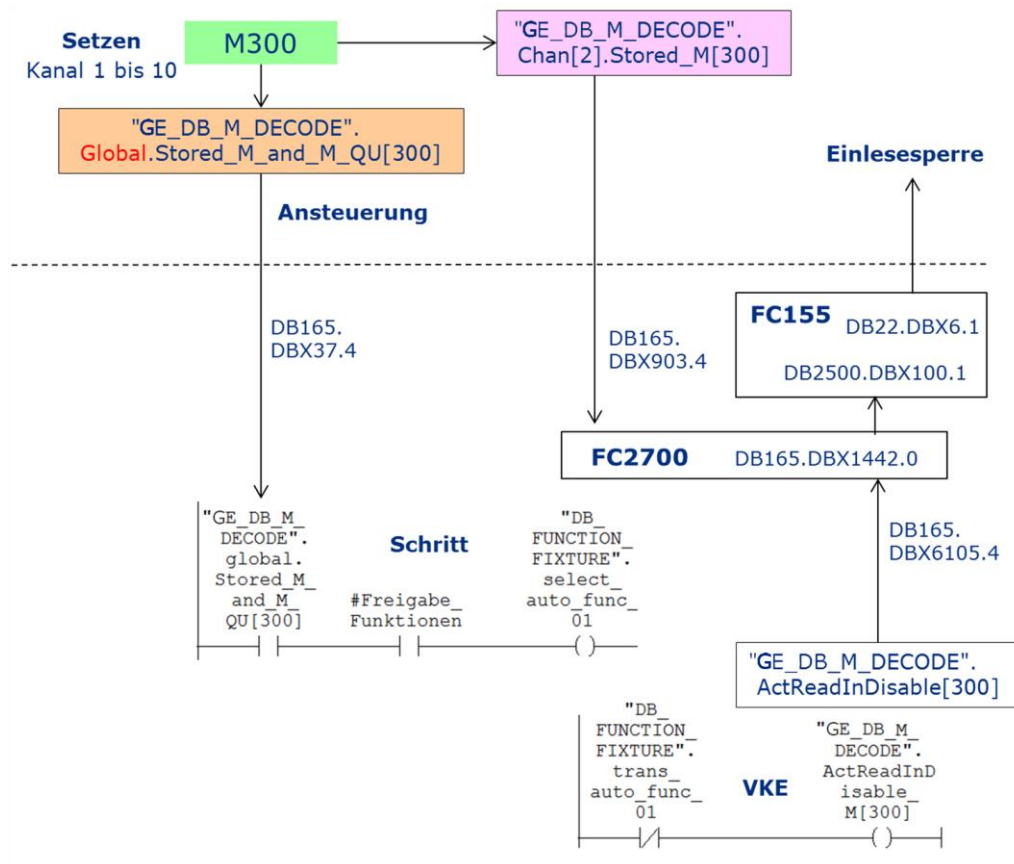
Statische Variable

Ausgehend von statischen Variablen, die von der Funktionsart abhängig sind, wird die M-Funktion ausgeführt. Die gesetzten Datenbits steuern dabei Einzelschritte der Funktion und/oder die integrierten Funktionsbausteine (FC2700/FC155) für die Einlesesperre an.



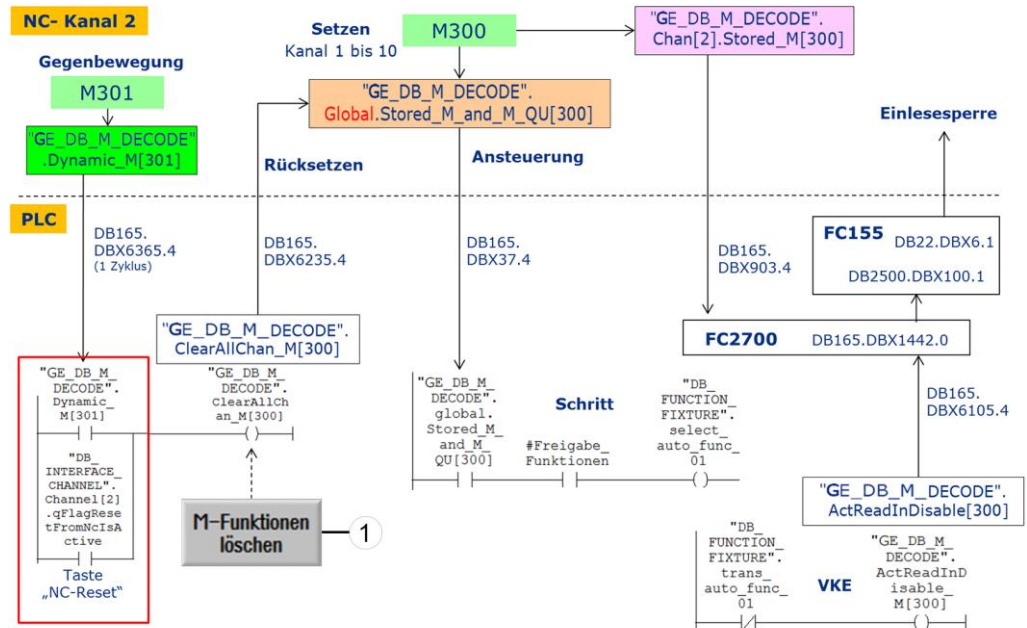
Hinweis

Einlesesperren wirken nur dann, wenn die zugehörige M-Funktion auch aktiv ist!



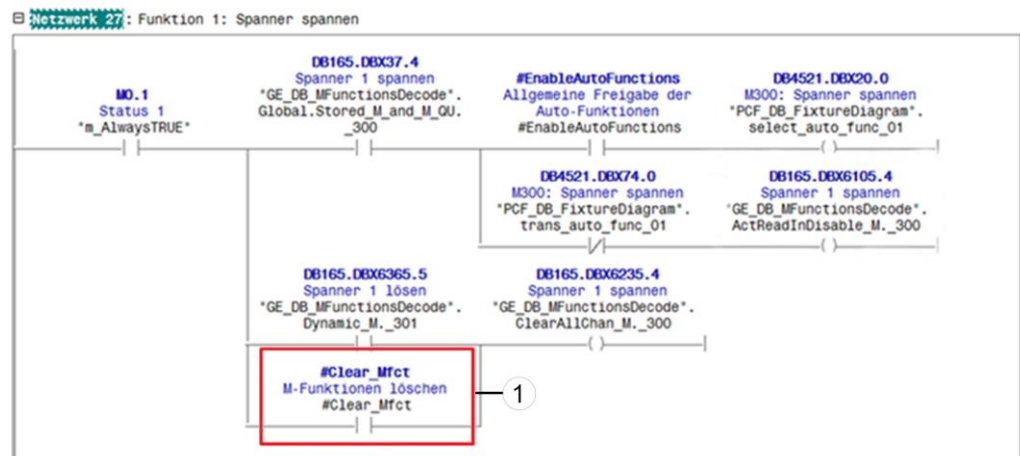
Dynamische Variable

Die dynamische Variable setzt die M-Funktionen zurück. Sie ist beiden M-Funktionsarten enthalten. Die Gegenbewegung wird ebenfalls im DB165 gespeichert und startet in der NC:



Durch Betätigen der Taste „NC-Reset“ und Bestätigen der Schaltfläche „M-Funktionen löschen“ (1) besteht die Möglichkeit, alle aktiven Variablen einer M-Funktion am HMI zurückzusetzen. Im Netzwerk und HMI-Bild findet sie sich wie folgt:

Netzwerk



Rev. 1.0

baustufe 6 neuigkeiten 1,1.docx

Hubert Schmid



Rev.

Beispiel

Je nach M-Funktionsart werden verschiedene Datenbits im DB165 gesetzt. Wenn beispielsweise die M-Funktion M7 im 1 Kanal aufgerufen wird, werden folgende Bits im DB165 gesetzt:

DB165.DBX	144.7	„GE_DB_MFunctionsDecode“.Global.Stored_M_and_M_QU_7
DB165.DBX	562.7	„GE_DB_MFunctionsDecode“.Chan[1].Stored_M_7
DB165.DBX	720.7	„GE_DB_MFunctionsDecode“. Chan[1].Stored_M_and_M_QU_7
DB165.DBX	6328.7	„GE_DB_MFunctionsDecode“.Dynamic_M_7

Wenn eine erweiterte M-Funktion M1=7 im Kanal 1 aufgerufen wird, werden folgende Bits im DB165 gesetzt:

DB165.DBX	144.7	„GE_DB_MFunctionsDecode“.Global.Stored_ExtM_and_ExtM_QU[1].Wert_7
DB165.DBX	562.7	„GE_DB_MFunctionsDecode“.Chan[1].Stored_ExtM[1]_7
DB165.DBX	720.7	„GE_DB_MFunctionsDecode“. Chan[1].Stored_ExtM_and_ExtM_QU[1].Wert_7
DB165.DBX	6328.7	„GE_DB_MFunctionsDecode“.Dynamic_M_7

Wenn eine „Quick“ M-Funktion M=QU(7) im Kanal 1 aufgerufen wird, wird das Kanalbit im DB165 **nicht** gesetzt:

DB165.DBX	0.7	„GE_DB_MFunctionsDecode“.Global.Stored_M_and_M_QU_7
DB165.DBX	418.7	„GE_DB_MFunctionsDecode“.Chan[1].Stored_M_and_M_QU_7
DB165.DBX	6328.7	„GE_DB_MFunctionsDecode“.Dynamic_M_7

Baustufe 5

In Baustufe 5 mussten die M-Funktionen in der Logik gespeichert werden (Speicherbereich). M-Funktionen waren auf den Kanal beschränkt. Für die Programmierung mit und ohne Einlesesperre waren zwei M-Funktionen nötig.

baustufe 6 neuigkeiten 1.1.docx

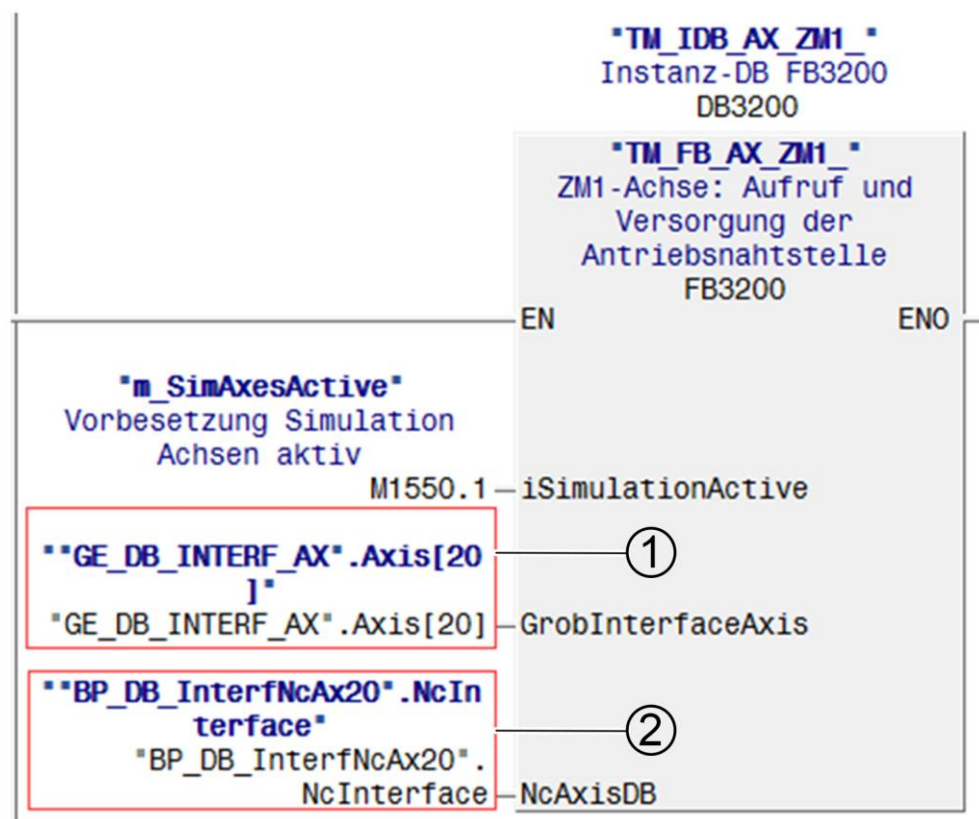
Hubert Schmid

10 Achs-Handling überarbeitet

In Baustufe 6 wurden die Achsen auf 31 gleichzeitig vorhandene NC-Achsen ausgebaut. Generelle Achsfunktionen sind jetzt in einem Standard-FB untergebracht. Für die Verwaltung von mehr als 31 Achsen in der Software wurde ein GROB-Achsinterface eingeführt. Achsbausteine sind damit übersichtlicher und schlanker und die Achs-Standardisierung bleibt.

Achsbaustein Für die Verwaltung der Achsen gibt es einen Standard-FB für Achsen. Jeder Achs-FB verwendet als Eingangsparameter jeweils einen ANY-Pointer für das GROB-Achsinterface (GrobInterfaceAxis) und den Standard-FB für Achsfunktionen (NCAxisDB):

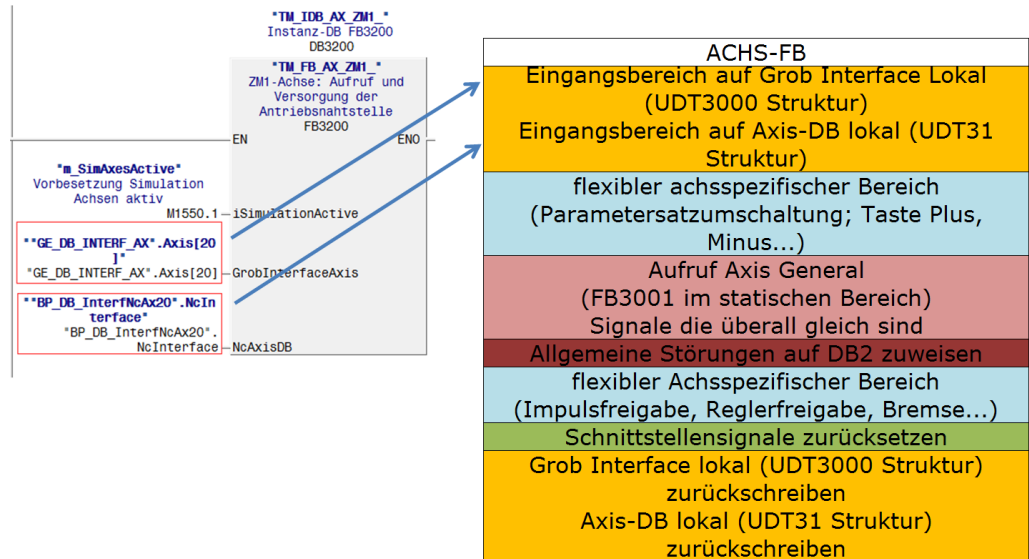
Netzwerk 1: Aufruf Achsbaustein



GrobInterfaceAxis (1) gibt dabei die Startadresse für die jeweilige Achsschnittstelle im DB3000 an. NcAxisDB (2) gibt die Startadresse für die gewünschte reale Schnittstelle zur Siemens - Achse (DB31 bis 61) an.

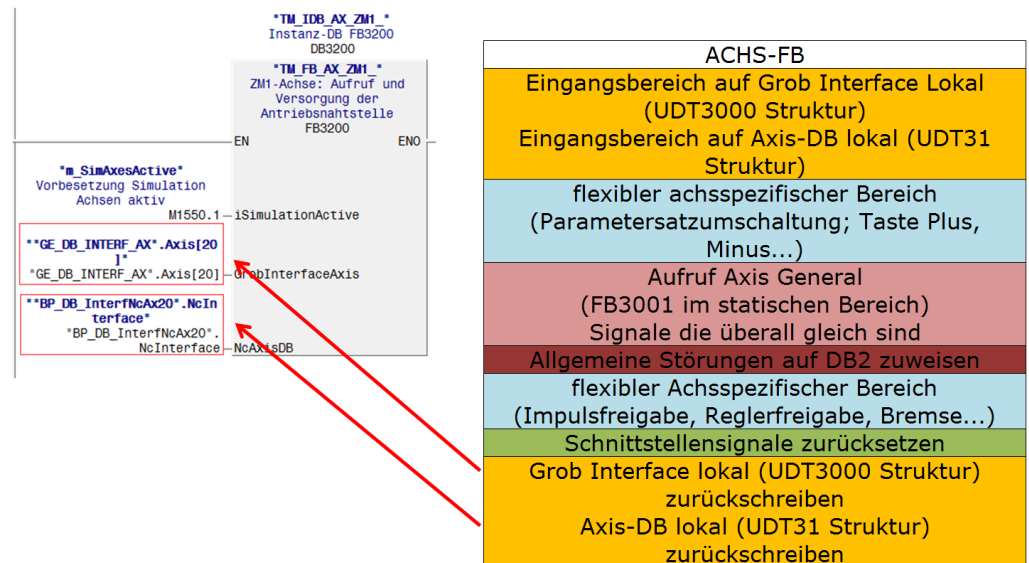
Intern Im Achs-FB werden die beiden Any-Pointer zuerst auf den lokalen Bereich umgeschrieben. Die Verarbeitung erfolgt über flexible achsspezifische Bereiche und einen generellen Bereich für achsübergreifende Signale und Störungen (Axis General):

Netzwerk 1: Aufruf Achsbaustein



Nach Verarbeitung werden die Schnittstellensignale zurück gesetzt und die Eingangsparameter „GrobInterfaceAxis“ und „NcAxisDb“ wieder in den globalen Bereich geschrieben:

Netzwerk 1: Aufruf Achsbaustein

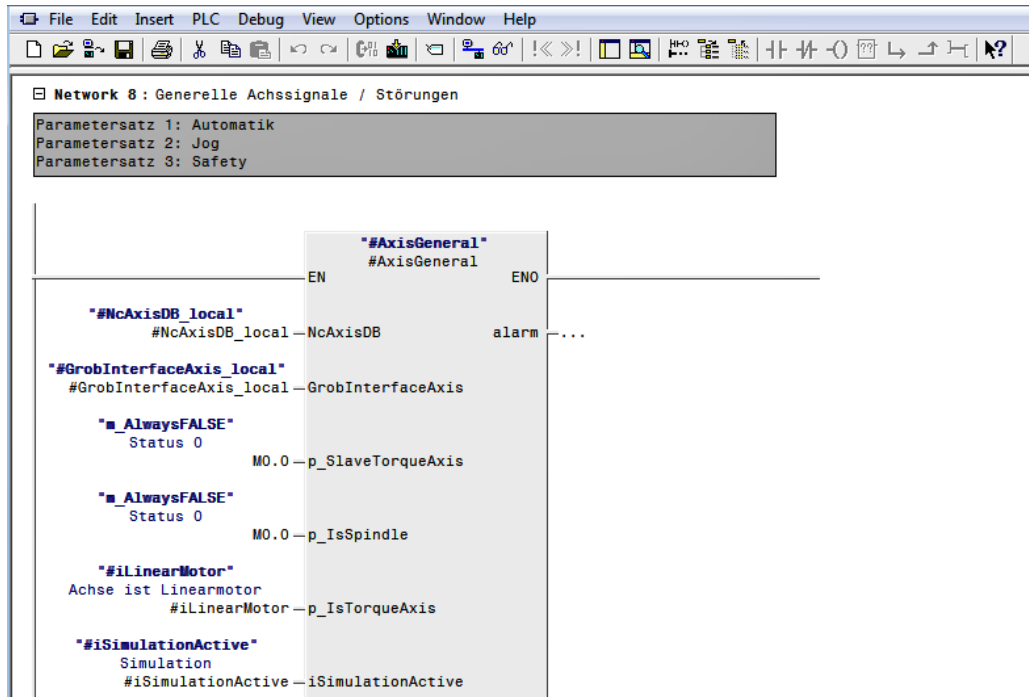


Rev.

baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid

Axis General Der Baustein „Axis General“ ist für überall gleiche Achssignale und Störungen zuständig. Er verwendet den FB30001 und weist Störungen auf den DB2 zu.



DB3000 Im DB3000 sind für 100 Achsen jeweils 50 Byte GROB-Achseninterface vorbereitet. Signale mit „ixxxx“ am Anfang und (S) im Kommentar, wie z.B iFeedStop (1), können überall gesetzt werden. Signale mit „qxxxxx“, wie z.B qAxisIsNotMoving (2), können überall abgefragt werden.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	iFeedStop	BOOL	FALSE	Vorschubsperrung auslösen (S)
+0.1	qEnableFeedStop	BOOL	FALSE	Erlaubnis für Vorschubsperrung auslösen
+0.2	iAxisIsDeactivated	BOOL	FALSE	Achse abgewartet von Bediener
+0.3	qAxisExists	BOOL	FALSE	Achse ist vorhanden
+0.4	iAxisToEnergySaveMode	BOOL	FALSE	Befehl an Achse: Achse in Energiesparmode schalten
+0.5	iTraversingKeyMinus	BOOL	FALSE	Verfahrtaste Minus von Extern (S)
+0.6	iTraversingKeyPlus	BOOL	FALSE	Verfahrtaste Plus von Extern (S)
+0.7	iControlDisableExtern	BOOL	FALSE	Reglerfreigabe ausschalten von extern (S)
+1.0	iSuppressFaultDriveReady	BOOL	FALSE	Störung Antrieb nicht Betriebsbereit unterdrücken bei
+1.1	iTeststop_running	BOOL	FALSE	Teststop läuft
+1.2	iSwLimit2Minus	BOOL	FALSE	2. Softwareendschalter wirksam minus von Extern (S)
+1.3	iSwLimit2Plus	BOOL	FALSE	2. Softwareendschalter wirksam plus von Extern (S)
+1.4	qTravelRequestActive	BOOL	FALSE	Achse hat Fahrbefehl
+1.5	qAxisIsNotMoving	BOOL	FALSE	Achse steht
+1.6	qAxisIsReadyForOperating	BOOL	FALSE	Achse ist betriebsbereit
+1.7	qAxisReferenced	BOOL	FALSE	Achse ist referenziert
+2.0	qAxInHomepos	BOOL	FALSE	Achse in Hompeos / Transportfreigabe
+2.1	qAxInExactCoarse	BOOL	FALSE	Achse in Position Grob
+2.2	qAxInExactFine	BOOL	FALSE	Achse in Position Fein

Baustufe 5 In Baustufe 5 sind Achsen immer knapp bemessen. Achslücken konnten aus Standardisierungsgründen nicht beliebig genutzt werden. Bei Änderungen, die alle Achsen betreffen, war der Pflegeaufwand hoch.

Rev. 1.0

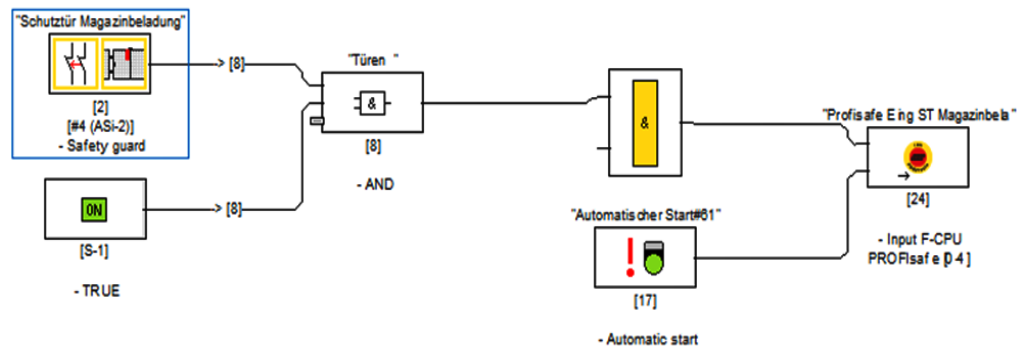
baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid

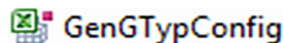
11 Safety neu strukturiert

In Baustufe 6 wurde der Bereich Sicherheit (Safety) neu strukturiert. Schutztüren sind jetzt separate Module und der sichere mechanische Bremsentest läuft antriebsintern ab. Die Programmierung wurde verändert. Die sichere Logik wird jetzt immer in SPL programmiert. Vorbesetzungen werden in Excel gesetzt bzw. verändert.

- Schutztüren** Schutztüren sind jetzt separate Module. Für jeden Schutztürtyp (egal ob mit oder ohne Zuhaltung) gibt es einen separaten Baustein. Er kann mehrfach aufgerufen werden.
- Bremsentest** Der sichere mechanische Bremsentest läuft nur noch antriebsintern ab. Er wird nicht mehr über den FC18 programmiert.
- Sichere Logik** Sichere Logik wird jetzt immer in SPL programmiert. Die SPL-Logik hat damit Bezug zu den Sicherheitsfunktionen. ASIMON wird nur noch als Gateway benötigt.
- ASIMON** In Baustufe 6 werden AS-i-Signale, soweit vorhanden, nur noch durchgeschleust. ASIMON wird nur noch als Netzübergang (Gateway) benötigt.



- Vorbesetzungen** Für die Erstellung einer Vorbesetzung wird zuerst die Datei „GenGTypConfig.xlsx“ in Excel geladen:

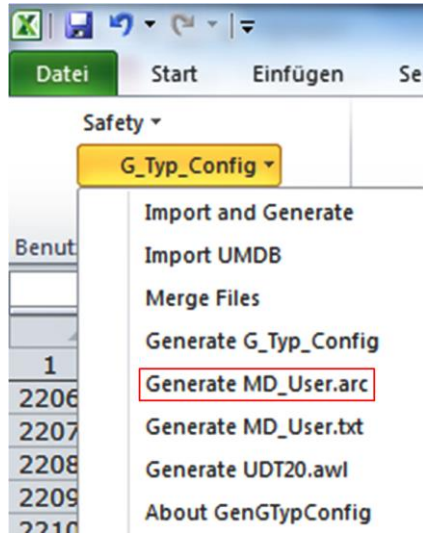


- Settingdatei** In der Excel-Tabelle werden dann Vorbesetzungen wie z.B. „Vorgezogene Teststoppwahl generell“ gesetzt oder verändert:

MD14512 USER_DATA_HEX [0-255]	Symbol	HEX	Binär	Kommentar
N14512 \$MN_USER_DATA_HEX[235]		0	00000000	
N14512 \$MN_USER_DATA_HEX[236] Bit 0	_236_BIT0		0	
N14512 \$MN_USER_DATA_HEX[236] Bit 1	_236_BIT1		0	
N14512 \$MN_USER_DATA_HEX[236] Bit 2	_236_BIT2		0	
N14512 \$MN_USER_DATA_HEX[236] Bit 3	_236_BIT3		0	
N14512 \$MN_USER_DATA_HEX[236] Bit 4	_236_BIT4		0	
N14512 \$MN_USER_DATA_HEX[236] Bit 5	_236_BIT5		0	
N14512 \$MN_USER_DATA_HEX[236] Bit 6	_236_BIT6		0	
N14512 \$MN_USER_DATA_HEX[236] Bit 7	_236_BIT7		0	
N14512 \$MN_USER_DATA_HEX[236]		0	00000000	
N14512 \$MN_USER_DATA_HEX[237] Bit 0	_237_BIT0		0	Vorgezogene Teststoppwahl generell
N14512 \$MN_USER_DATA_HEX[237] Bit 1	_237_BIT1		0	Vorgezogene Teststoppwahl nur bei Handbeladen
N14512 \$MN_USER_DATA_HEX[237] Bit 2	_237_BIT2		0	
N14512 \$MN_USER_DATA_HEX[237] Bit 3	_237_BIT3		0	
N14512 \$MN_USER_DATA_HEX[237] Bit 4	_237_BIT4		0	
N14512 \$MN_USER_DATA_HEX[237] Bit 5	_237_BIT5		0	
N14512 \$MN_USER_DATA_HEX[237] Bit 6	_237_BIT6		0	
N14512 \$MN_USER_DATA_HEX[237] Bit 7	_237_BIT7		0	
N14512 \$MN_USER_DATA_HEX[237]		0	00000000	
N14512 \$MN_USER_DATA_HEX[238] Bit 0	_238_BIT0		0	RTB Kupplung 1 Proportionalventil vorhanden
N14512 \$MN_USER_DATA_HEX[238] Bit 1	_238_BIT1		0	RTB Kupplung 1 mit OUTSE angesteuert
N14512 \$MN_USER_DATA_HEX[238] Bit 2	_238_BIT2		0	RTB Kupplung 2 Proportionalventil vorhanden
N14512 \$MN_USER_DATA_HEX[238] Bit 3	_238_BIT3		0	RTB Kupplung 2 mit OUTSE angesteuert

Über das programmierte Tool (Add-in) „G_Typ_Config“ wird jetzt über den Menüpunkt „Generate MD_User.arc“ eine einspielbare Settingdatei generiert:

Rev. 1.0



Bei dem Einspielen der Settingdatei in die Maschine werden die Userdaten mit den sicheren Vorbestzungen verglichen.

Fehlerprüfung

21.03.17 11:16:58.309		27090	Fehler bei kreuzw. Datenvergleich NCK-PLC, \$MN_SAFE_SPL_USER_DATA[0], NCK: 95220000H; Fehler bei Safety-SPL-IB-Status (DB18.DBX36.0)
--------------------------	--	-------	---



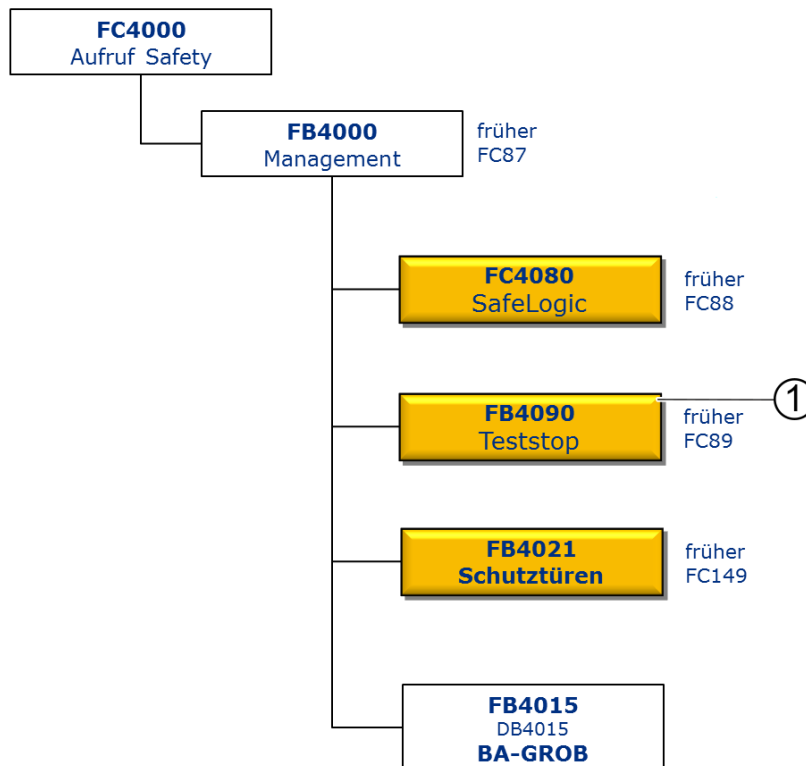
Hinweis

Falls Unterschiede bei unsicheren und sicheren Vorbesetzungen bestehen, erfolgt der KDV-Alarm!

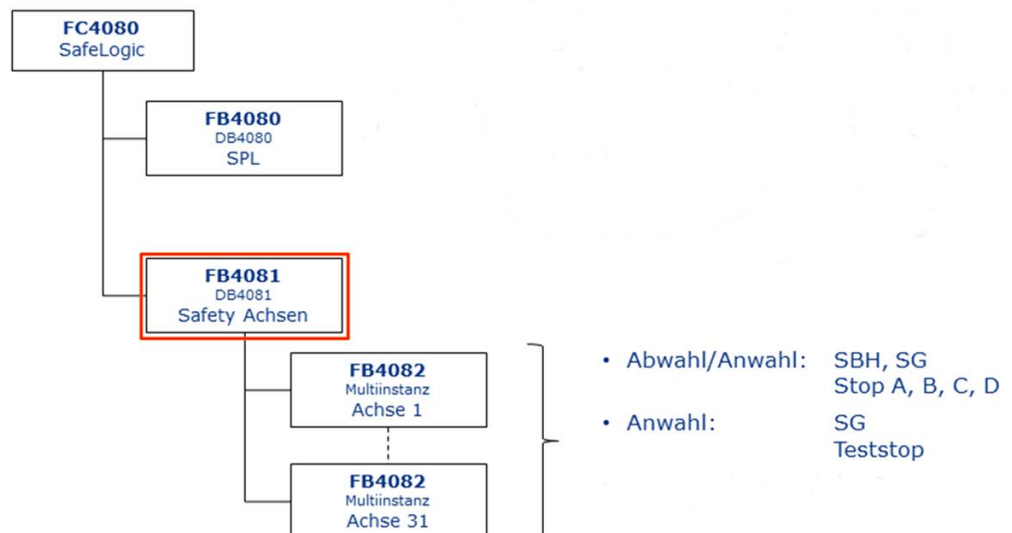
baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid

Bausteinstruktur Die Bausteinstruktur ist ähnlich Baustufe 5. Einzelne Funktionsbausteine wie z.B. der FB 4090, der Funktionsbaustein Teststop (1) sind dem Funktionsaufruf Safety (FC4000) untergeordnet:



Achsen Pro Achse gibt es einen Schnittstellenbaustein (FB4082). Im Schnittstellenbaustein werden die ausgehenden Signale für die Sicherheit (Outsi) gesetzt. Aufgerufen werden sie mit dem FB4081:

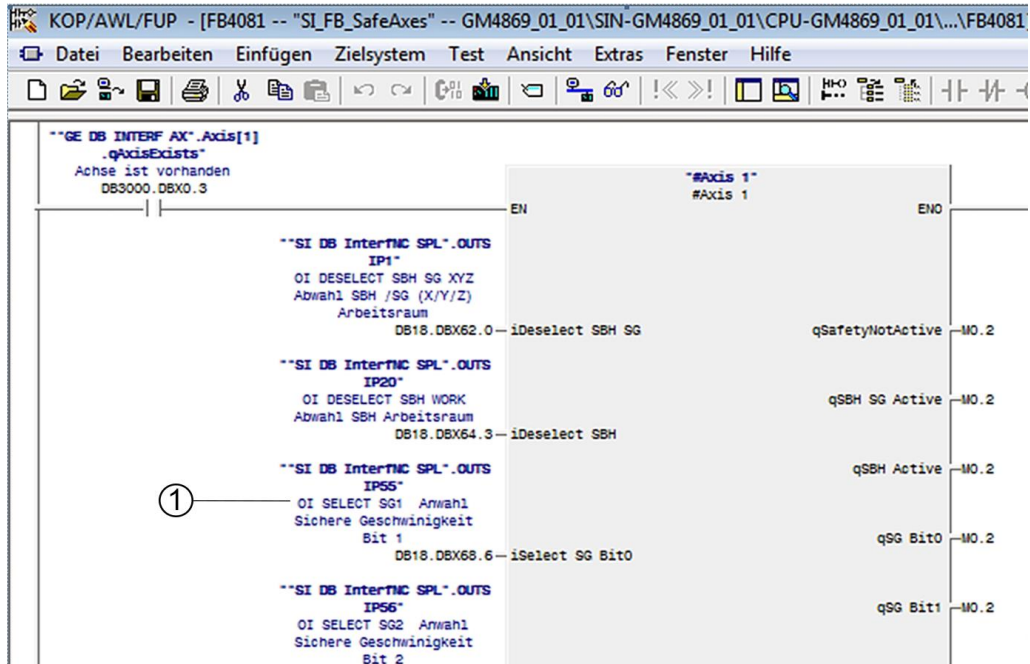


Rev.

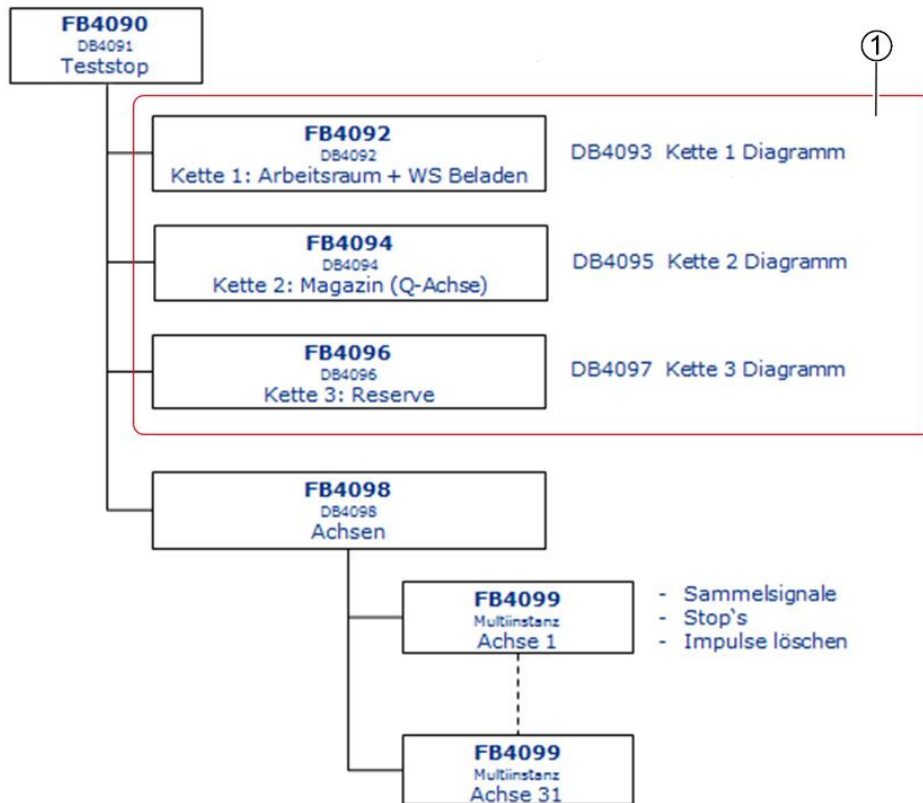
baustufe 6 neuigkeiten 1,1.docx

Hubert Schmid

Beispiel Nach Aufruf des FB4081 (Safety Achsen) werden in Kontaktplan für eine Instanz des FB4082 die Bits für die Anwahl der sicheren Geschwindigkeit gesetzt (1):



Teststops In Baustufe 6 gibt es, z.B. wegen Hauptzeit parallelem Beladen, bis zu drei Teststop-Ketten (1):

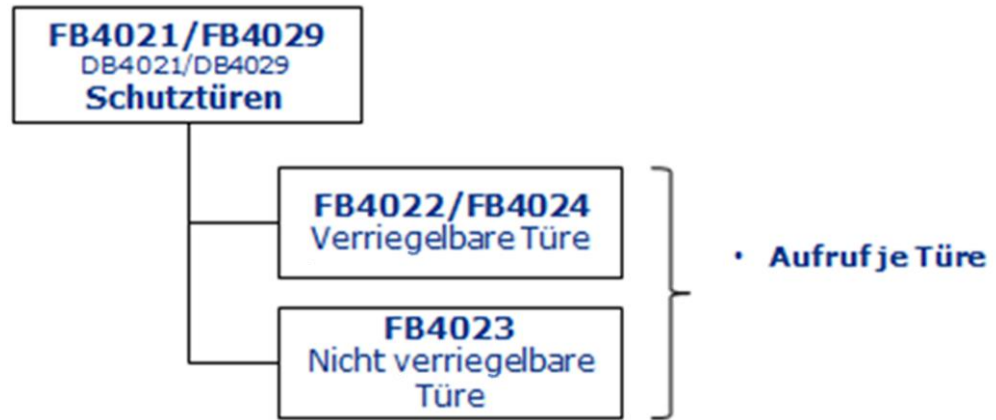


Rev. 1.0

baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid

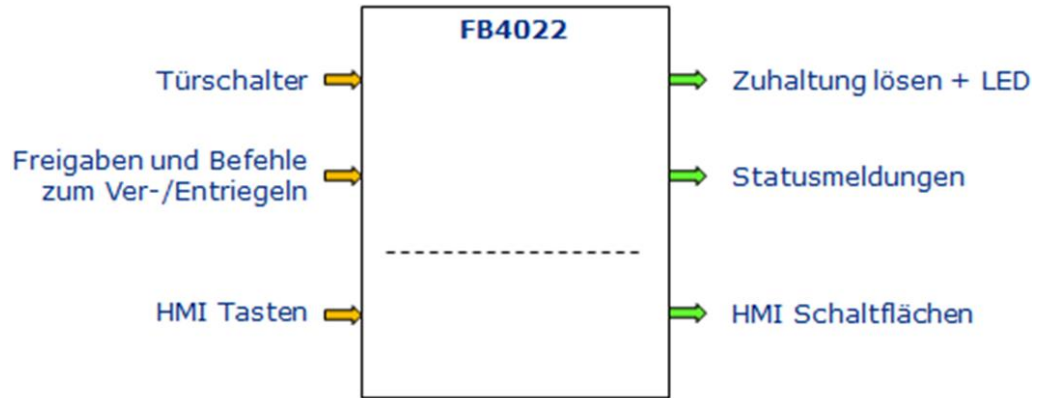
Schutztüren In Baustufe 6 gibt es für jeden Schutztürtyp einen eigenen Funktionsbaustein. Es gibt Funktionsbausteine für verriegelbare Türen und Funktionsbausteine für nicht verriegelbare Türen. Aufgerufen werden die Türen mit übergeordneten Funktionsbausteinen wie z.B. dem FB4029.



Ein Funktionsbaustein kann dadurch mehrfach instanziiert werden:

Name	Data Type	Address	Initial Value	Exclusion address	Termination address
SW_E101	FB 4022	0.0			
SW_E121	FB 4022	32.0			
SM_E121	FB 4022	64.0			
SM_E102	FB 4023	96.0			
STL_E101	FB 4022	128.0			
STL_E122	FB 4022	160.0			
ML_E206	FB 4023	192.0			
SM_E141	FB 4023	224.0			
SW_E232	FB 4023	256.0			

Beispiel Der FB4022 ist ein Funktionsbaustein für verriegelbare Türen. Schnittstellenparameter auf der Ausgangsseite sind u.a. LED und Zuhaltung lösen oder HMI Schaltflächen. Eingangsseitig sind Parameter für Türschalter, HMI Tasten und Freigaben und Befehle zum Ver- und Entriegeln vorhanden.



Hinweis

In Baustufe 6 werden die Rückmeldungen Stop D aktiv und SBH aktiv nicht mehr als Voraussetzungen zum Öffnen der Türe verwendet.

Zuhaltung

Um die Zuhaltung zu lösen, werden Bits bzw. Freigaben für Achsstillstand, Halt nach Taktende (H.n.T.) erreicht, Achsen geklemmt und Abnahmetest (Teststop) benötigt:



Anwahl
„Zuhaltung lösen“

Freigaben

- Achsstillstand
- H.n.T. erreicht
- Achsen geklemmt
- Abnahmetest i.O. (nach 8h)



Zuhaltung lösen



Nach dem Lösen der Zuhaltung geht die Maschine von Stop D auf den sicheren Betriebshalt (SBH) über.

Baustufe 5

Der Aufbau des Bereichs Sicherheit in Baustufe 5 war undurchsichtig. Der Anpassungsaufwand für die Schutztüren war hoch. Sicherheitsfunktionen waren zum Teil unzureichend und der Programmieraufwand für Anpassungen generell groß.

12 SCL-Bausteine



Wichtig

SCL-Quellen sollten während der Inbetriebnahme nicht gelöscht werden! Sie befinden sich im Unterverzeichnis „Quellen“ im Projektverzeichnis.



Hinweis

Nach Übersetzung können SCL-Bausteine im Editor nicht mehr geändert werden! Für ihre Änderung müssen die Quellen verändert und dann erneut kompiliert werden.

In Baustufe 6 ist S7-SCL (Structured Control Language) eine optionale Hochsprache. Sie verbindet PASCAL mit SPS und eignet sich besonders für komplexe Algorithmen, Prozessoptimierung und mathematische Funktionen.

Allgemein

SCL ermöglicht die Erstellung von FC-, FB-, UDT-, OB- und DB-Bausteinen. Programme können im SCL-Editor oder als ASCII-Quellen programmiert werden und beliebig viele Bausteine enthalten. Nach Programmierung werden sie übersetzt (kompiliert). Programmierte Bausteine können dabei mit Bausteinen aus anderen Sprachen kombiniert werden. Sie können auch in AWL rückübersetzt werden, sind dann aber nicht mehr in SCL zu ändern.

Programmierung

SCL-Programme werden im SCL-Editor oder als ASCII-Files erstellt und dann kompiliert. Vor der Kompilierung tragen sie die Endung „.scl“. Sie werden im Ordner „Quellen“ im Projektverzeichnis gespeichert:

Objektname	Symbolischer Name	Typ	Größe	Autor	Änderungsdatum
GMxxxx_MagZange 11_2_GR	---	AWL-Quelle	30662	---	06.03.2017 15:44:19
GMxxxx_MagZange 12_1_GR	---	AWL-Quelle	30614	---	06.03.2017 15:44:39
GMxxxx_MagZange 12_2_GR	---	AWL-Quelle	30659	---	06.03.2017 15:44:59
GMxxxx_RT8_Coupl_1_GR	---	AWL-Quelle	40335	---	06.03.2017 15:46:01
GMxxxx_RT8_Coupl_2_GR	---	AWL-Quelle	40395	---	06.03.2017 15:46:21
GMxxxx_RT8_Coupl_3_GR	---	AWL-Quelle	40335	---	06.03.2017 15:46:42
GMxxxx_RT8_Coupl_4_GR	---	AWL-Quelle	40335	---	06.03.2017 15:46:57
GMxxxx_RT8_Coupl_5_GR	---	AWL-Quelle	40335	---	06.03.2017 15:47:17
GMxxxx_RT8_Coupl_6_GR	---	AWL-Quelle	40335	---	06.03.2017 15:47:37
GMxxxx_RT8_Coupl_7_GR	---	AWL-Quelle	40335	---	06.03.2017 15:47:52
GMxxxx_RT8_Coupl_8_GR	---	AWL-Quelle	40335	---	06.03.2017 15:48:13
GMxxxx_SpindelZonus_bla	---	AWL-Quelle	27808	---	06.03.2017 15:49:39
GMxxxx_SpindelZonus_GR	---	AWL-Quelle	27567	---	06.03.2017 15:49:59
GMxxxx_SpindelZange_GR	---	AWL-Quelle	32207	---	06.03.2017 15:50:20
GMxxxx_SpindelZonus_blas	---	AWL-Quelle	27808	---	06.03.2017 15:50:40
GMxxxx_Spindelpos_GR	---	AWL-Quelle	27616	---	06.03.2017 15:51:01
GMxxxx_Spindelzange_GR	---	AWL-Quelle	32192	---	06.03.2017 15:51:22
GMxxxx_SST2_Coupl_5_nur	---	AWL-Quelle	40475	---	06.03.2017 15:51:42
GMxxxx_SST2_Coupl_6_nur	---	AWL-Quelle	40475	---	06.03.2017 14:56:05
GMxxxx_SST_Coupl_1_GR	---	AWL-Quelle	40334	---	06.03.2017 15:52:16
GMxxxx_SST_Coupl_2_GR	---	AWL-Quelle	40334	---	06.03.2017 15:52:38
GMxxxx_SST_Coupl_3_GR	---	AWL-Quelle	40389	---	06.03.2017 15:53:00
GMxxxx_SST_Coupl_4_GR	---	AWL-Quelle	40334	---	06.03.2017 15:53:23
GMxxxx_SST_Coupl_5_GR	---	AWL-Quelle	40326	---	06.03.2017 15:53:46
GMxxxx_SST_Coupl_6_GR	---	AWL-Quelle	40320	---	06.03.2017 15:54:12
GMxxxx_SST_Coupl_7_GR	---	AWL-Quelle	40399	---	06.03.2017 15:54:37
GMxxxx_SST_Coupl_8_GR	---	AWL-Quelle	40327	---	06.03.2017 15:55:02
GMxxxx_Toolcheck1_GR	---	AWL-Quelle	29736	---	06.03.2017 15:45:21
GMxxxx_Toolcheck2_GR	---	AWL-Quelle	29736	---	06.03.2017 15:45:42
GMxxxx_yy_zz_SI_FD_Q_Axi	---	AWL-Quelle	137330	---	06.03.2017 15:48:33
GMxxxx_yy_zz_SI_FD_Setup	---	AWL-Quelle	136370	---	06.03.2017 15:48:55
GMxxxx_yy_zz_SI_FD_Work_	---	AWL-Quelle	137340	---	06.03.2017 15:49:16
Gx2x_Kuehlmittel1_2Spin	---	AWL-Quelle	79965	---	06.03.2017 15:40:08
Gxxx_G_Typ_Config	---	AWL-Quelle	60607	---	01.03.2017 13:19:35
MFunctionsDecode_DB165_G	---	AWL-Quelle	261983	---	01.03.2017 11:27:53
S7SourceAla	---	AWL-Quelle	2741...	---	08.02.2017 14:59:20
UDT118_GR	---	AWL-Quelle	119411	---	06.03.2017 15:02:08
UDT3005_GR_Softwarenocke	---	AWL-Quelle	6243	---	01.03.2017 11:27:29
UDT3521_GR_Softwarenocke	---	AWL-Quelle	7316	---	01.03.2017 11:27:34
UDT5031_GR_Softwarenocke	---	AWL-Quelle	5288	---	01.03.2017 11:27:39
UDT Alarmer GR	---	AWL-Quelle	937074	---	01.03.2017 13:25:26
SCL_AX_FC_Deselect	---	SCL-Quelle	3793	---	24.08.2015 10:22:24
SCL_GE_M_DECODE	---	SCL-Quelle	5910	---	11.07.2016 09:26:54
SCL_GE_M_DECODE_test	---	SCL-Quelle	5946	---	06.12.2016 14:42:56
SCL_HMI_JOG_Line	---	SCL-Quelle	8041	---	24.08.2015 10:24:30
SCL_HMI_VariableAlarms	---	SCL-Quelle	7079	---	25.05.2016 13:42:40

Sie sind im „Simatic Manager“ mit dem Typ „SCL-Quelle“ gekennzeichnet.

Löschung Werden SCL-Quellen aus dem S7-Quellen-Ordner gelöscht, lassen sich die Bausteine im SCL-Editor nicht mehr öffnen. In diesem Fall erscheint im KOP/AWL/FUP-Editor ein Warnhinweis und der Programmierer öffnet den SCL-Baustein als AWL-Netzwerk:

The screenshot shows the SCL editor interface for FC3002. A warning message is displayed: "compiled by SCL compiler version: SCLCOMP K05.03.07.00_01.02.00.01 release". Below the warning, a network diagram is shown with the following code:

```

SET
SAVE
= L 2.1
CLR
= #qWarnOneOrMoreAxisDeSe #qWarnOneOrMoreAxisDeSe -- Warnung: Eine oder mehrere Achsen sind
= #qChangeInKeyword #qChangeInKeyword -- Eine Abwahl hat sich geändert
= #qError #qError
L 1
T #1 #1
A7d0: L #1 #1
L "BP_IDB_GetParameter".MaxAxis DB7.DBW170
<=I
SPBN A7d1
L #1 #1
ITD
L L#1391
+D
I AR1
    
```

Eine weitere Bearbeitung ist in SCL dann so nicht mehr möglich (die Quellen müssen neu importiert und übersetzt werden).

Syntax SCL-Programme müssen nach syntaktischen Regeln ähnlich denen von PASCAL (typischer Aufbau von Funktionsbausteinen, Groß- und Kleinschreibung unterschieden, Semikolon nach Anweisung) erstellt werden. Sowohl symbolische als auch absolute Adressen können verwendet werden.

The screenshot shows the SCL editor interface with the following code snippet:

```

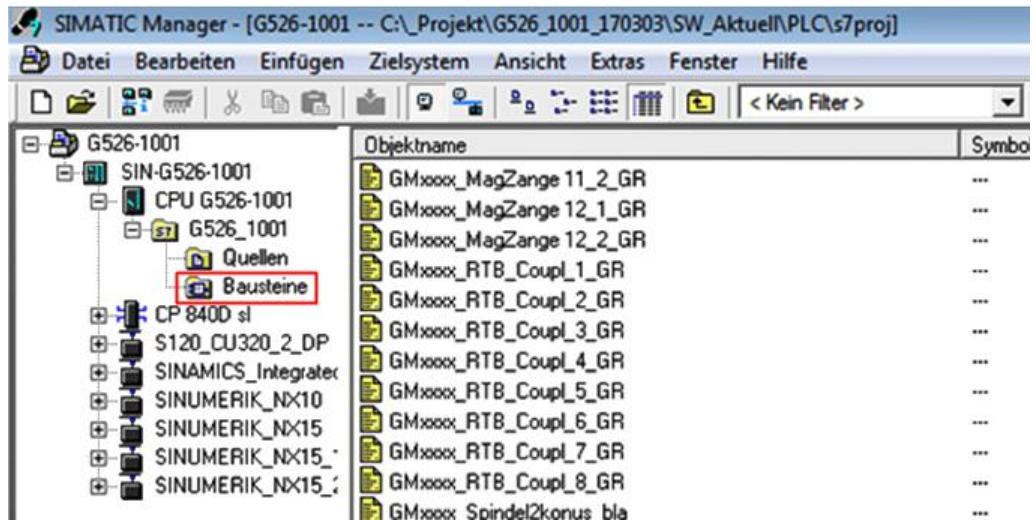
IF WDB.L_RESET THEN
    L_RESET_OS := TRUE;
    WDB.L_RESET := FALSE;
END_IF;

IF L_RESET_OS OR G_RESET OR NOT LATCH THEN
    RESET := TRUE;
ELSE
    RESET := FALSE;
END_IF;
    
```



In SCL können Absolutadressen mit der Programmoption „Gehe zur Verwendungsstelle“ nicht direkt angesprungen werden!

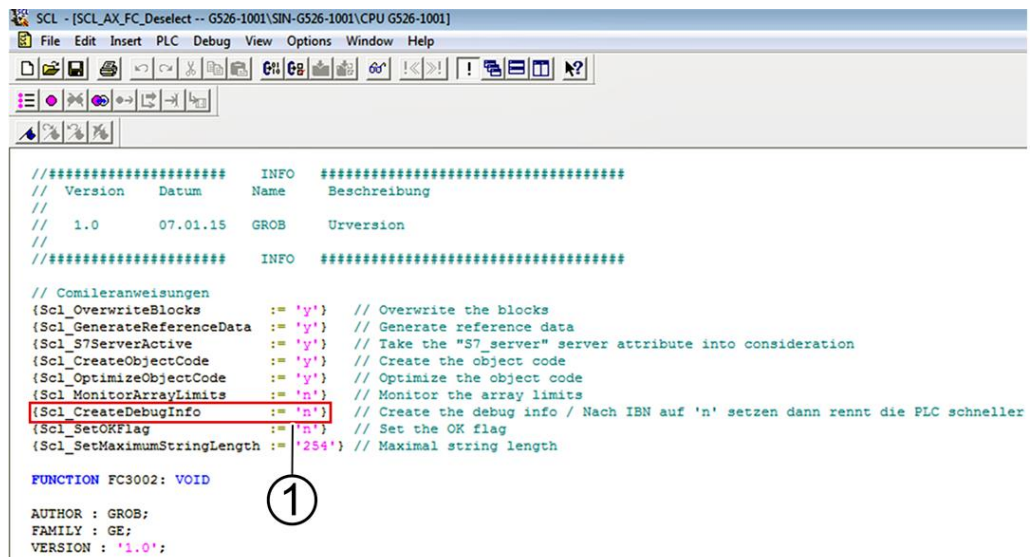
Kompilieren Ein SCL-Programm muss vor der Verwendung übersetzt (kompiliert) werden. Dazu wird der Compiler verwendet. Die Quelle wird damit in ein AWL-Netzwerk übersetzt und im Ordner „Bausteine“ des S7-Programmes gespeichert.



Zum Einspielen werden SCL- Bausteine im SIMATIC-Manager mit dem Befehl „Datei>Laden“ auf die CPU geladen.

Beobachten Syntaktische Fehler in SCL werden vom Kompiler angezeigt, Laufzeitfehler durch Systemalarme bei Programmausführung. Logische Fehler können nach der Kompilierung durch Einzelschritt-Testläufe oder durch Beobachten ermittelt werden.

Um ein Beobachten von Bausteinen zu ermöglichen, muss die Einstellung "Create Debug Info" im Header der SCL-Quelle aktiviert werden.



Dazu rechts der Zeile „Create_Debug_Info“ ein „y“ eintragen (1).



Beobachten von Bausteinen ist nur möglich, wenn alle zugehörigen Bausteine online aktuell sind!

13 Multiinstanzen



Wichtig

Zum Beobachten im Aufrufpfad muss der Baustein aktuell in die Steuerung geladen oder im Online-Projekt beobachtet werden!



Hinweis

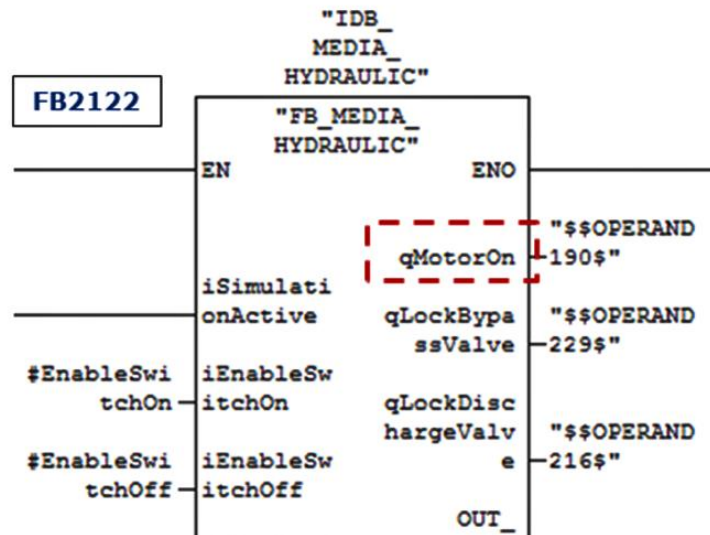
Der Testbetrieb schaltet sich beim Schließen des KOP-Fensters wieder selbständig ab.

13.1 Instanz und Multiinstanz

Wird in Baustufe 6 ein Funktionsbaustein direkt aufgerufen, wird ihm direkt ein Interface-DB zugeordnet. Beim Aufruf mehrerer Instanzen eines Funktionsbausteins werden separate Instanzen mit unterschiedlichen Namen im statischen Bereich des Aufrufbausteins definiert.

Einzelaufruf Wird ein FB einmalig aufgerufen, kann dem FB ein Interface-DB zugeordnet werden. In diesem Fall wird das Eingangssignal von der Schnittstelle direkt im Baustein verarbeitet. Danach wird mit dem Schnittstellensignal das Ausgangsbit beschaltet.

Beispiel Um eine Rückmeldung über den Status eines Motors für die Hydraulik zu erhalten, wird das Ausgangsbit „qMotorOn“ des FB 2122 (FB Media Hydraulik) mit einem symbolisch adressierten Datenbit beschaltet.



Zuerst wird innerhalb des Bausteins die Eingangsschnittstelle abgefragt und verknüpft. Sind alle Bedingungen erfüllt wird das Schnittstellenbit qOrderOn gesetzt.

Dieses wird wiederum am Ende des Bausteins dem OutParameter zugewiesen.

```

U   "DB_GE_INTERF_MEDIA".Hydraulic.iSwitchOn
U   #iEnableSwitchOn
U   "DB_GE_INTERF_MEDIA".Hydraulic.qReadyForOperation
S   "DB_GE_INTERF_MEDIA".Hydraulic.qOrderOn
    
```

Der Interface-Datenbaustein (DB_GE_INTERF_MEDIA) wird dabei dem FB2122 direkt zugeordnet. Es erfolgt kein Aufruf über den temporären Bereich.

```

U   "DB_GE_INTERF_MEDIA".Hydraulic.iSwitchOn
U   #iEnableSwitchOn
U   "DB GE INTERF MEDIA".Hydraulic.qReadyForOperation
S   "DB_GE_INTERF_MEDIA".Hydraulic.qOrderOn
    
```

Danach wird mit dem Schnittstellensignal das Ausgangsbit beschaltet:

```

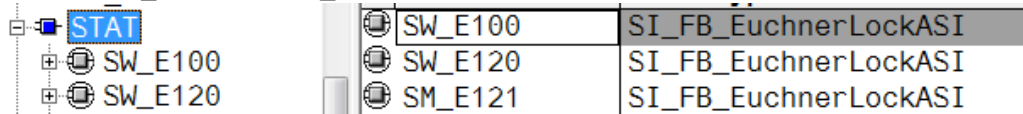
U   "DB GE INTERF MEDIA".Hydraulic.qOrderOn
=   #qMotorOn
    
```

Mehrfachaufruf

Bei Mehrfachaufruf eines Funktionsbausteines wird zunächst das Interface vom Eingangs-Any-Pointer P##ixxxxxx in den temporären Bereich (Temp) geladen. Danach wird es im Baustein verarbeitet und auf den Temp-Ausgangsbereich #qxxxxxx geschrieben. Am Ende des Bausteins wird der Temp-Ausgangsbereich wieder zurück auf den Bereich des Eingangs-Any-Pointer geladen.

Beispiel

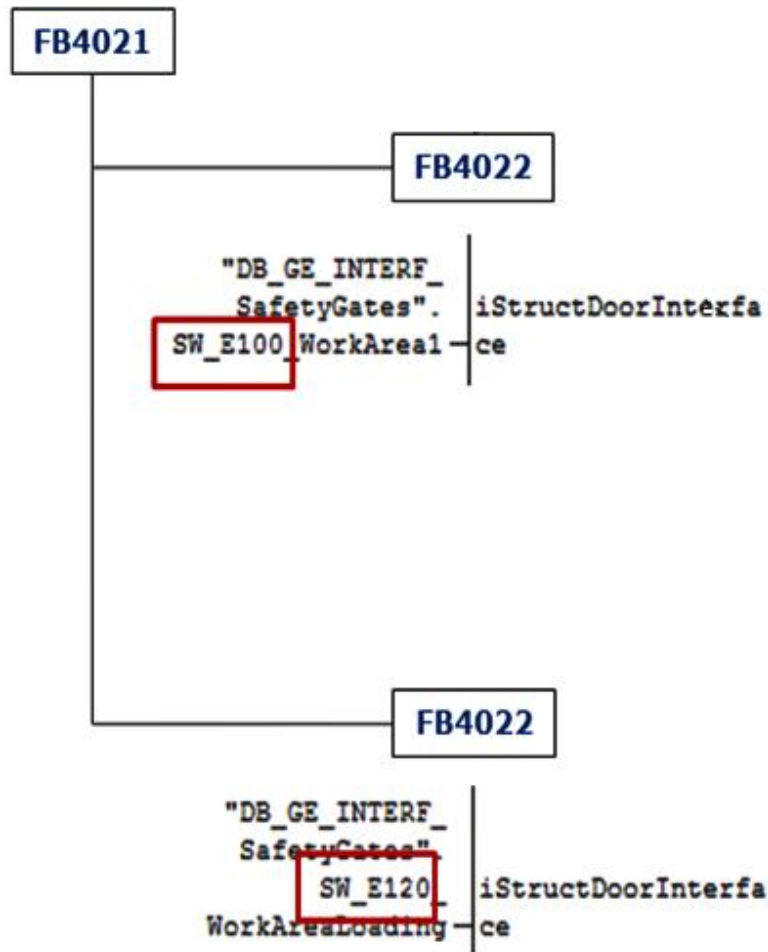
Im FB4021 werden zwei Funktionsbausteine (FB4022) für verriegelbare Schutztüren programmiert. Für beide wird eine Instanz des GE-Interfaces „DB_GE_INTERF_SafetyGates“ aufgerufen. Sie werden über die statischen Aufrufe SW_E100 und SW_E120 adressiert.



Rev.

baustufe 6 neuigkeiten 1.1.docx

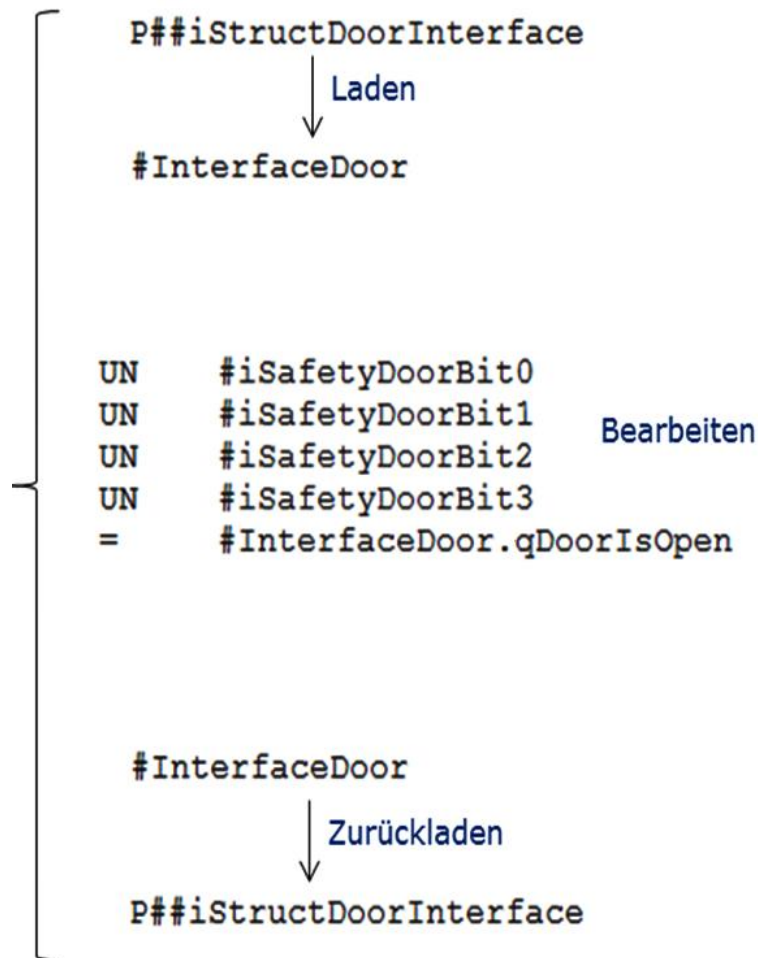
Hubert Schmid



Für das folgende Beispiel wird das Bit „qDoorIsOpen“ (Schutztüren geöffnet) verwendet.



Beide Funktionsbausteine verwenden den gleichen Interface-DB „DB_GE_INTERF_SafetyGates“.

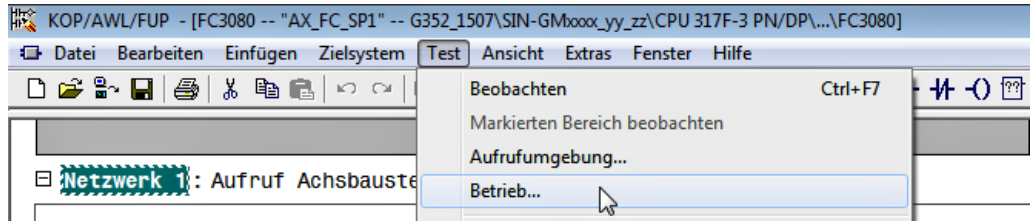


Das Interface wird vom Eingangs-Any-Pointer (P##iStructDoorInterface) in den temporären Bereich #InterfaceDoor geladen (1). Es wird im Baustein bearbeitet und anschließend auf InterfaceDoor geschrieben (2). Anschließend wird es mit Hilfe des Pointers zurück geladen (3).

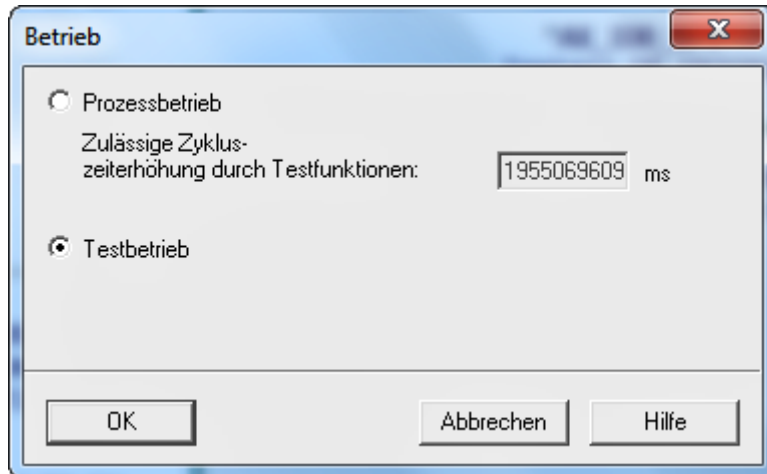
13.2 Testbetrieb aktivieren

Multiinstanzen eines Bausteins können im KOP/AWL/FUP-Editor direkt beobachtet werden. Dazu wird zuerst der Testbetrieb aktiviert. Über den Befehl „Beobachten im Aufrufpfad“ kann dann die gewünschte Instanz des Bausteins online oder offline (in der CPU) beobachtet werden

- Baustein**
1. Projekt in die Steuerung laden und gewünschten Baustein öffnen.
 2. Im Menü „Test“ den Menüpunkt „Betrieb“ wählen.



3. Im darauf erscheinenden Optionsfeld „Betrieb“ die Option „Testbetrieb“ auswählen:



4. Mit „OK“ bestätigen.
→ Der Testbetrieb ist aktiviert.

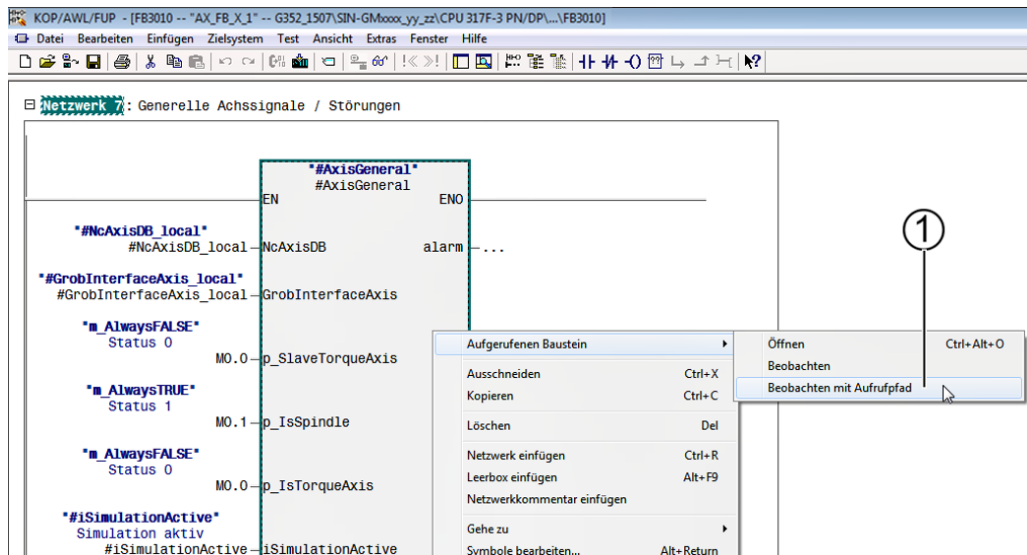
Rev. 1.0

baustufe 6 neuigkeiten 1.1.docx

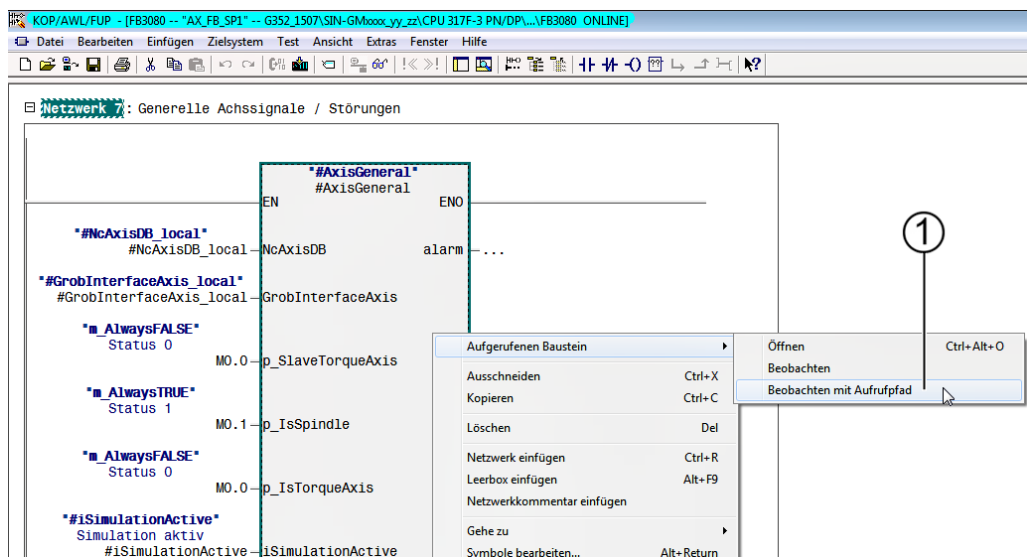
Hubert Schmid

13.3 Bausteine beobachten

- CPU**
1. Bausteine speichern und wieder laden.
 2. Mauszeiger oder Cursor auf die gewünschte Aufrufanweisung positionieren (Call-Zeile in AWL oder Box des Bausteins in KOP/FUP).
 3. Mit der rechten Maustaste auf die Aufrufanweisung klicken .
 4. Aus dem Kontextmenü den Befehl "Aufgerufenen Baustein > Beobachten im Aufrufpfad" auswählen.



- Online**
1. Online-Projekt öffnen.
 2. Mauszeiger oder Cursor auf die gewünschte Aufrufanweisung positionieren (Call-Zeile in AWL oder Box des Bausteins in KOP/FUP).
 3. Mit der rechten Maustaste auf die Aufrufanweisung klicken.
 4. Aus dem Kontextmenü den Befehl "Aufgerufenen Baustein > Beobachten im Aufrufpfad" auswählen.

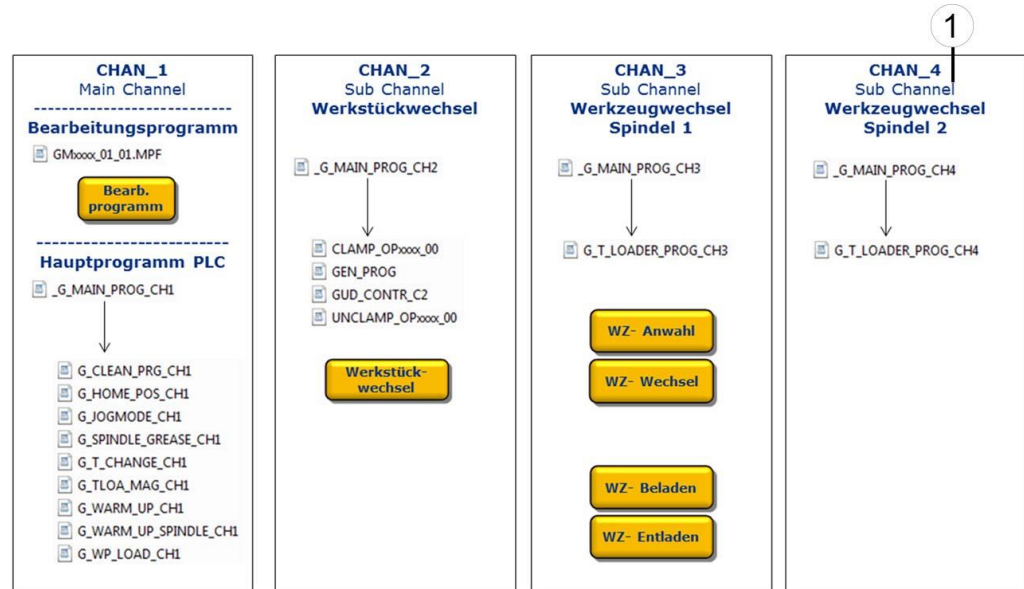


14 Kanalstruktur Baustufe 6

Für Baustufe 6 wurde die Kanalstruktur für G-Module überarbeitet. In jedem Kanal laufen Hauptprogramme, die Unterprogramme aufrufen. Programme für den Werkzeugwechsel an Spindel 2 laufen immer in Kanal 4.

Spindel 2

In Baustufe 6 laufen Programme für den Werkzeugwechsel an Spindel 2 immer im Kanal 4 (1).

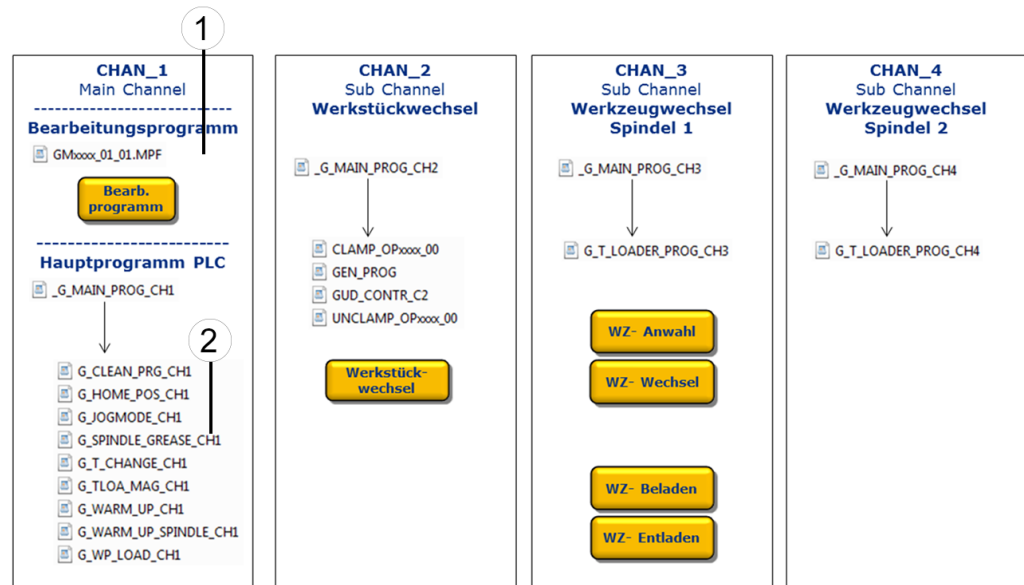


StartManagement

In Baustufe 6 haben sowohl Unterkanäle (Sub Channel) als auch der Hauptkanal (Main Channel) eigene Main-Programme (G_MAIN). Jeder Kanal ruft dabei eigene Unterprogramme auf.

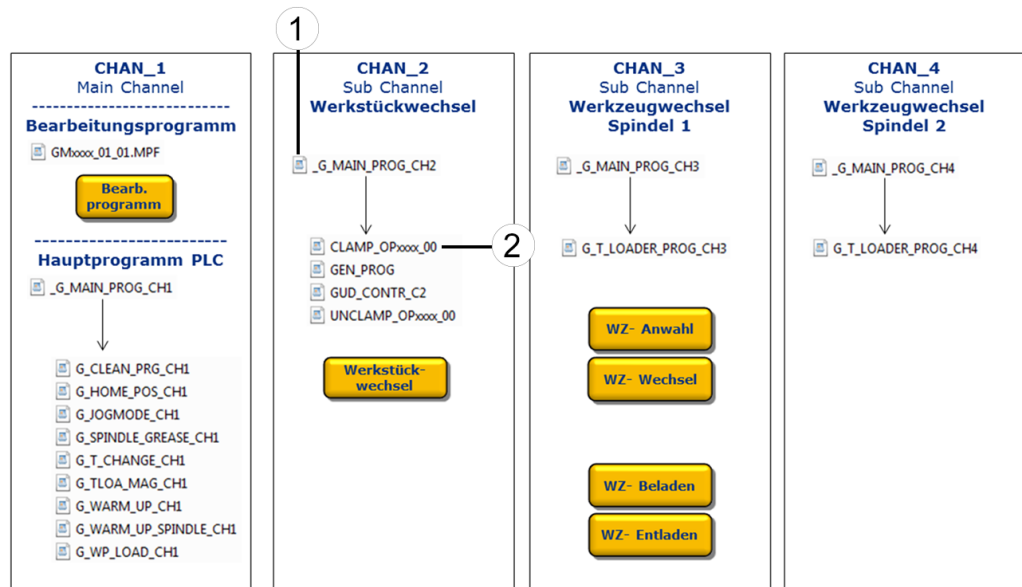
Beispiel

In Kanal 1 (Main Channel) läuft das Hauptprogramm „G_MAIN_PROG_CH1“ (1). Das Hauptprogramm ruft Unterprogramme wie z.B das Unterprogramm „G_SPINDLE_GREASE_CH1“ (2) auf.



Auch Kanal 2 hat ein eigenes Main-Program (1). Es ruft seinerseits Unterprogramme

wie z.B. CLAMP_OPxxxx_00 (2) auf.



Rev.

Baustufe 5 In Baustufe 5 wurden die Sonderprogramme separat über die PLC aufgerufen. Es gab Schleifenprogramme nur im Kanal 2 und 3.

baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid

15 Grundstellungskonzept



Hinweis

In Baustufe 6 beinhaltet die M-Funktion M25 eine Einleesperre!

In Baustufe 6 wird die Grundstellung aufgeteilt in Grundstellung gesamt und Grundstellung für Wiederanlauf (Grundstellung Arbeitsraum). M25 hat in Baustufe 6 eine Einleesperre. Kanal 1 regelt über Auftrag 99 die komplette Grundstellungsfahrt der Maschine.

PLC Die Gesamt-Grundstellung wird in der PLC über CH_FB_ManageHompos (FB2610) verwaltet.

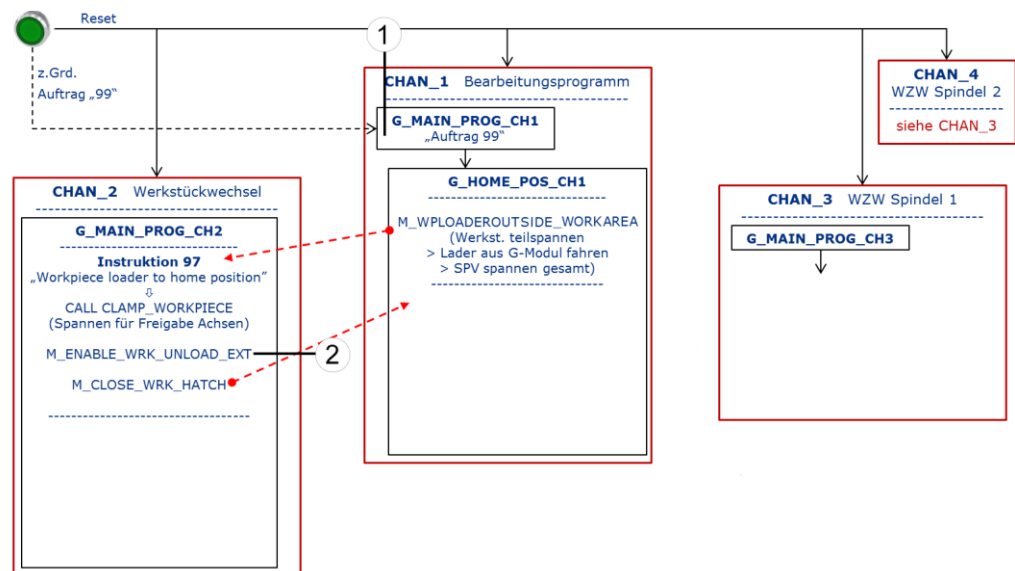
M25 Die M-Funktion M25 hat in der Baustufe 6 eine Einleesperre.

Ablauf Schematisch wird die Grundstellungsfahrt wie folgt programmiert:

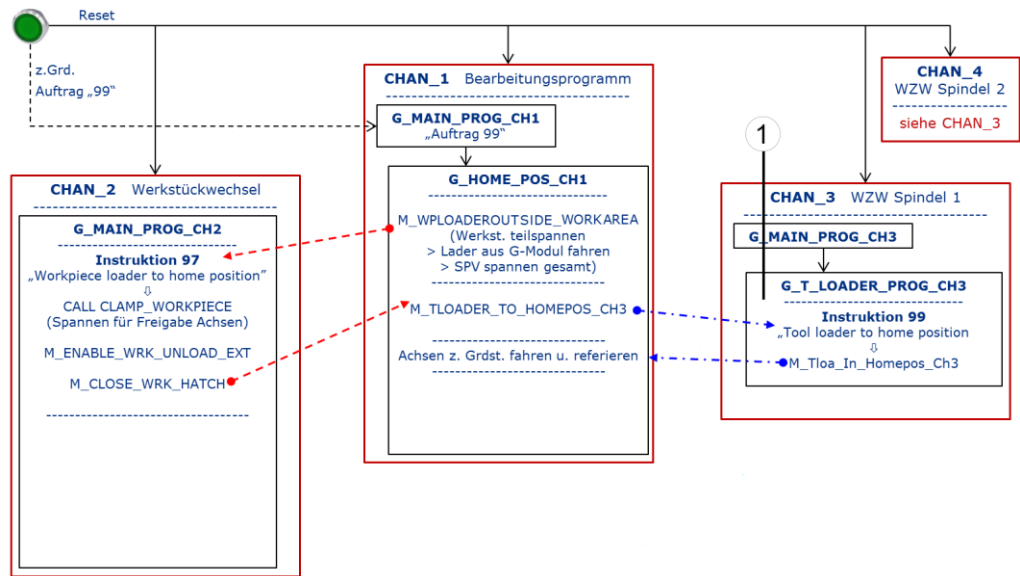
M_WPLoaderOutsideWorkarea
M_ClampFixtureAxisEnable
M_TLOA_TO_HOMEPOS_CH3
Eigene Achsen Fahren
M_ClampAndRefFixture
M25

Grundstellung Die Positionierung in die Grundstellung wird mit Auftrag „99“ an Kanal 1 ausgelöst. Kanal 1 sendet dann Instruktionen an Unterkanäle. Nach Abarbeiten der Instruktionen wird in Kanal 1 mit der M-Funktion M25 auf Grundstellung gefahren.

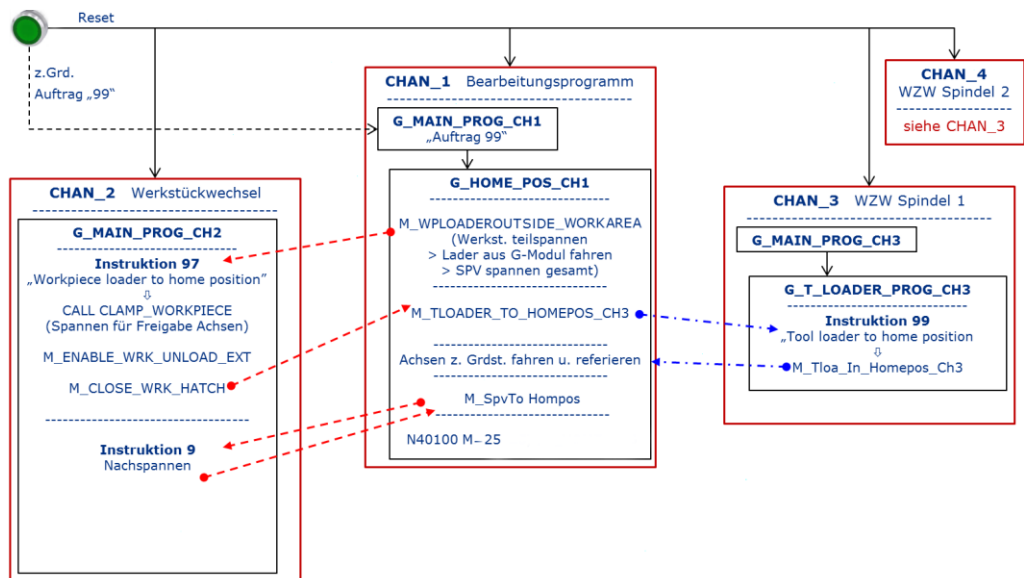
Beispiel An Kanal 1 wird Auftrag 99 (zur Grundstellung) gesendet (1). Kanal 1 sendet jetzt „Workpiece loader to home position“ (M_WPLoaderOutside_Workarea) an Kanal 2 (2).



Nachdem der Befehl in Kanal 2 abgearbeitet wurde, sendet Kanal 1 den Befehl „Tool loader to home position“ (M_TLOADER_TO_HOMEPOS) an Kanal 3 (1).



Danach sendet Kanal 1 den Befehl zum Nachspannen (M-SPVTO HOMEPOS) an Kanal 2 (1). Wenn nachgespannt wurde, wird in der M25 (1) gewartet, bis die Grundstellung erreicht ist.



Baustufe 5 In Baustufe 5 war die Grundstellung unterteilt in:

- Grundstellung NC
- Grundstellung asynchrone Abläufe
- Grundstellung Lift

16 Kundenspezifische Netzwerke (SPEC)

In Baustufe 6 erfolgen kundenspezifische Anpassungen über vier vordefinierte Bausteine. Damit müssen die Standardbausteine nicht verändert werden und die Updatefähigkeit ist gesichert.



Verwendung Die Verwendung der Bausteine richtet sich nach dem gewünschten Aufrufzeitpunkt in Operationsbaustein 1 (OB1):

FC1550	SPEC_FC_SpecEquipment1	Aufruf Sonderausrüstung nach Grundprogramm/Datenempfang
FC1551	SPEC_FC_SpecEquipment2	Aufruf Sonderausrüstung nach Betriebsarten/Maschinensteuertafel
FC1552	SPEC_FC_SpecEquipment3	Aufruf Sonderausrüstung vor Daten senden/Störungsverwaltung
FC1553	SPEC_FC_SpecEquipment4	Aufruf Sonderausrüstung nach Daten senden/Störungsverwaltung

17 Sonstige Neuerungen

- Betriebsart 3 (BA3) „Service“ entfällt
- UserMD für Vorbesetzungen (*G-Typ konfig*)
- Brückenwerkzeuge dürfen nur mit normalem Werkzeug gewechselt werden
- NC oder Skript- Aufträge: Ansteuerung in gleicher Struktur
(*Skript-ordner im HMI*)
- Hauptverantwortliche für die einzelnen Funktionsmodule (bei GROB)
- Die gepflegte Liste der Hauptverantwortlichen befindet sich unter:
<http://grob-portal/Abteilungen/KO-EL/Lists/Aufgabe%20Kompetenz%20Verantwortung/AllItems.aspx>
Mit einer Schaltfläche des GROB-Tools „Schnellstart“ kann die Liste ebenfalls geöffnet werden. Es muss dazu konfiguriert sein.



18 Projektvorlage



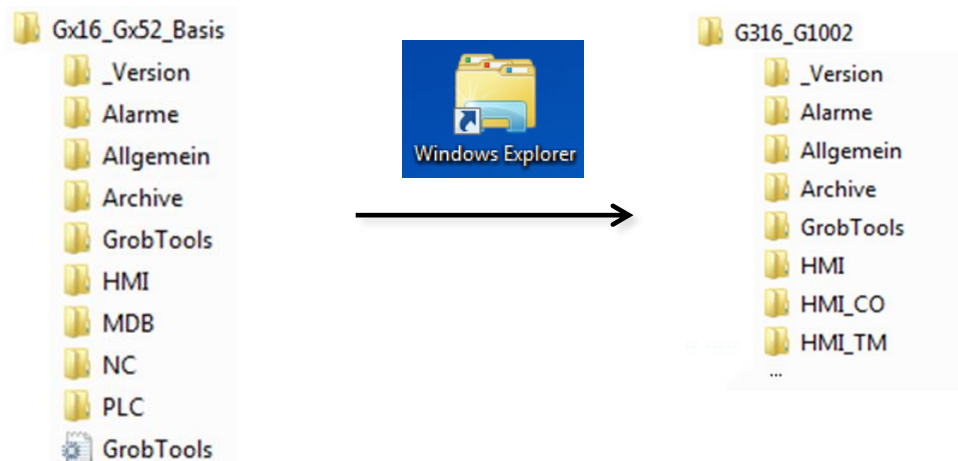
Projekt vor Integration der HW-Zuli und dem Einbinden der Module unbedingt sichern!

18.1 Projektvorlage erstellen

Projektvorlagen werden in Baustufe 6 aus einzelnen Funktionsmodulen erstellt. Dazu werden die benötigten Module und die Basisvorlage kopiert und anschließend mit HMI Pro CS, dem ProjectBuilder und GenAll zusammengeführt.

Speicherorte Auf Laufwerk S: befinden sich die einzelnen Funktionsmodule, aus denen eine Projektvorlage erstellt wird. Die Projektvorlage für Universalmaschinen befindet sich auf Laufwerk V. Hier gibt es für jeden Softwarestand ein Verzeichnis.

Projektvorlage Für die Erstellung der Projektvorlage zuerst Grundordnerstruktur von der Basis kopieren. Anschließend die benötigten Module in die Struktur kopieren.



1. Grundordnerstruktur von der Basisvorlage auf lokalen Rechner kopieren.
2. Inhalt von Verzeichnis „SW-Aktuel“ löschen.
3. Mit Windows Explorer oder „Beyond Compare“ alle benötigten Module in die Struktur kopieren.

HMI Für jedes Funktionsmodul ergibt sich ein jetzt ein separater HMI-Ordner. Die HMI-Ordner werden über Mehrfachaufruf von HMI Pro CS zusammen geführt:



1. In HMI PRO CS den HMI Ordner aus der Basisvorlage öffnen und mit der

- Schaltfläche „Übernehmen“ laden.
2. Mit einer zweiten Instanz von HMI PRO CS den HMI Ordner eines anderen Moduls laden.
 3. Bereiche aus der zweiten Instanz in das zuerst geöffnete HMI Pro CS kopieren.
 4. Bei Einrichtbildern sichtbare Zeilen kontrollieren.
 5. Nach Kopieren sämtlicher Module HMI-Verzeichnisse aus den Einzelmodulen löschen.

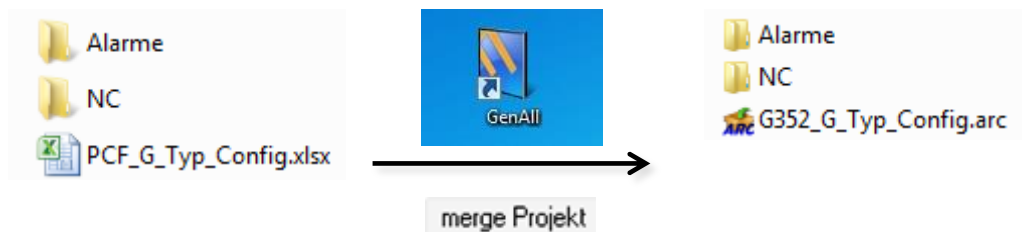
Rev.

PLC PLC-Bausteine der Funktionsmodule werden im S7 ProjectBuilder in den PLC-Ordner des Basismoduls integriert.



1. Das Projekt im S7-Manager mit GM-Nummer und Maschinenummer umbenennen.
2. Projektvorlage im S7-Projektbuilder als Target-Projekt auswählen.
3. PLC des Funktionsmoduls als Modulprojekt auswählen.
4. Mit „Integrate Module Project“ Übertragung starten.

GenAll Die Alarmer, M-Funktionen und Vorbesetzungen der Einzelmodule werden mit dem GenAll-Tool zusammengeführt bzw. in die „G_Typ_Config.arc“ der Maschine eingefügt. Maschinenkonfigurationen von Universalmaschinen werden nach Export aus SAP ebenfalls in die „G_Typ_Config.arc“ eingefügt.



1. GenAll starten.
2. Registerkarte „merge Projekt“ auswählen.
3. Kontrollkästchen bei „Merge Aktionen“ und „MachineConfiguration integrieren“ markieren.
4. Für Universalmaschinen das Kontrollkästchen „import umDB“ setzen.

Anleitung Die komplette Anleitung zur Erstellung der Projektvorlage befindet sich unter folgendem Link: [Projektvorlage neu erstellen](#)

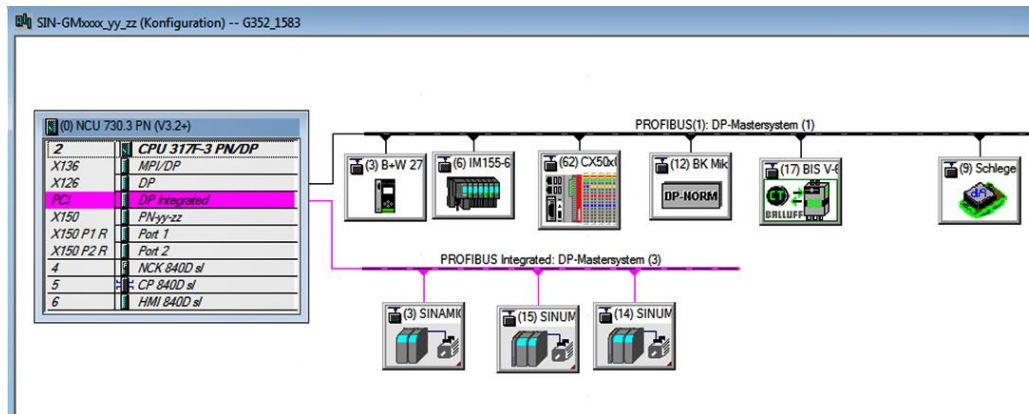
baustufe 6 neuigkeiten 1,1.docx

Hubert Schmid

18.2 Maschine erstellen

In Baustufe 6 wird für die Erstellung der Maschine die Konfiguration angepasst und dann die Zuordnungsliste (Zuli) eingelesen. Die Konfiguration von Universalmaschinen wird dabei aus SAP entnommen.

Konfiguration Zum Erstellen der Maschine wird zuerst die Hardwarekonfiguration in Step7 angepasst:

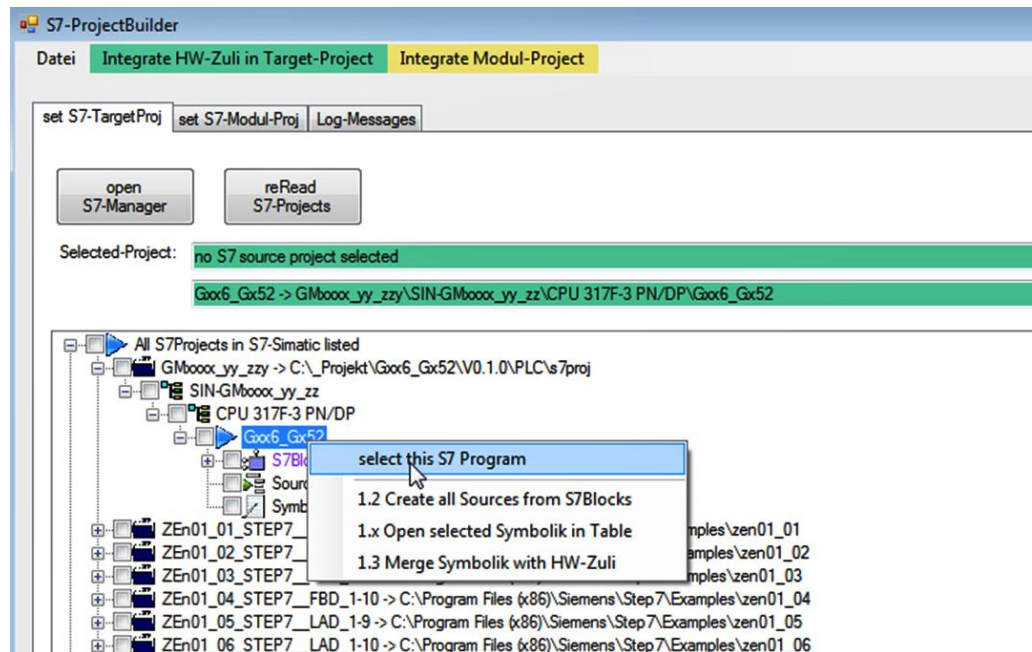


Anschließend werden ggf. nicht in der Projektvorlage enthaltene Sonderoptionen, wie z.B. Softlathe, in das Projekt kopiert (siehe 11.1).

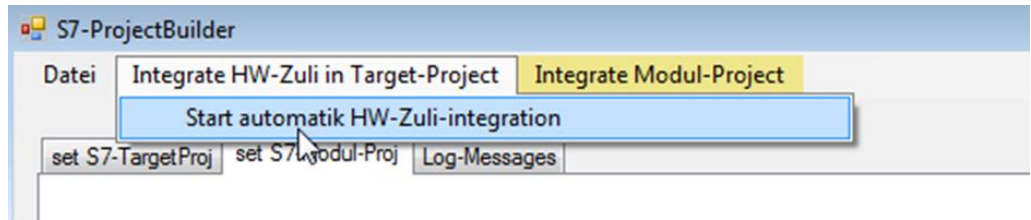
Zuordnungsliste Zum Einlesen der Zuordnungsliste (HW-Zuli) folgende Schritte durchführen:

1. HW-Zuli aus dem Verzeichnis „V:__Projekte_SM\Gxxx\Gxxx-xxxx\EKO\Zul“ in das Unterverzeichnis „... \Gxxx-xxxx\PLC\Symbole“ kopieren.

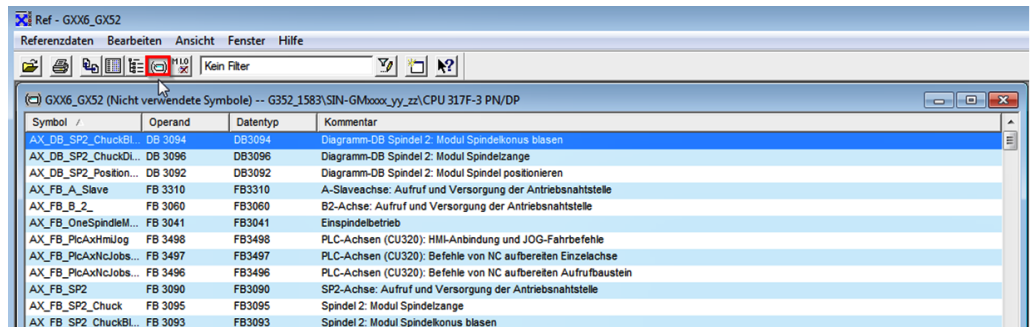
1. Im S7-ProjectBuilder die Projektvorlage als Target-Projekt auswählen:



2. Die Zuli mit dem Befehl „Start automatic HW-Zuli-integration“ integrieren:



3. Nach Einlesen und Übersetzen der ZULI die Verknüpfung der Symbole im Step7-Manager prüfen:

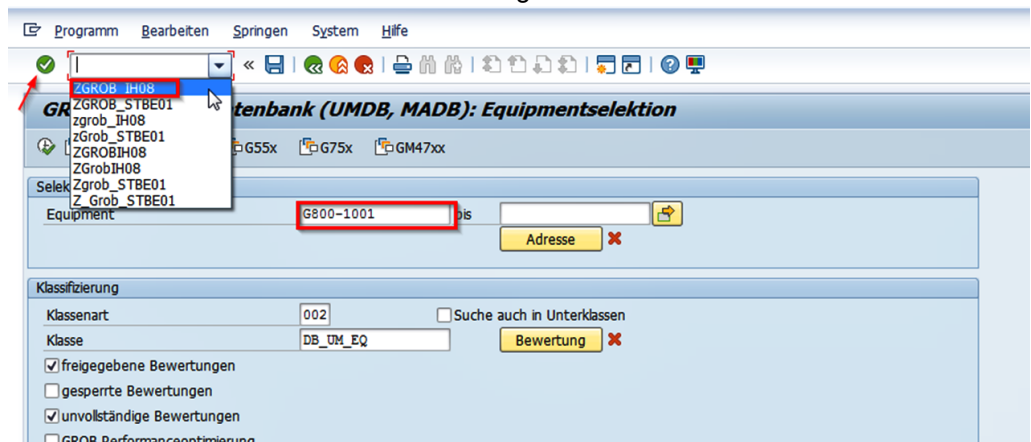


Werden hier Ein- oder Ausgänge angezeigt, sind diese nicht in der Software verwendet.

Konfiguration

Für Universalmaschine wird die Konfiguration aus dem SAP herausgelesen, ggf. ergänzt und dann mit HMI Pro Cs und dem GenAll-Tool eingelesen.

1. Bei Universalmaschinen zuerst die Konfiguration aus dem SAP herauslesen:



2. Die UMDB-Liste unter „Projekt \Allgemein \Dokumentation“ speichern.

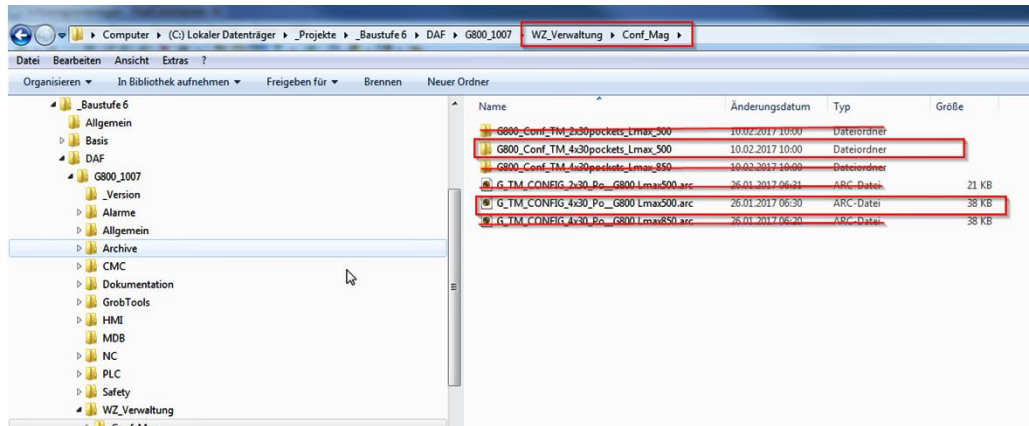
→ Die UMDB-Liste kann jetzt mit dem GenAll-Tool in die Maschinenkonfiguration integriert werden.

3. Wenn notwendig, Maschinenkonfiguration ergänzen. Dazu das Logfile beachten.

4. Wird HMI Operate auf einer NCU verwendet, das Verzeichnis „install“ löschen. Wird eine PCU verwendet, das „install“-Verzeichnis löschen.

5. Je nach Ausführung nicht zutreffende Magazin Konfiguration und Werkzeugjobliste löschen:

Rev. 1.0

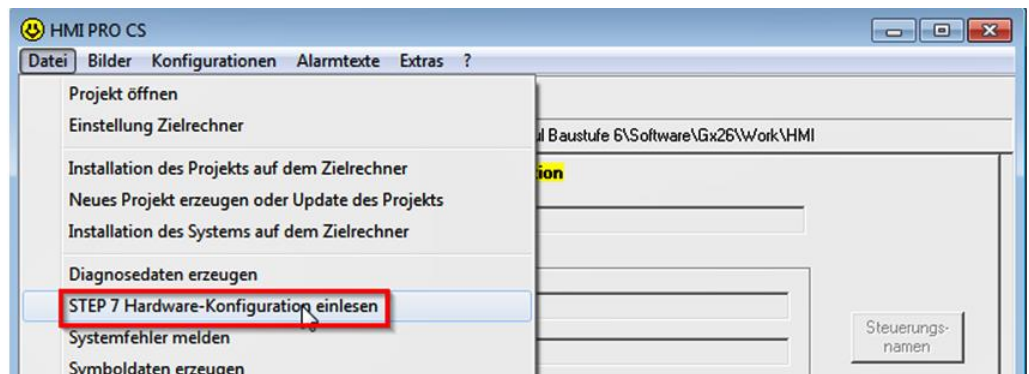


6. Die Datei „grmag.ini“ (1) aus dem Werkzeugjoblisten-Verzeichnis „oem\sinumerik\hmi\grjoblist\cfg“ in das Verzeichnis „HMI/Install“ kopieren.

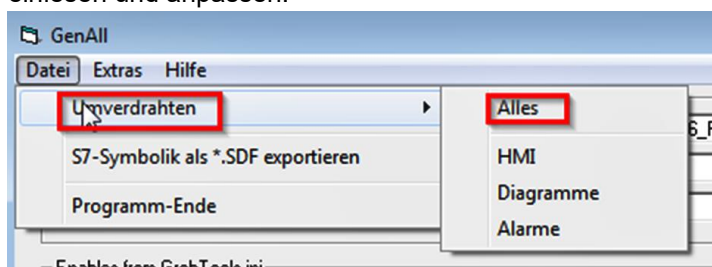
7. Je nach Maschinentyp nicht zutreffende Dateien im NC-, HMI- und/oder PLC-Ordner löschen.

Generieren

1. Die Hardwarekonfiguration mit dem Befehl „STEP 7 Hardware-Konfiguration einlesen“ in HMI PRO CS einlesen:



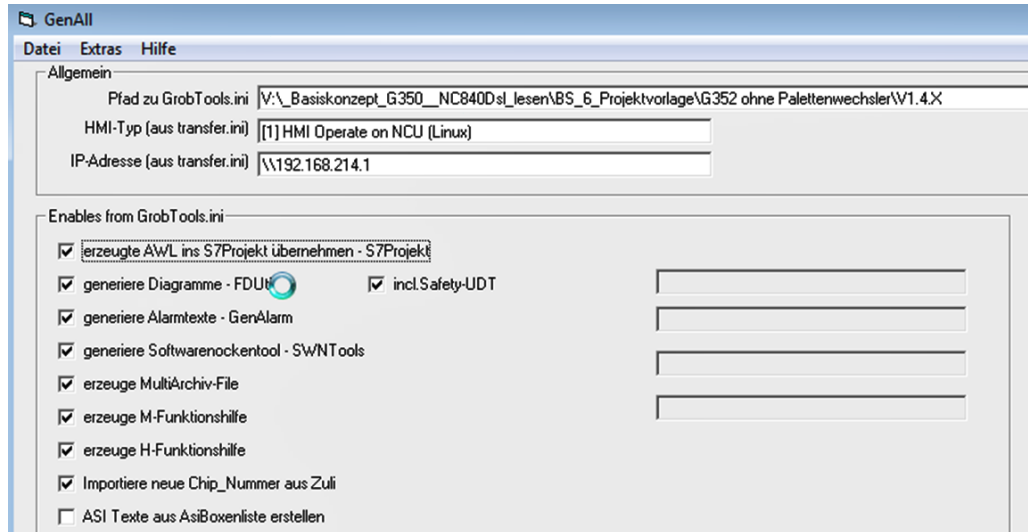
2. In GenAll die Hardwarekonfiguration mit dem Befehl „Umverdrahten“ > „Alles“ einlesen und anpassen.



3. Abschließend mit GenAll alle benötigten Dateien generieren.

baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid



→ Die benötigten Dateien sind generiert.

Anleitung Die komplette Anleitung zur Erstellung der Maschine befindet sich unter folgendem Link: [Maschine erstellen](#)

Rev.

baustufe 6 neuigkeiten 1.1.docx

Hubert Schmid