

Funktionsbaustein FB 20

zur Ansteuerung der SCHUNK-Module für Siemens S7-300/400 SPS-System

Softwareanleitung

Impressum

Urheberrecht:

Diese Anleitung bleibt urheberrechtlich Eigentum der SCHUNK GmbH & Co. KG. Sie wird nur unseren Kunden und den Betreibern unserer Produkte mitgeliefert und ist Bestandteil des Produktes. Ohne unsere ausdrückliche Genehmigung dürfen diese Unterlagen weder vervielfältigt noch dritten Personen, insbesondere Wettbewerbsfirmen, zugänglich gemacht werden.

Technische Änderungen:

Änderungen im Sinne technischer Verbesserungen sind uns vorbehalten.

Auflage: 01.01 | 06.08.2014 | de

© SCHUNK GmbH & Co. KG

Alle Rechte vorbehalten

Sehr geehrter Kunde,

wir gratulieren zu Ihrer Entscheidung für SCHUNK. Damit haben Sie sich für höchste Präzision, hervorragende Qualität und besten Service entschieden.

Sie erhöhen die Prozesssicherheit in Ihrer Fertigung und erzielen beste Bearbeitungsergebnisse – für die Zufriedenheit Ihrer Kunden.

SCHUNK-Produkte werden Sie begeistern.

Unsere ausführlichen Montage- und Betriebshinweise unterstützen Sie dabei.

Sie haben Fragen? Wir sind auch nach Ihrem Kauf jederzeit für Sie da.

Mit freundlichen Grüßen

Ihre SCHUNK GmbH & Co. KG

Spann- und Greiftechnik

Bahnhofstr. 106 – 134

D-74348 Lauffen/Neckar

Tel. +49-7133-103-0

Fax +49-7133-103-2399

info@de.schunk.com

www.schunk.com



Reg. No. 003496 QM08



Reg. No. 003496 QM08

Inhaltsverzeichnis

1	Zu dieser Anleitung	4
1.1	Mitgeltende Unterlagen	4
2	Systemvoraussetzungen	5
2.1	Softwareanforderungen	5
2.2	Hardwareanforderungen.....	5
3	Get a start (Schnellstart - Einstieg)	6
4	FB20 im SPS Programm einbinden	48
4.1	FB20 und die Datenbausteine	48
4.2	Einstellung der Schnittstelle/Kommunikation	48
4.3	Datenbaustein für die Service- bzw. Systemdaten	49
4.4	Zeitwert für den FB20.....	50
4.5	Warnungen / Störungen anzeigen	52
4.6	Weitere Einstellungen der Kommunikation.....	54
4.7	Statussignal, Rückmeldungen.....	56
4.8	Digitale Ein-/Ausgänge am Modul.....	57
4.9	Allgemeine Signale und Daten	59
4.10	Referenzieren Auto	61
4.11	Referenzierung Hand.....	63
4.12	Antrieb tippen	64
4.13	Allgemeine Vorgaben	65
4.14	Regelungsfunktionen.....	66
4.15	Verfahrenstabelle	69
5	Unterschiede zwischen FB10 / FB20.....	70
5.1	FB Beschaltung: Input + In-/Output	70
5.2	FB Beschaltung: Output.....	74
6	FB CMD Kommandos	76
7	Versionshinweise	78

1 Zu dieser Anleitung

Diese Anleitung beschreibt den Funktionsbaustein FB20.
Er dient als Antriebsmodul um den Profibus anzusteuern.

1.1 Mitgeltende Unterlagen

- Der FB20 wurde für das SPS System S7-300/400 der Firma Siemens AG entwickelt. Daher sollte die Dokumentation der entsprechenden CPU beachtet werden, inkl. der Handbücher zu der STEP7 Programmiersoftware.
- Beim Profibus sind die Aufbaurichtlinien zu befolgen, die von der Profibusnutzorganisation PNO erstellt wurden.
- Die Funktionen der Antriebsmodule sind im Handbuch 'Motion Tool Schunk, Software Handbuch' beschrieben, die vom FB20 teilweise angestoßen werden.

2 Systemvoraussetzungen

2.1 Softwareanforderungen

Der FB20 wurde mit dem STEP7 Manager V5.5 programmiert und getestet.

Ältere Programmiersoftware oder alternative Programmierertools könnten auch funktionieren, diese wurden von SCHUNK jedoch nicht getestet.

SCHUNK empfiehlt daher, den aktuellen STEP7 Manager zu verwenden, damit bei Rückfragen ein bestmöglicher Support möglich ist.

2.2 Hardwareanforderungen

Bei der Entwicklung des FB20 wurden verschiedene SPS Systeme eingesetzt (z.B. S7-315 2DP, ET200s der IM151-8 PN/DP CPU mit Profibus-Master-Baugruppe, sowie einer VIPA CPU315-SB, aus verschiedenen Herstellungsjahren).

Diese verursachten keine Probleme.

Bei den verschiedenen CPU's und dessen Firmware kann nie ausgeschlossen werden, dass inkompatibles Verhalten auftreten könnte, gerade in Verbindung mit dem Programmiertool.

empfohlene Hardware Voraussetzung

- S7-300/400 SPS System (auch kompatible Systeme sind möglich)
- Profibus DPV0
- Profibus max. 1,5Mbit
- Min. Arbeitsspeicher 32kByte
- Min. Ladespeicher 32kByte
- CPU's müssen Programmbausteine bis 16kByte verarbeiten können
- Lokaldatenspeicher von min. 512Byte müssen beim FB20 bei der Ausführung frei sein.
- Antriebsmodul Version Firmware SMP V1.56

3 Get a start (Schnellstart - Einstieg)

In diesem Kapitel wird beispielhaft ein Projekt erstellt, wie z.B. ein Antriebsmodul projektiert werden kann, so dass am Ende das Antriebsmodul über die Variablentabelle gesteuert werden kann.

SCHUNK empfiehlt besonders unerfahrenen Technikern dieses Kapitel sorgfältig zu lesen, und Punkt für Punkt abzuarbeiten.

SPS Projekt erstellen

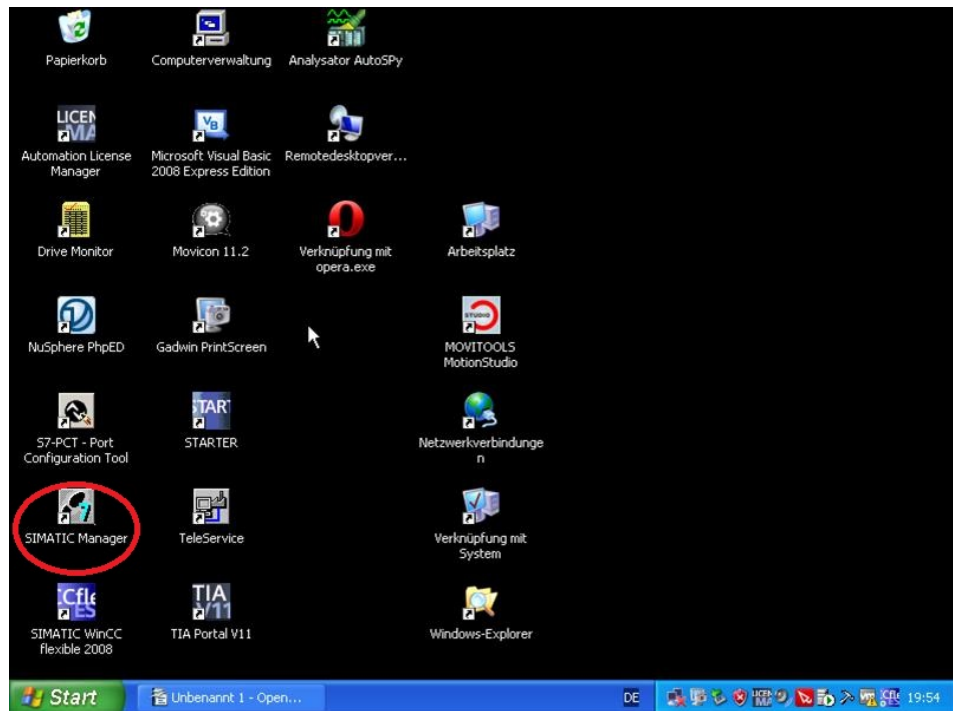


Abb. 1

1 SIMATIC Manager ÖFFNEN

Um ein neues Projekt zu erstellen muss der Simatic Manager gestartet werden.

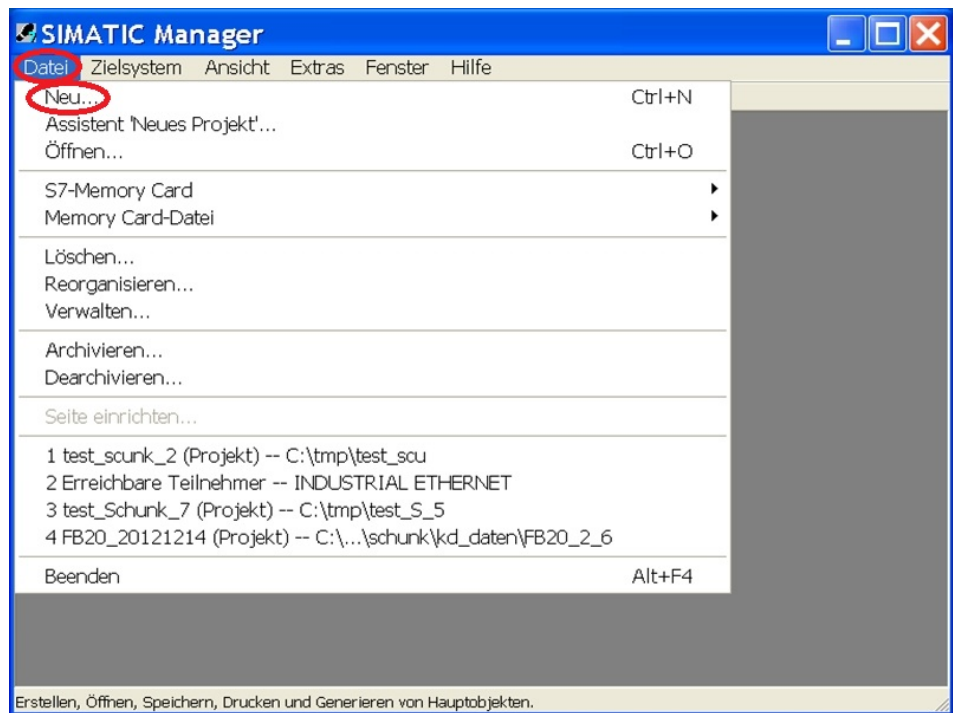


Abb. 2

2 DATEI > NEU

Zu Beginn muss ein neues Projekt erstellt werden.

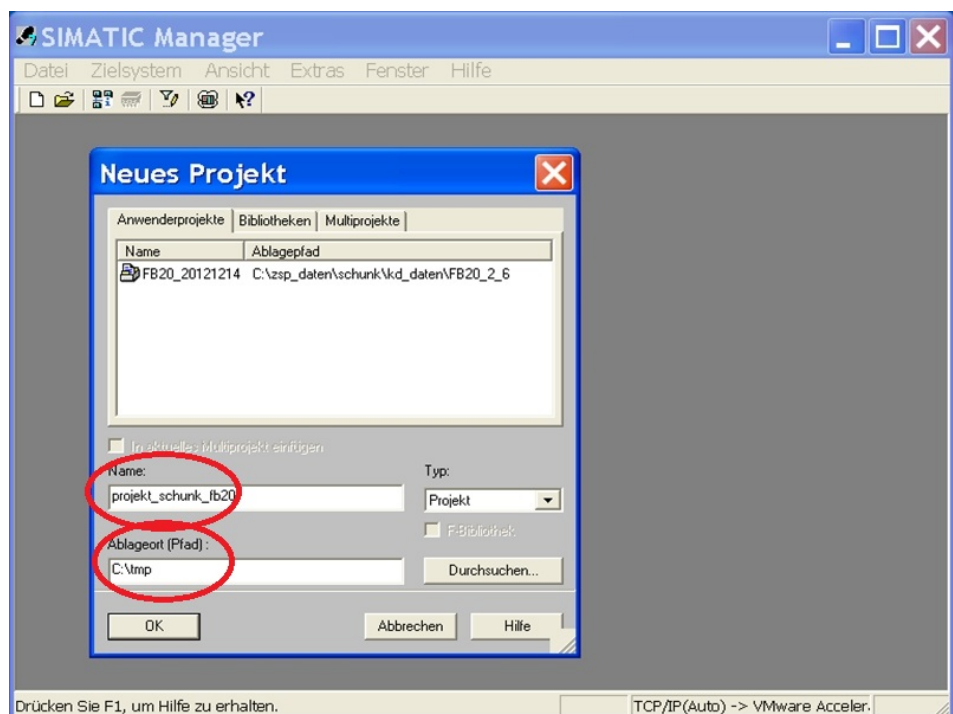


Abb. 3

3 Projekt NAME und PFAD festlegen

Jedes Projekt benötigt einen Namen. Dieser sollte eindeutig sein und einen Ablageort haben, der hier festgelegt wird.

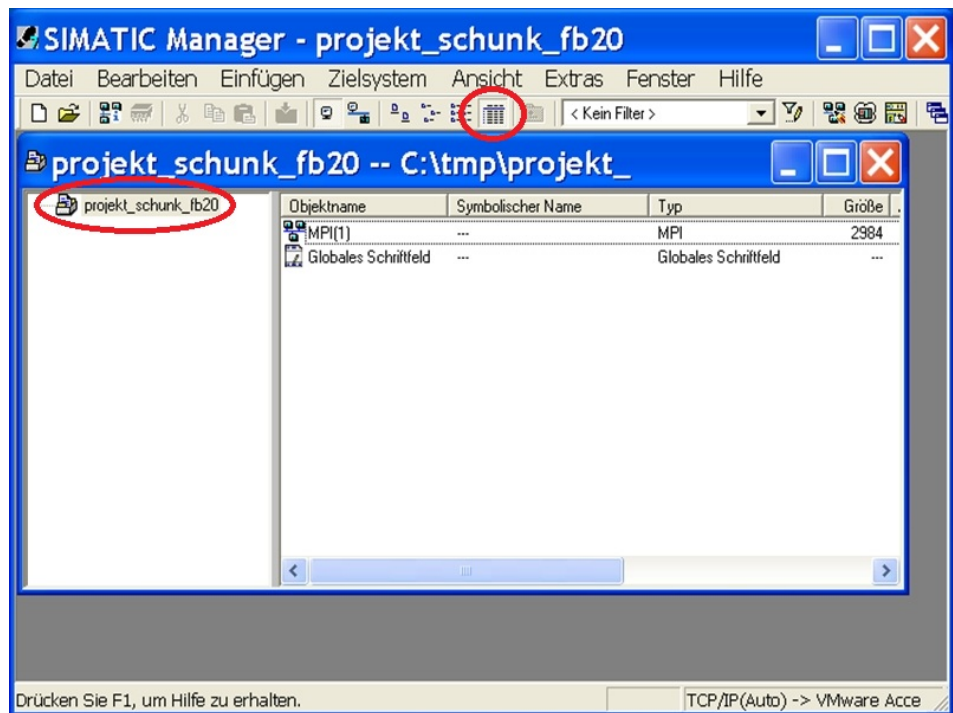


Abb. 4

4 Projekt Ansicht

Der Projektname wird links angezeigt.

Um eine bessere Übersicht zu erhalten, empfehlen wir die Ansicht auf Details einzustellen.

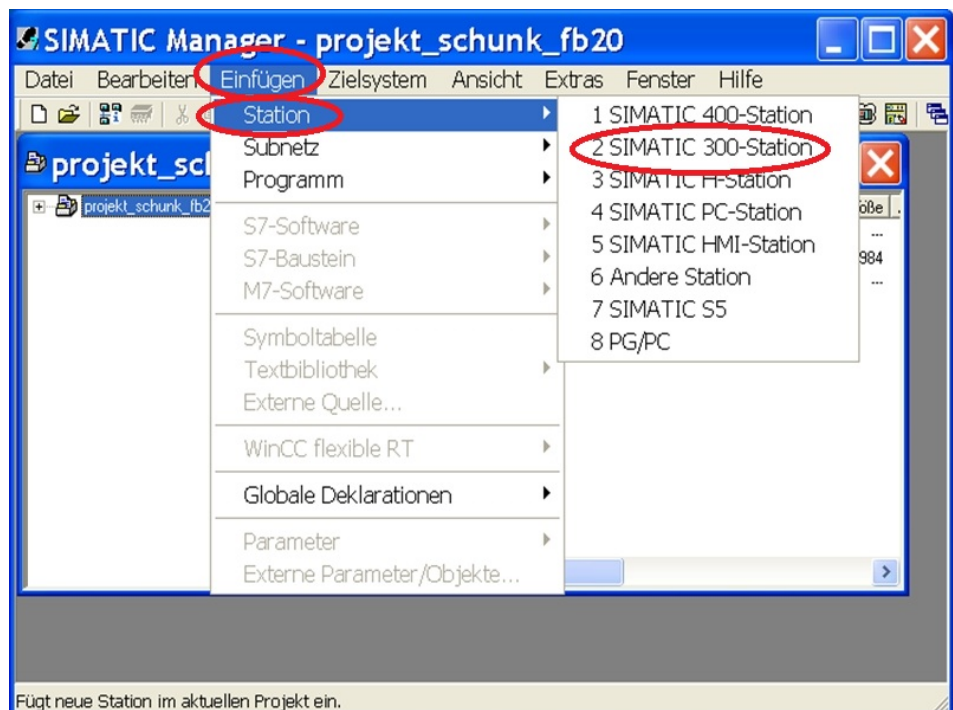


Abb. 5

5 SPS Station einfügen: EINFÜGEN.> STATIONEN

Projekt markieren und Baureihe der Simatic 300/400 auswählen. Dieses ist Grundlage der Hardwarekonfiguration.

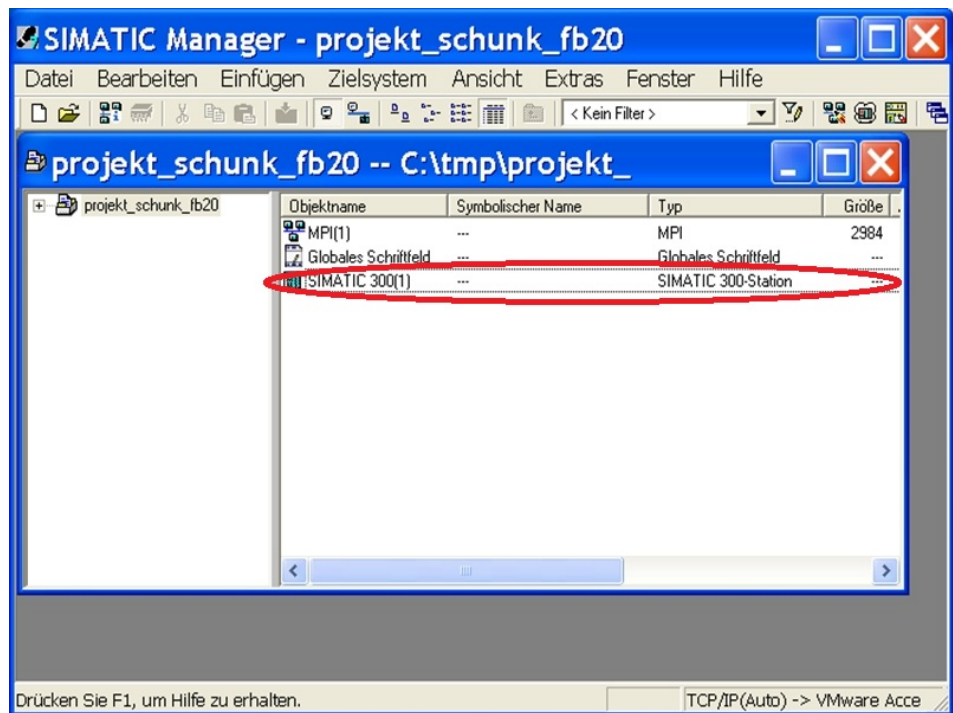


Abb. 6

6 Grundlage Simatic 300 (1) ist hinzugefügt

Im rechten Fenster wird Simatic 300 (1) angezeigt.

Um die Ordner Struktur weiter aufzuklappen, im linken Fenster vor dem Projektnamen auf (+) klicken.

GSD Dateien installieren Um Mit der Projekterstellung fortfahren zu können, müssen die GSD Dateien der Firma Schunk installiert sein.

Die Installation der GSD Dateien ist auf den nachfolgenden Seiten beschrieben. Wenn die GSD Dateien bereits installiert sind, können die nächsten 17 Schritte übersprungen werden.

Bezugsquelle der **GSD Datei**

- Auf dem mitgelieferten Datenträger
- Download unter www.schunk.com

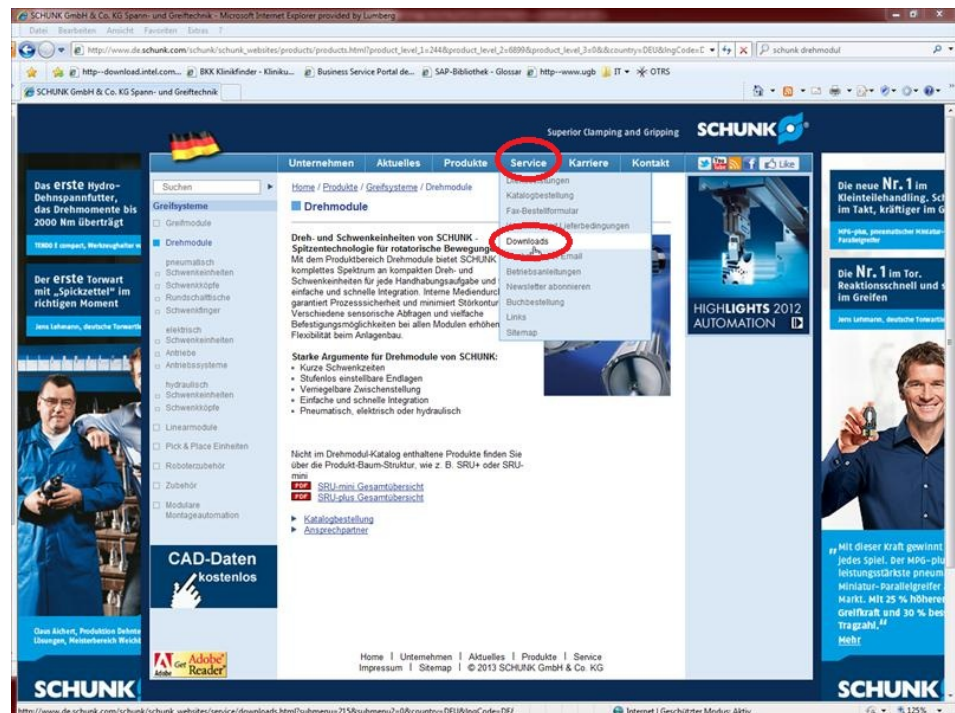


Abb. 7

- 1 **Herunterladen aus dem Internet** <http://www.schunk.com>
Den Bereich Service Downloads öffnen

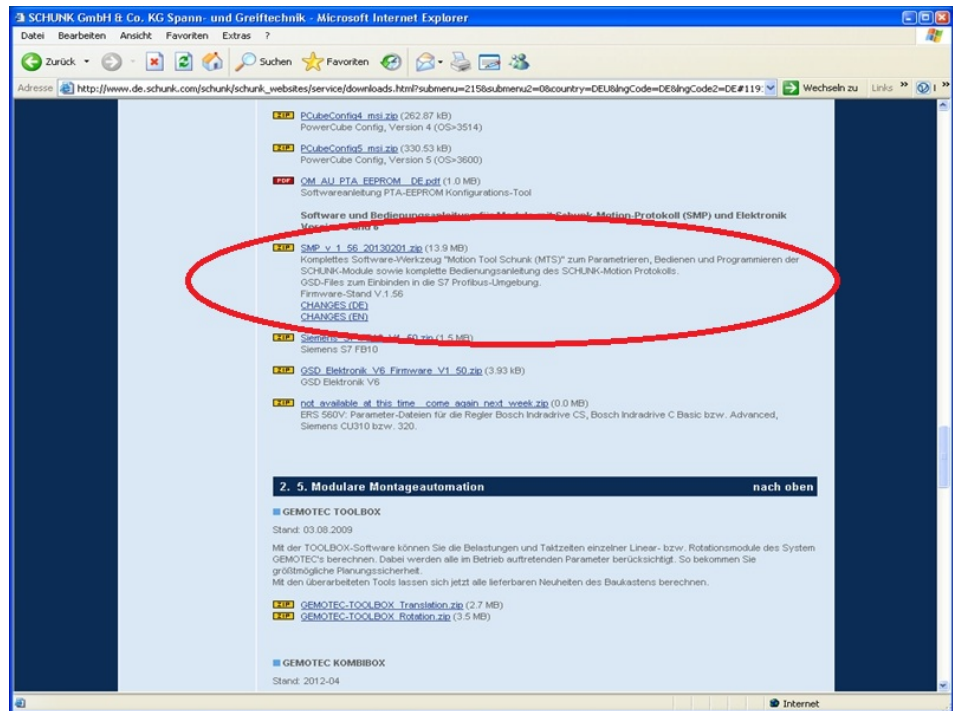


Abb. 8

2 Im Bereich "2.4 Mechatronik" die Datei "SMP_v_X_XX_XXXXXXX.zip" anklicken.

⇒ Die Seite zum Download der Datei öffnet sich:

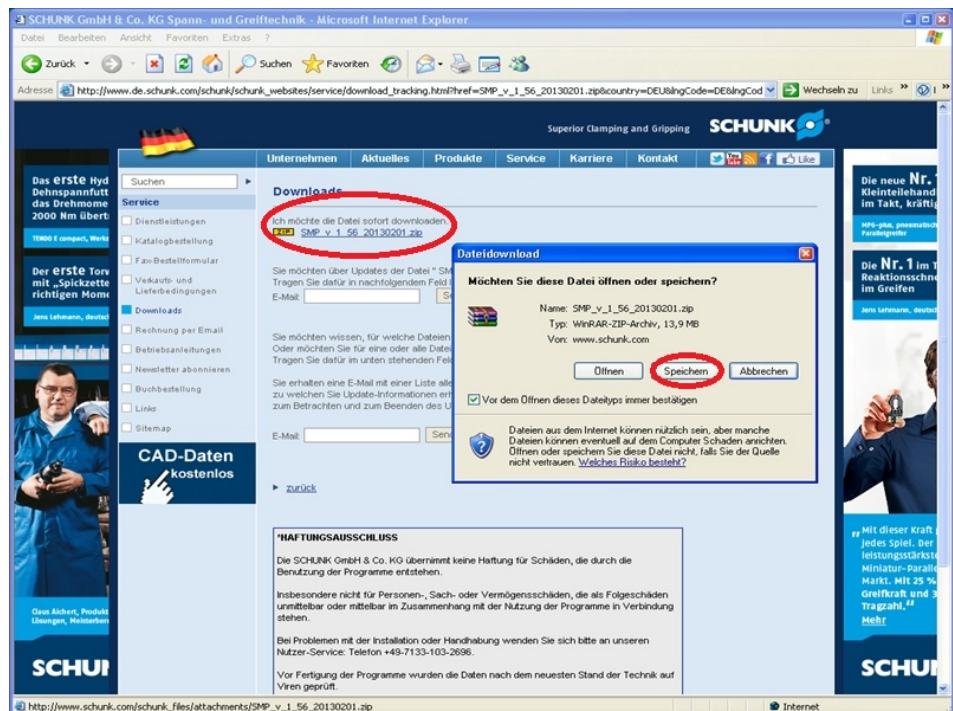


Abb. 9

3 Datei "SMP_v_X_XX_XXXXXXX.zip" anklicken und z.B. unter C:\Sammelbox speichern.

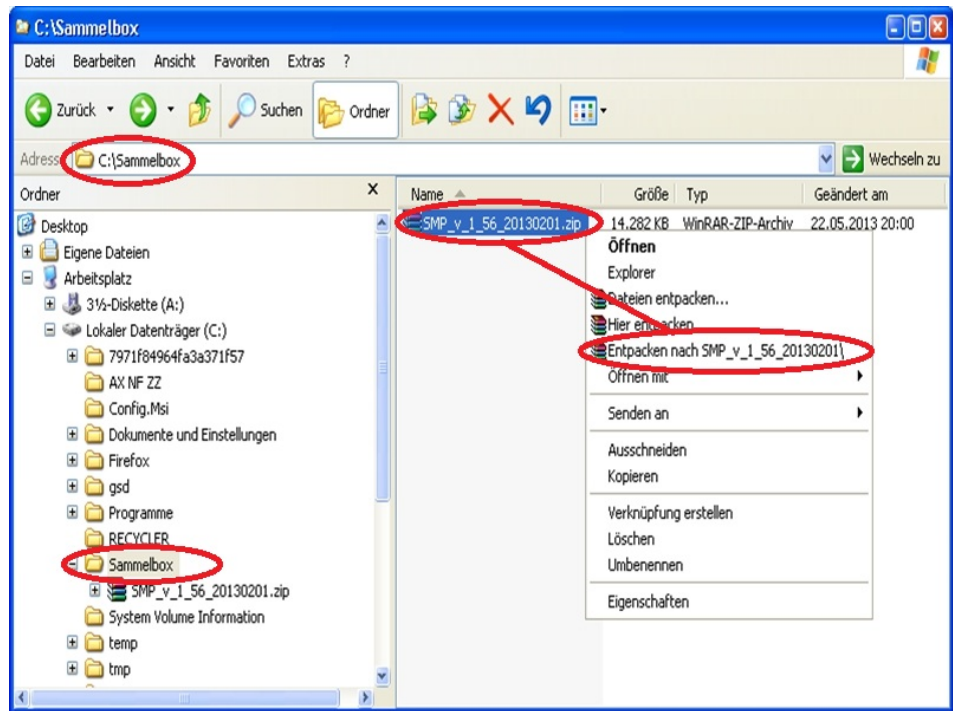


Abb. 10

- 4 Verzeichnis C:\Sammelbox öffnen und die Zip Datei in den Ordner SMP_v_X_XX_XXXXXXXXX entpacken.

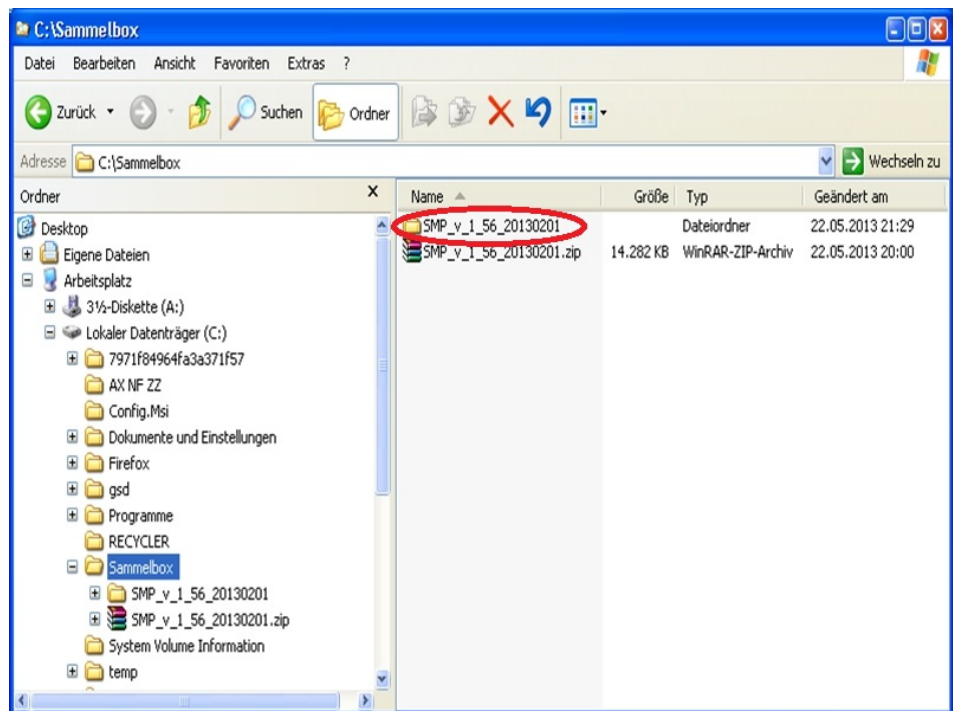


Abb. 11

⇒ Zip Datei ist entpackt.

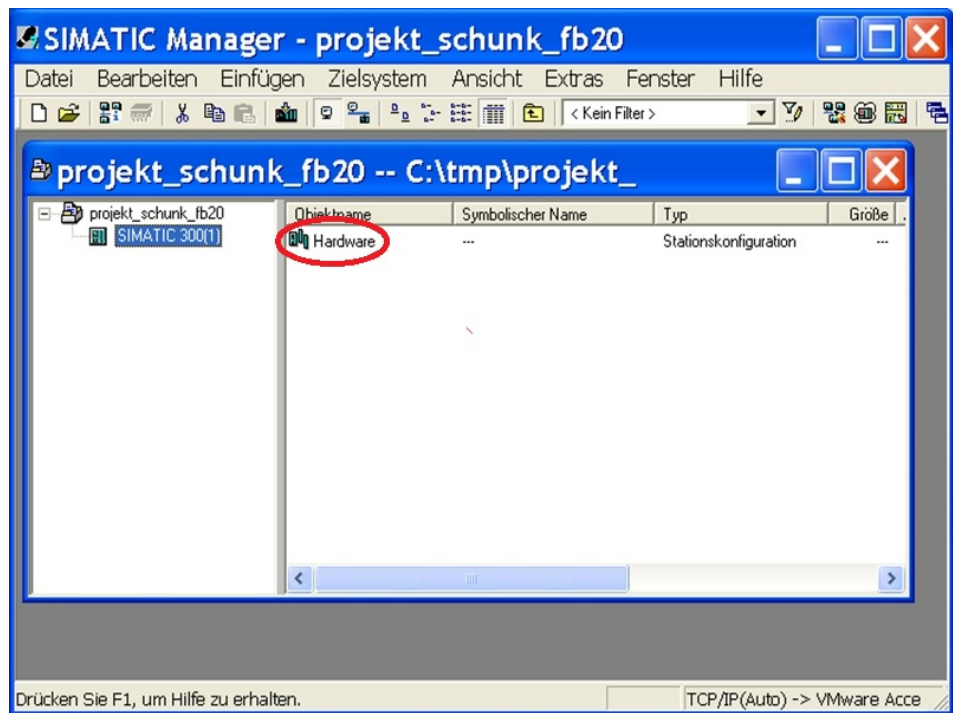


Abb. 12

- 5 **SIMATIC Manager: Hardwarekonfiguration öffnen.**
Mit einem Doppelklick auf Hardware öffnet sich das entsprechende Programmfenster.

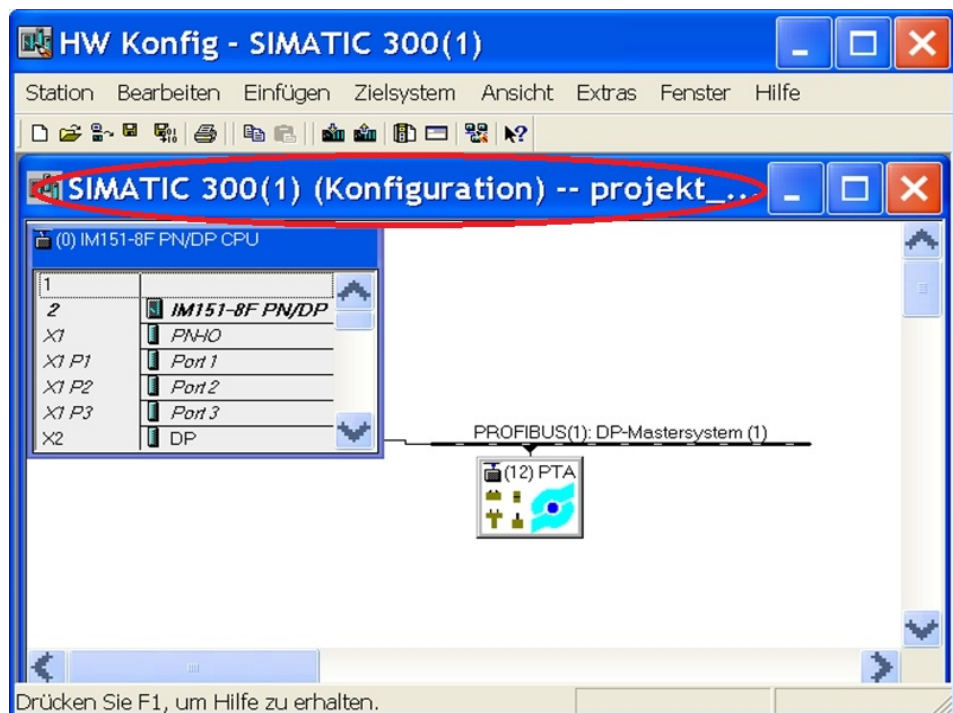


Abb. 13

- 6 Das geöffnete Projekt in der Hardwarekonfiguration schließen.

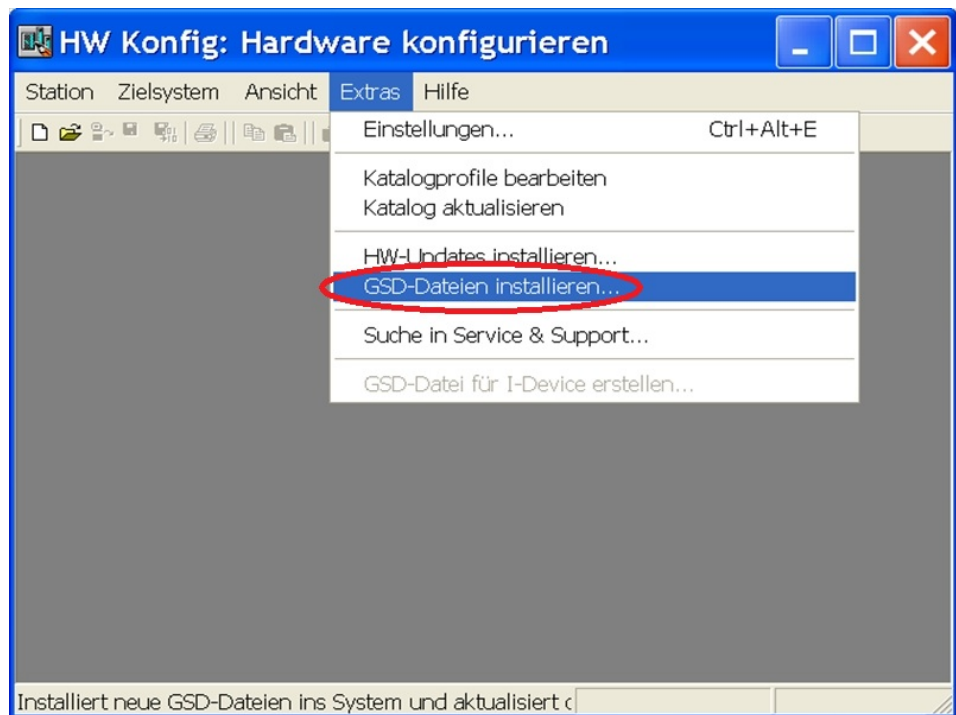


Abb. 14

- 7 In der Menüleiste "Extras", den Menüpunkt "GSD-Dateien installieren" auswählen.

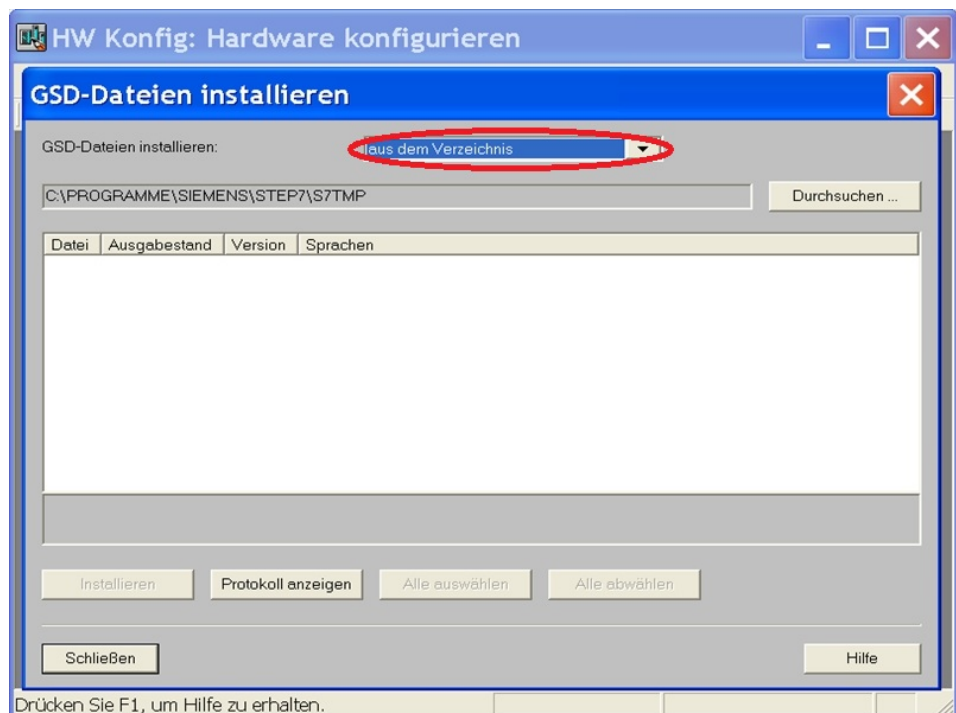


Abb. 15

- 8 Im Auswahlfeld "GSD Dateien installieren" den Eintrag "aus dem Verzeichnis" auswählen.



Abb. 16

9 Schaltfläche "Durchsuchen" anklicken.

⇒ Das Auswahlfenster "Ordner suchen" öffnet sich

10 Pfad mit der zuvor entpackten GSD Datei auszuwählen (In unserem Beispiel C:\Sammelbox). In diesem Verzeichnis den Ordner "SMP_v_X_XX_XXXXXXX\GSD\Hardware Version 5.x" auswählen und auf Schaltfläche "OK" klicken.

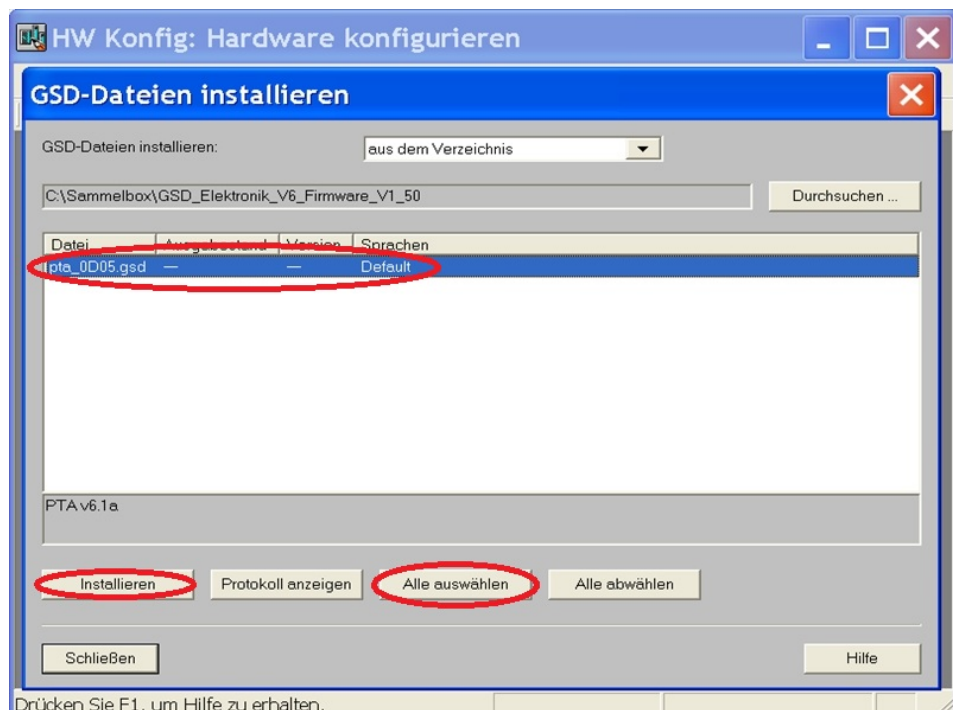


Abb. 17

11 "Alle angezeigten Dateien" anklicken.

12 "Installieren" anklicken.



Abb. 18

- 13 Bevor die GSD Dateien installiert werden erscheint eine **Meldung**. Diese Meldung **lesen** und mit **Ja** bestätigen, wenn die Installation ausgeführt werden soll.



Abb. 19

- 14 Daraufhin kann sich ein weiteres Fenster öffnen. Diese Meldung **lesen** und über die weitere Vorgehensweise entscheiden. SCHUNK empfiehlt, die Datei zu ersetzen. Mit "Ja" bestätigen. Es erscheinen noch weitere Meldungen. Da wir uns für das Ersetzen entschieden haben, müssen diese mit Ja bestätigt werden.

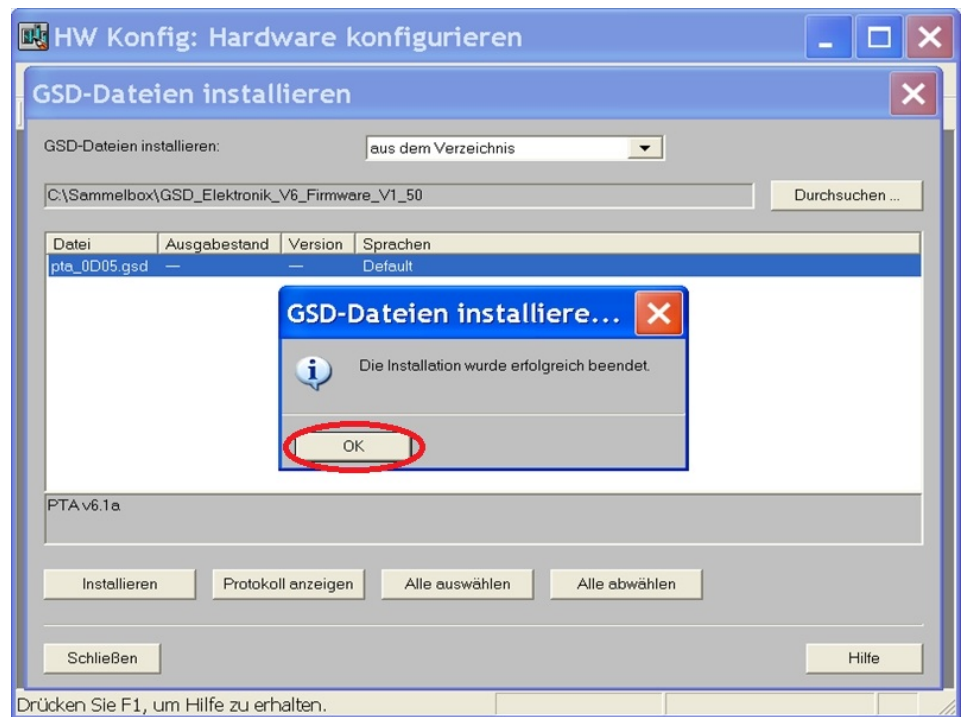


Abb. 20

15 Installation wurde erfolgreich beendet. Weiter mit "OK".

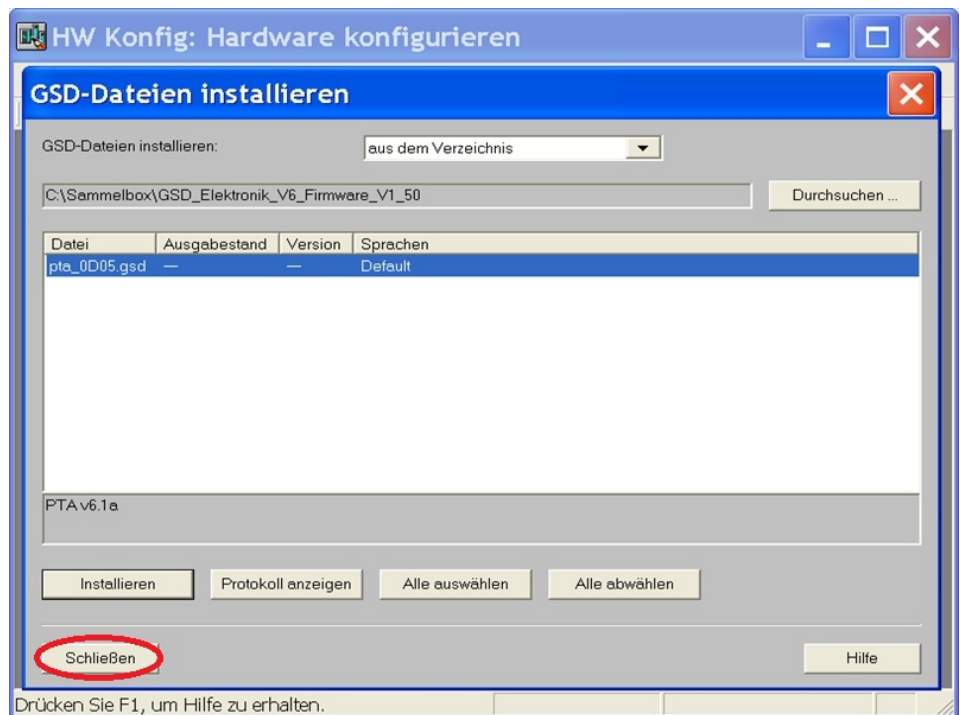


Abb. 21

16 Schaltfläche "Schließen" anklicken, um die Installation abzuschließen.



Abb. 22

17 Hardwarekonfiguration schließen und erneut öffnen damit die Daten übernommen werden.

Hardwarekonfiguration

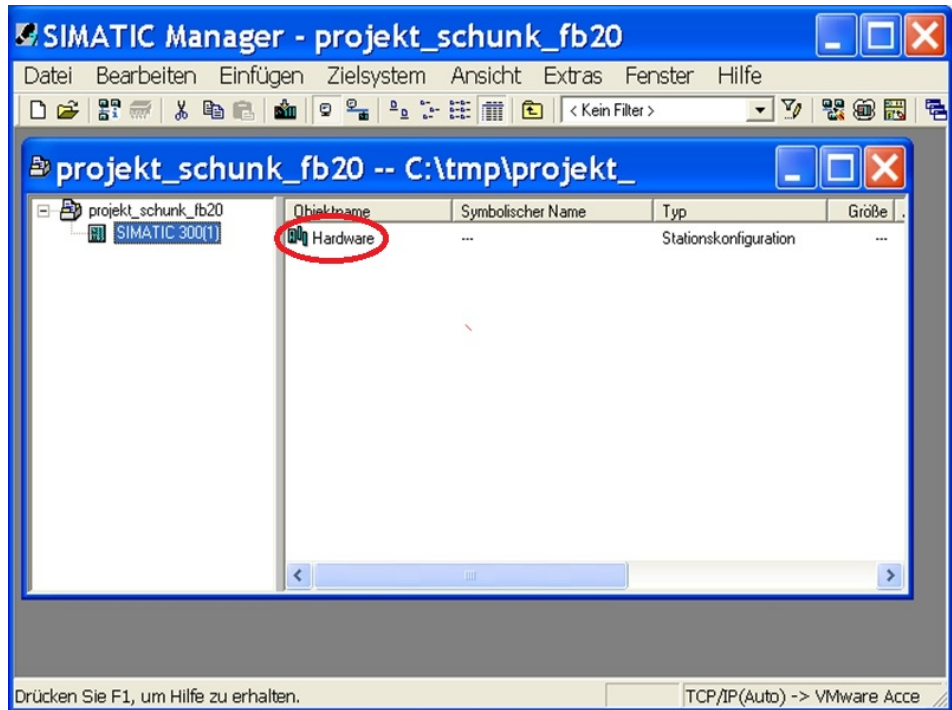


Abb. 23

1 HARDWARE konfigurieren

Für das Projekt müssen die Baugruppen in der Hardware Konfiguration eingetragen werden.

Doppelklick auf "Hardware"

⇒ Das entsprechende Programmfenster öffnet sich.

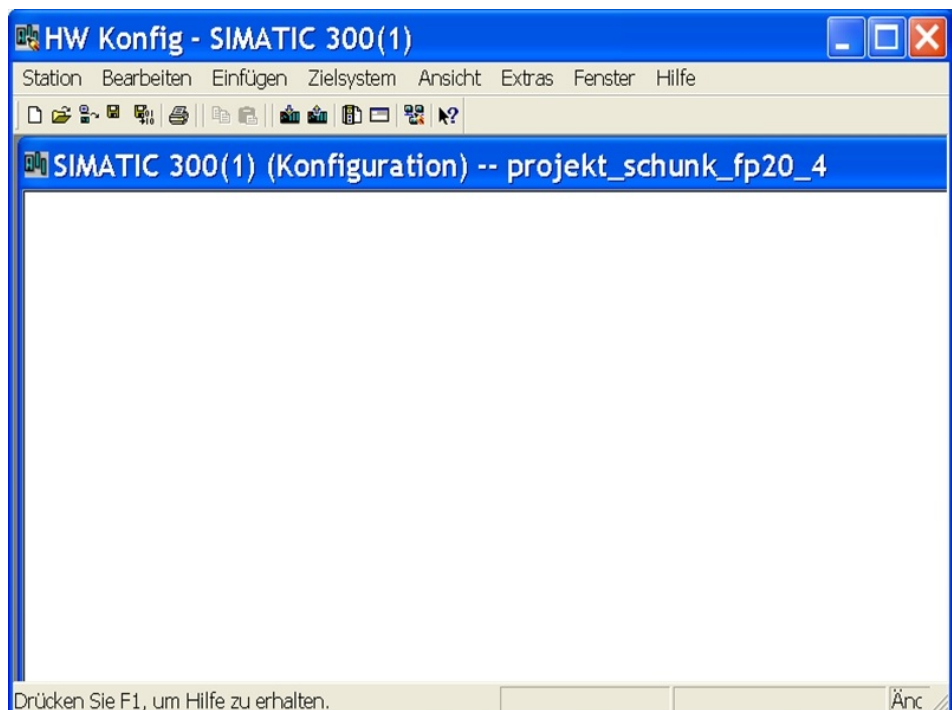


Abb. 24

2 Wenn Hardware Konfiguration ohne Hardwarekatalog geöffnet wird, könnte es so aussehen.

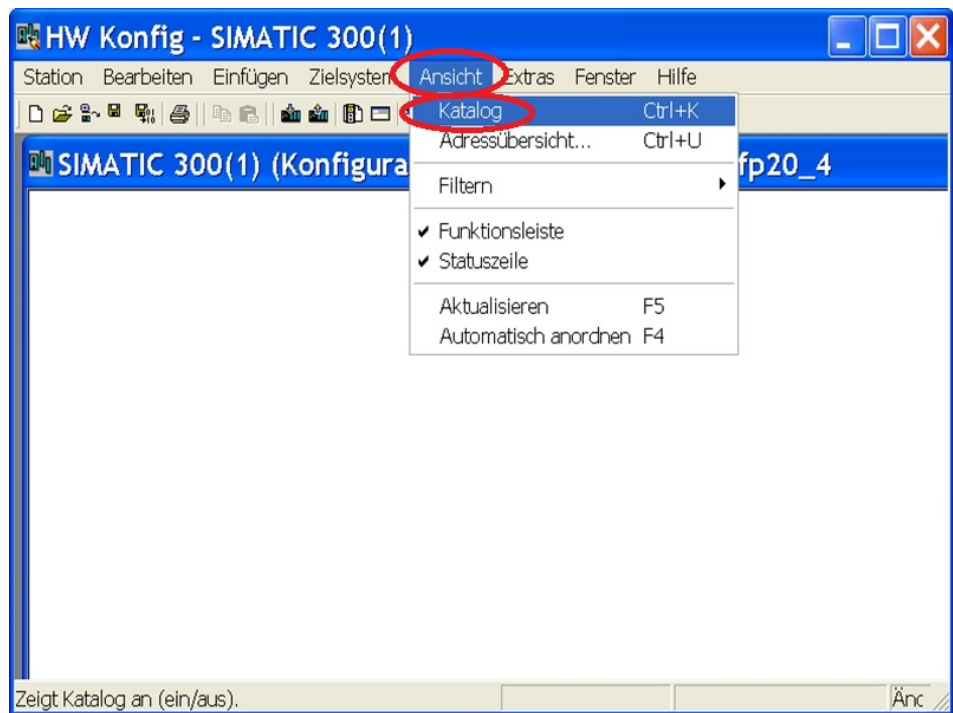


Abb. 25

- 3 Wenn der **Hardwarekatalog** nicht geöffnet ist, kann über die Menüleiste unter "Ansicht" der Menüpunkt "Katalog" angeklickt werden. Damit lässt sich der **Hardwarekatalog** ein-/ausblenden. Alternativ kann die Tastenkombination 'Ctrl+K' bzw. 'Strg+K' genutzt werden.

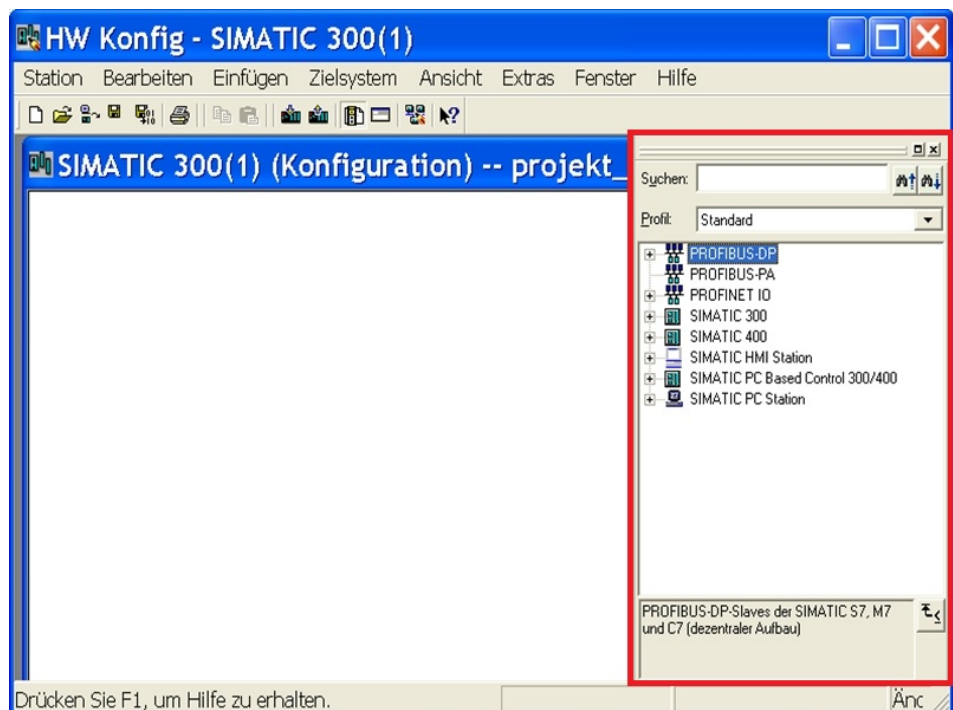


Abb. 26

⇒ Der Hardwarekatalog ist nun eingeblendet.

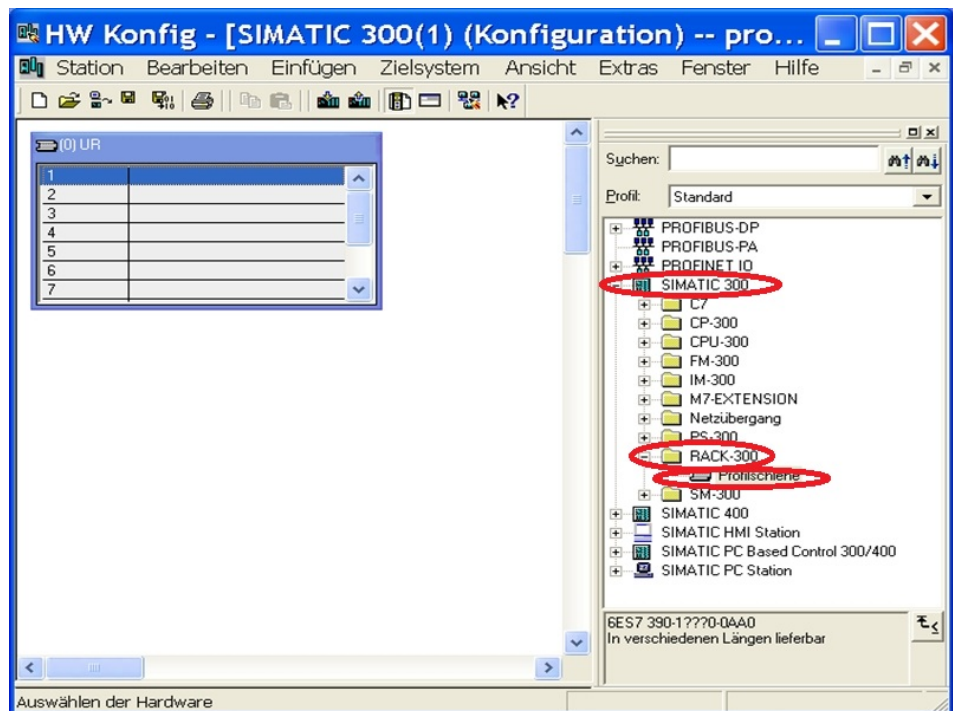


Abb. 27

4 PROFILSCHIENE hinzufügen

Als Grundlage „wie in der Realität“ wird für den Aufbau der Hardwarekomponenten eine Profilschiene benötigt.

Im Ordner Simatic 300 den Ordner Rack öffnen und diesen per Drag and Drop in das leere linke Fenster ziehen.

⇒ Für Steuerungen die auf einer C- Schiene montiert werden (z.B. ET200s mit CPU), entfällt dieser Schritt.

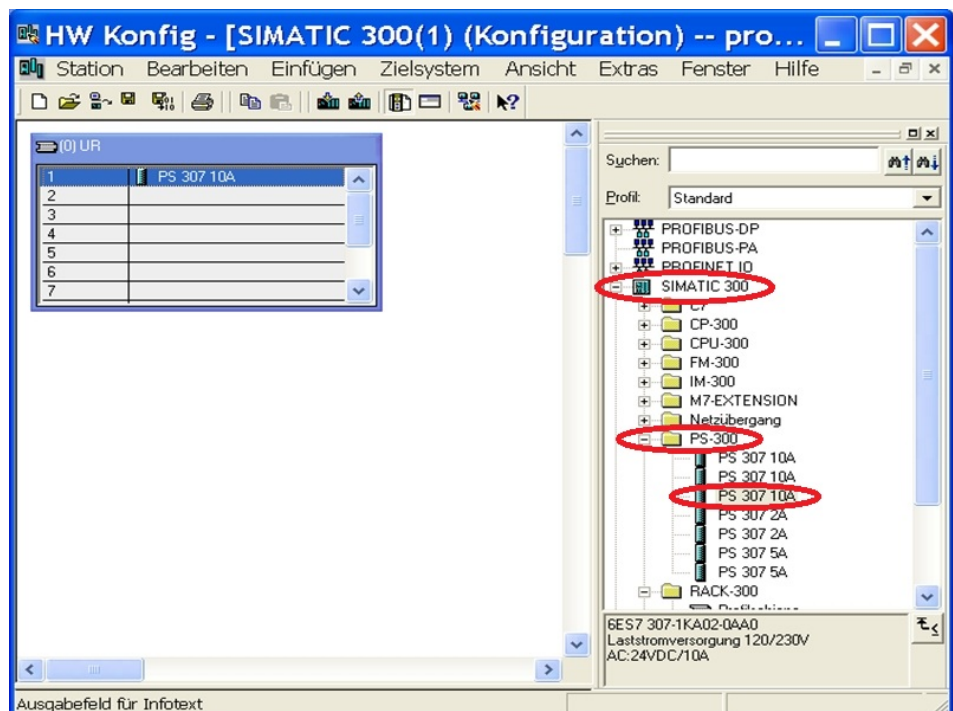


Abb. 28

5 NETZTEIL hinzufügen

Optional kann ein **Netzteil** aus dem Ordner PS300 per Drag and Drop an **1. Stelle** der Profilschiene gezogen werden.

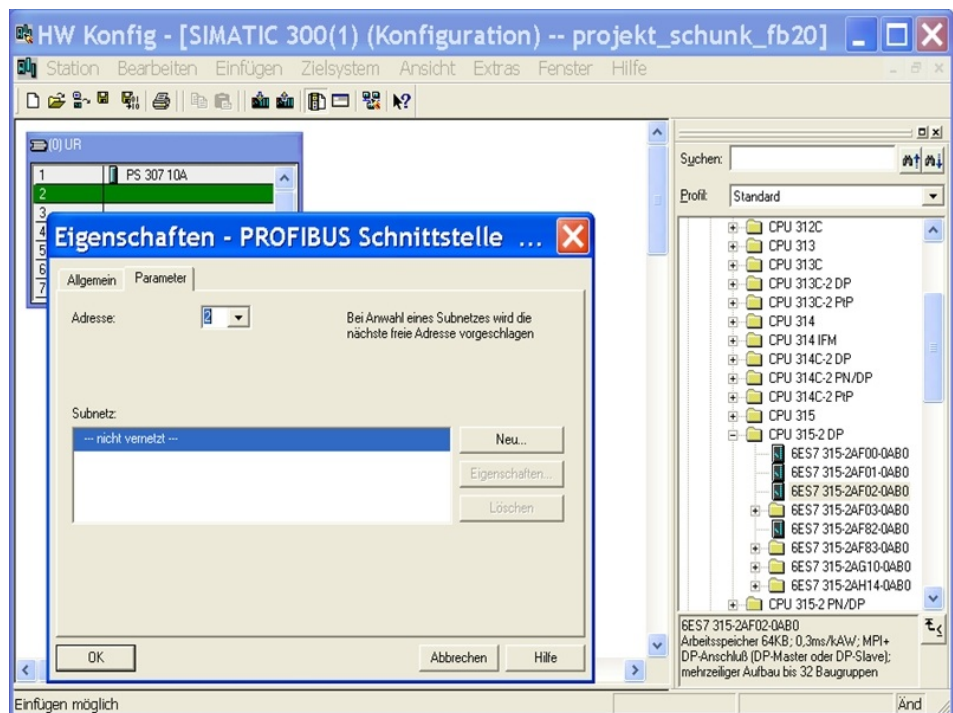


Abb. 29

6 CPU auswählen und konfigurieren

CPU per Drag and Drop an **2 Stelle** der Profilschiene ziehen. Bei der Auswahl der **CPU** unbedingt auf die **Bestellnummer** der Simatic S7 achten. Details zu der Baugruppe sind unten rechts ersichtlich.

ACHTUNG: Für dieses Beispielprojekt sollte die **PROFIBUS Schnittstelle** auf **Adresse 2** eingestellt werden.

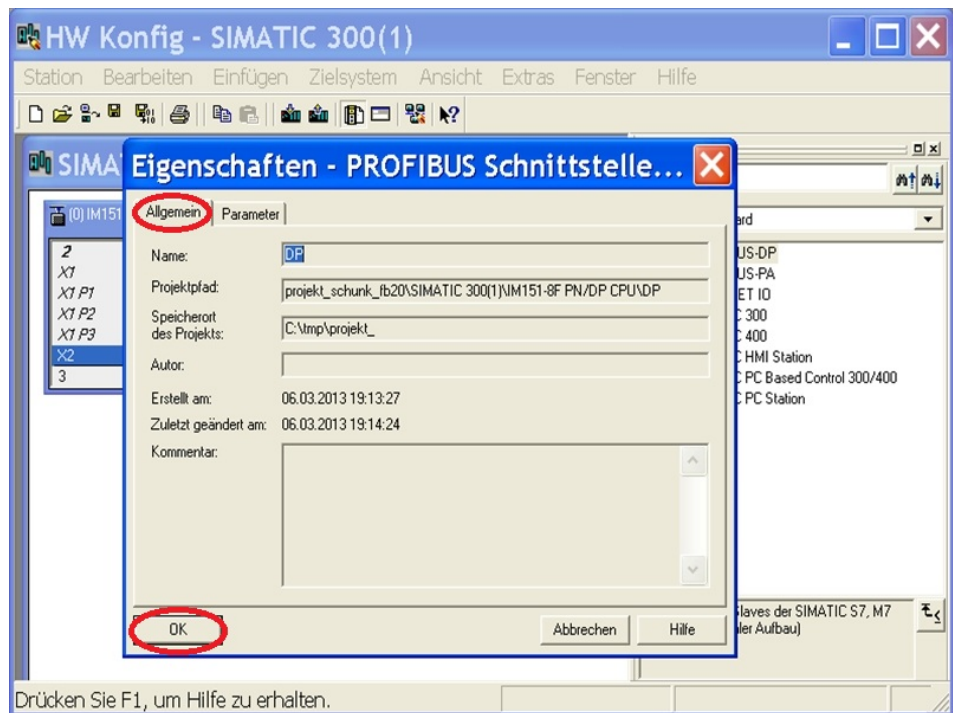


Abb. 30

- 7 Unter **Allgemein** wird der **Profilpfad** und der **Speicherort** angezeigt, kann dort aber nicht **verändert** werden. Auf "Neu" klicken und ein **Subnetz** hinzuzufügen.

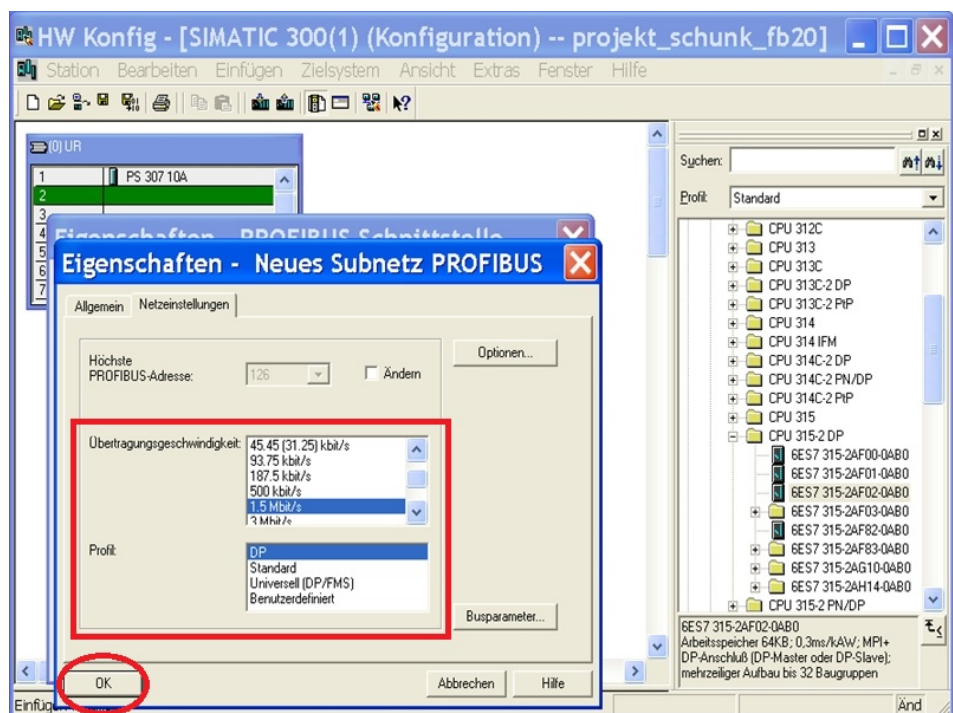


Abb. 31

- 8 **Netzwerkeinstellungen kontrollieren**
STANDARDEINSTELLUNGEN beibehalten. Diese sollten auf 1,5Mbit sein, mit dem Profil DP.
 Weiter mit "OK".

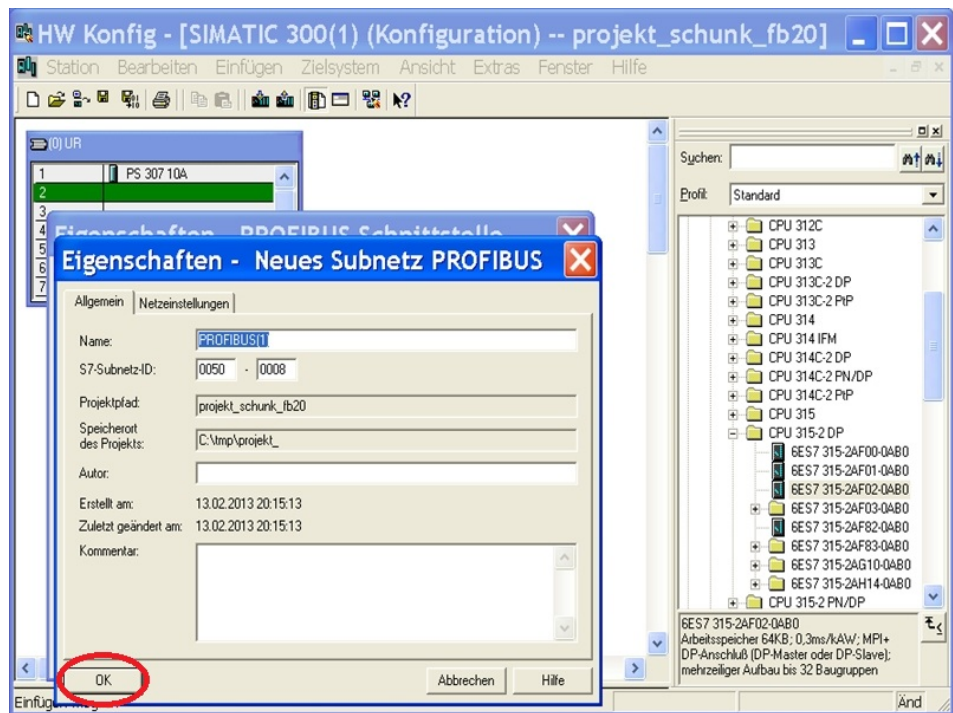


Abb. 32

9 Eigenschaften Subnetz

STANDARDEINSTELLUNGEN beibehalten.

Weiter mit "OK".

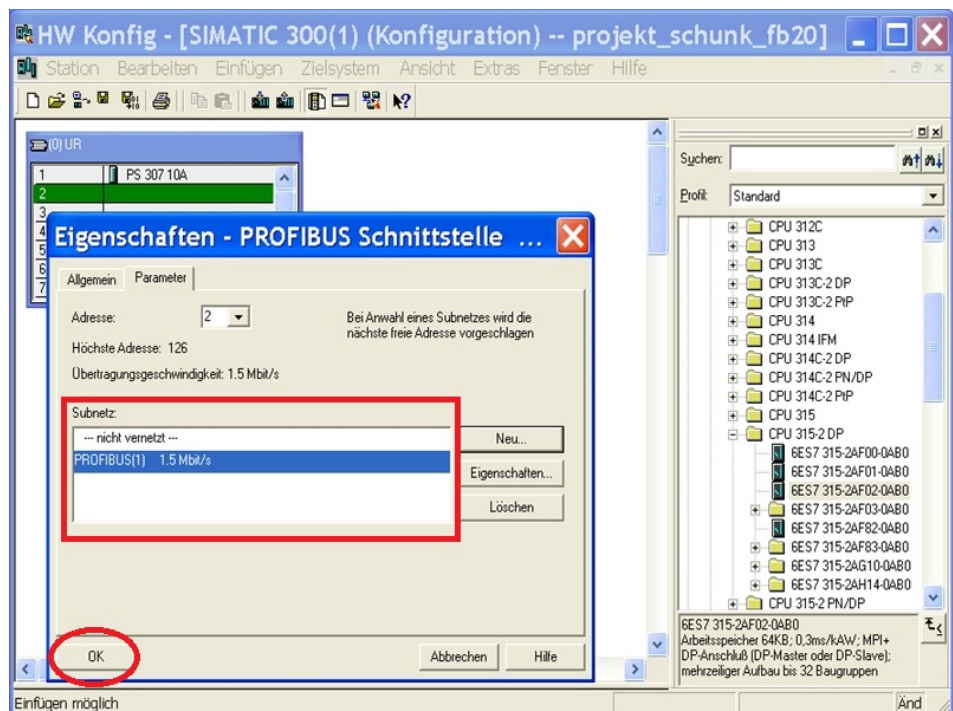


Abb. 33

10 Konfiguration vom Subnetz abgeschlossen

Weiter mit "OK".

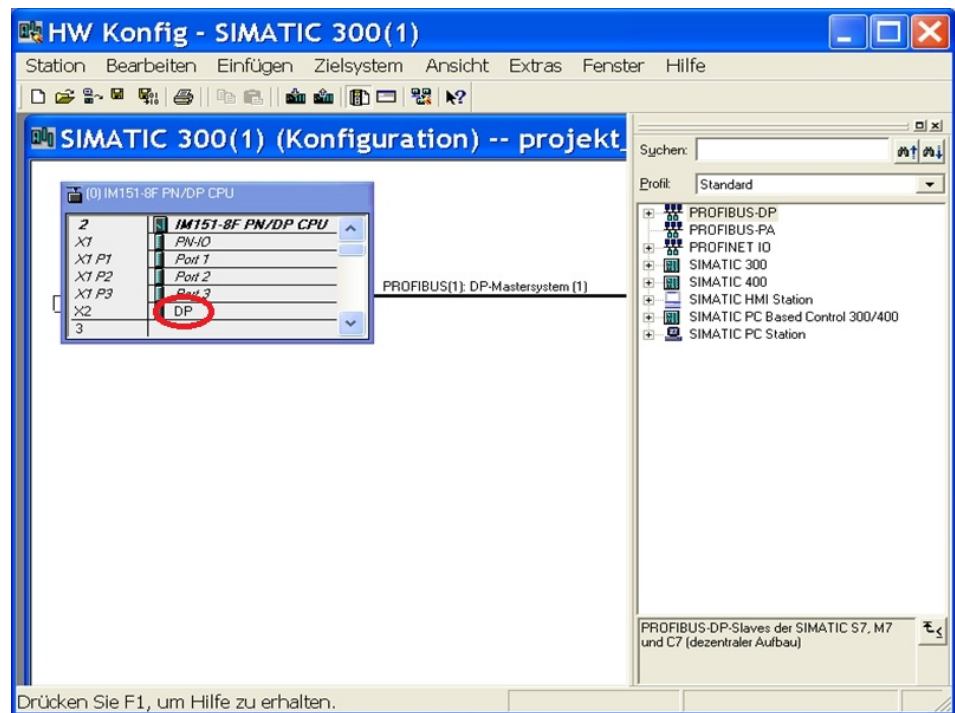


Abb. 34

- 11 Alternativ kann die Konfiguration über den DP- Masterbus eingestellt werden. Mit Doppelklick auf "DP "öffnen sich die **Eigenschaften**.

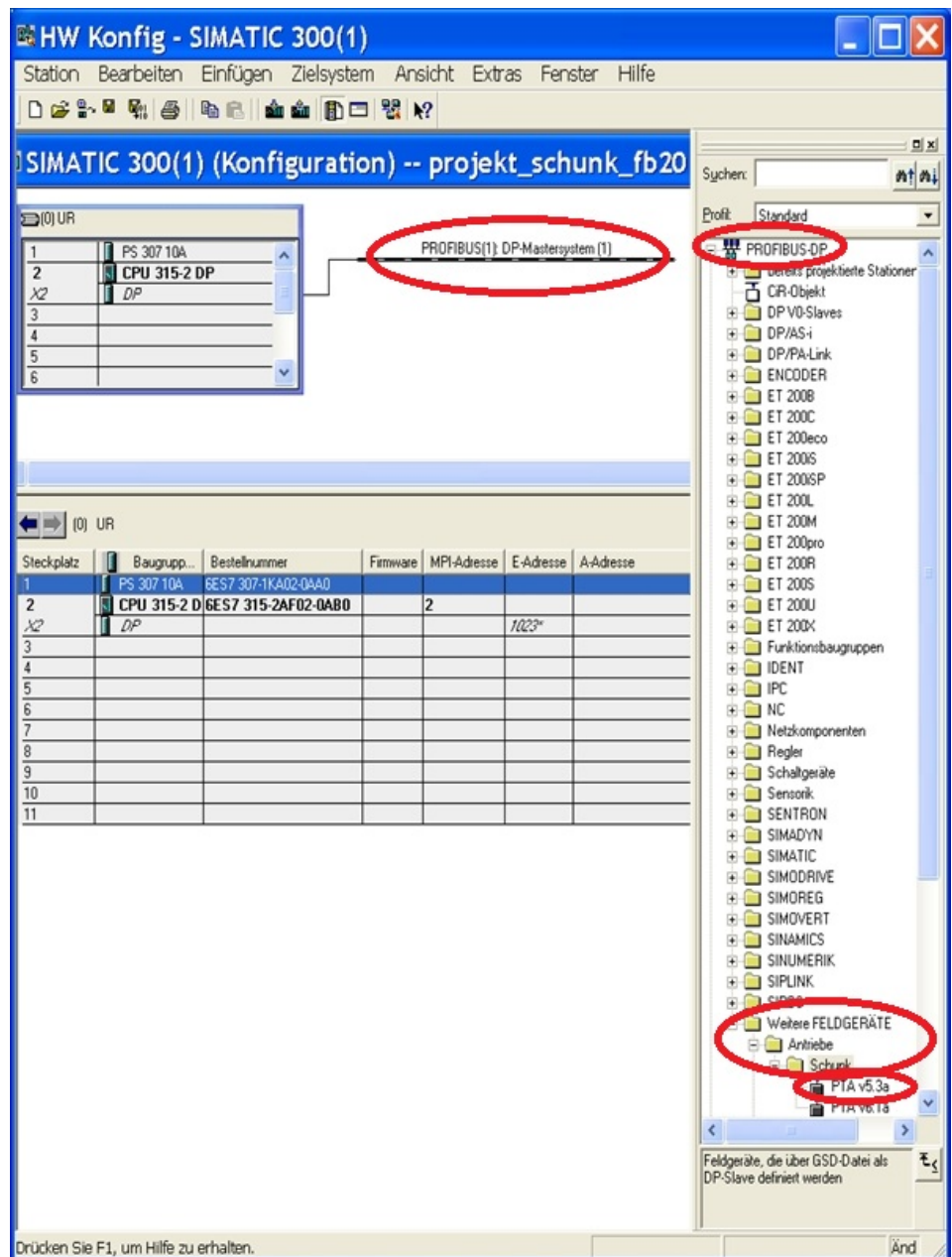


Abb. 35

12 ANTRIEB Schunk PTA v5.3a hinzufügen

Unter **PROFIBUS-DB** weitere **FELDGERÄTE** > **Antriebe** im Ordner **Schunk** **PTA v5.3a** auswählen und per Drag and Drop nach **PROFIBUS(1) DP Mastersystem (1)** ziehen.

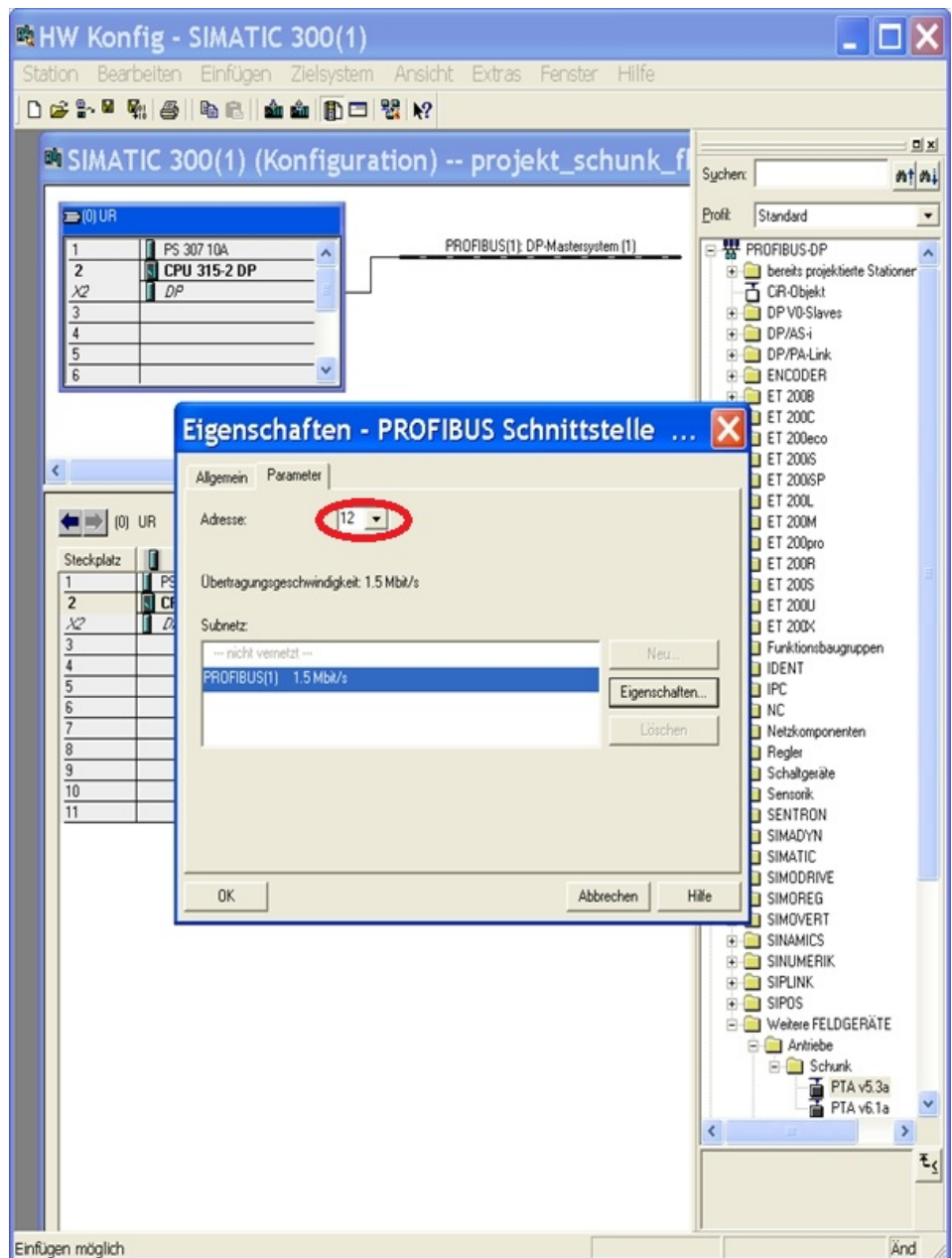


Abb. 36

13 ADRESSE einrichten

ADRESSE der PROFIBUSS Schnittstelle laut Vorgabe des Herstellers einstellen.

Die Adresse des Moduls kann über das MTS "Motion Tool Schunk" entsprechend vorgegeben werden. Weitere Informationen enthält die entsprechende Dokumentation.

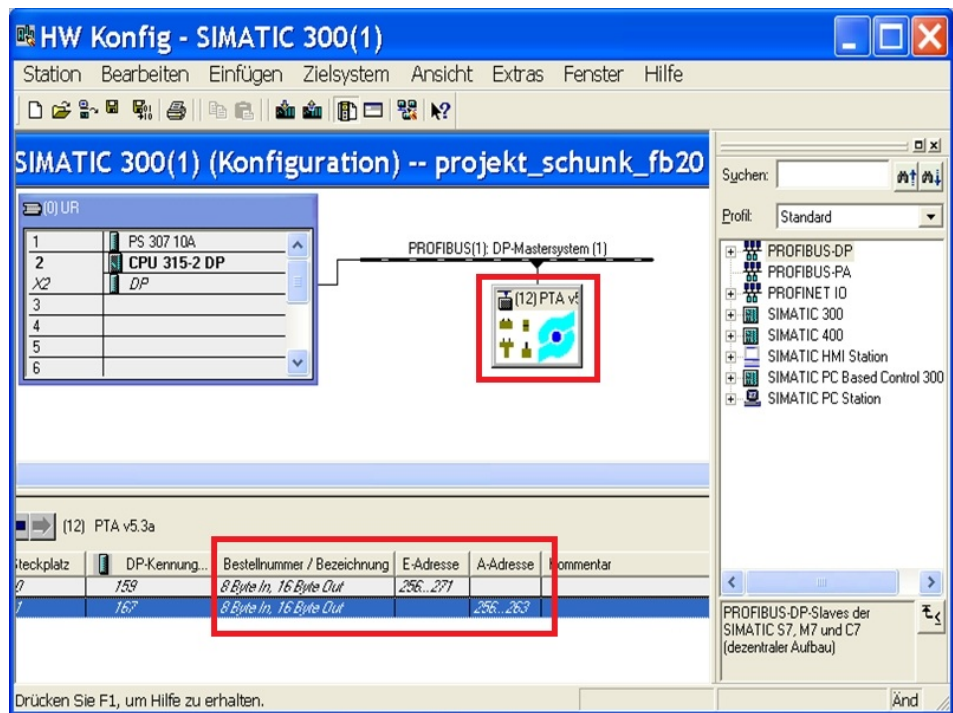


Abb. 37

14 Einstellen der E/A Adressen des Moduls

Die **Start-Adressen** für den Eingangs- und Ausgangsbereich durch einen Doppelklick auf die Adresse konfigurieren.

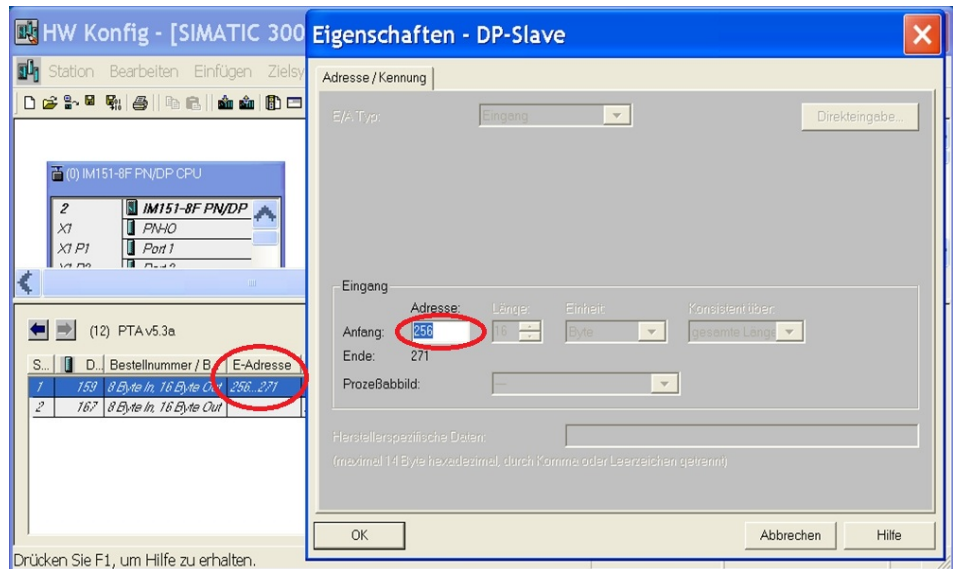


Abb. 38

15 **Doppelklick** auf **E-Adresse** und Einstellung vornehmen.

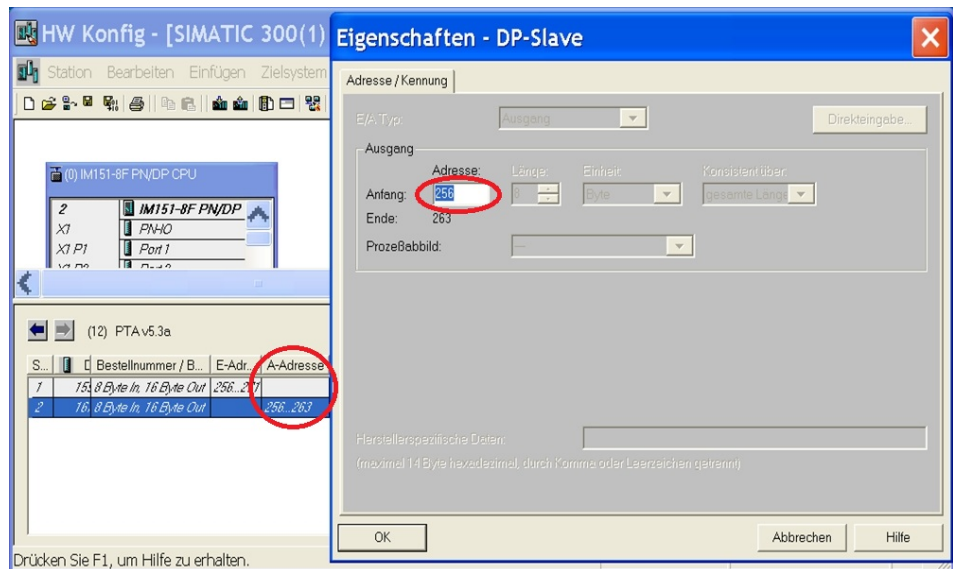


Abb. 39

16 **Doppelklick** auf **A-Adresse** und Einstellung vornehmen.

HINWEIS

Die **Start-Adresse** vom **Ausgang-Bereich** ist **unabhängig** von der Start-Adresse des **Eingang- Bereichs**! SCHUNK empfiehlt den **Eingangs- und Ausgangsbereich** gleich zu wählen.

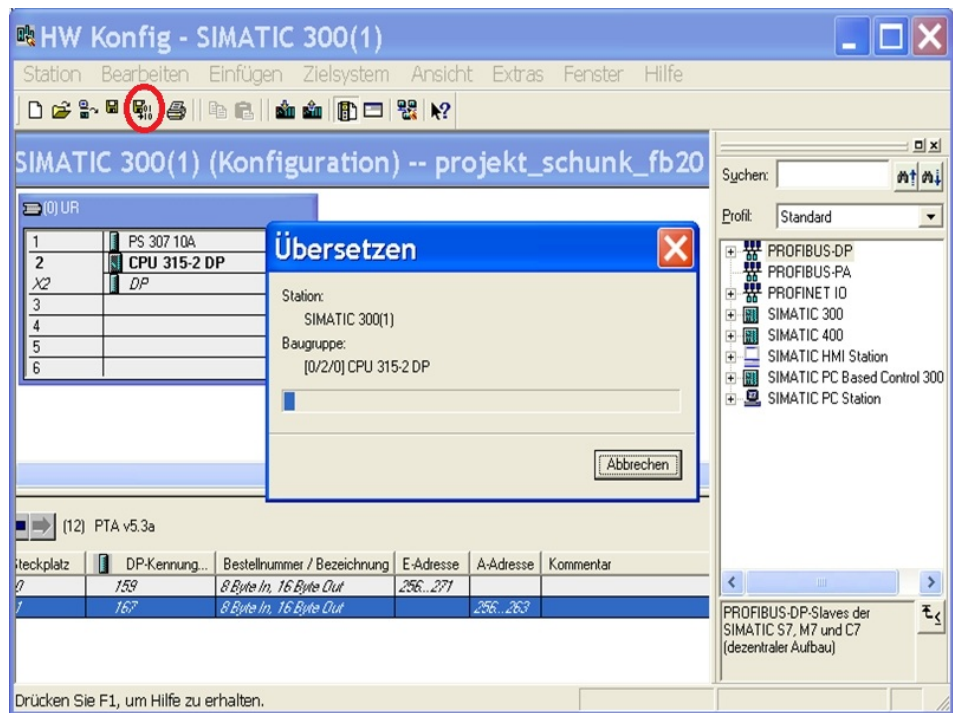


Abb. 40

17 PROJEKT speichern und übersetzen

Speichern und Übersetzen über die Schaltfläche **starten**.

Das Projekt wird gespeichert und für die Übertragung an die SPS vorbereitet.

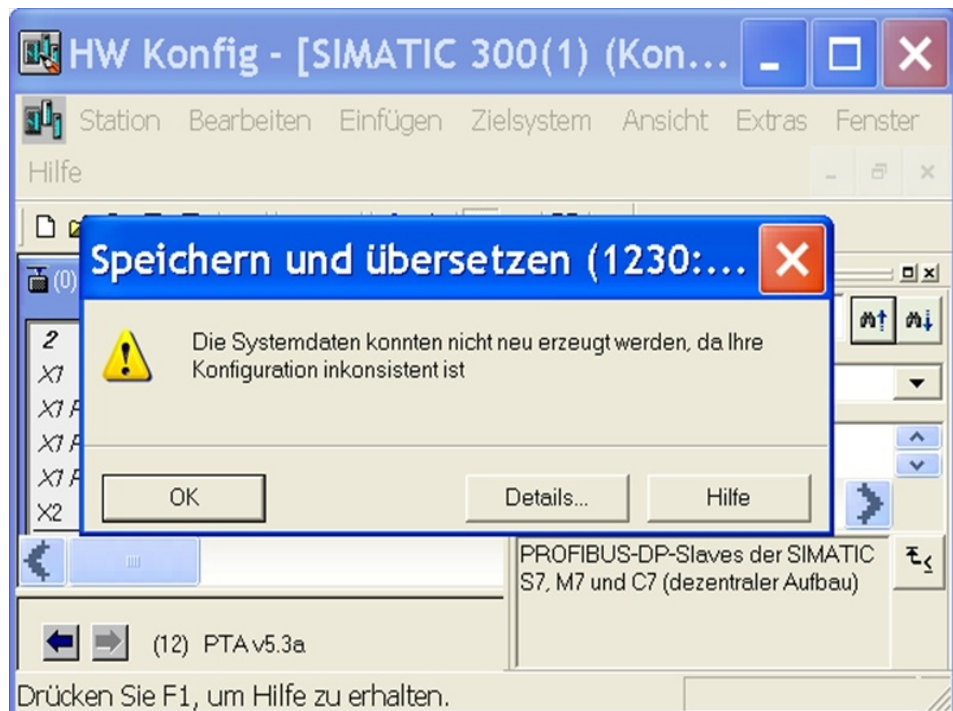


Abb. 41

18 Wenn diese Meldung erscheint gibt es einen Fehler im Projekt / Hardwarekonfiguration.

⇒ Ursache finden und zu beseitigen.

PG/PC Schnittstelle

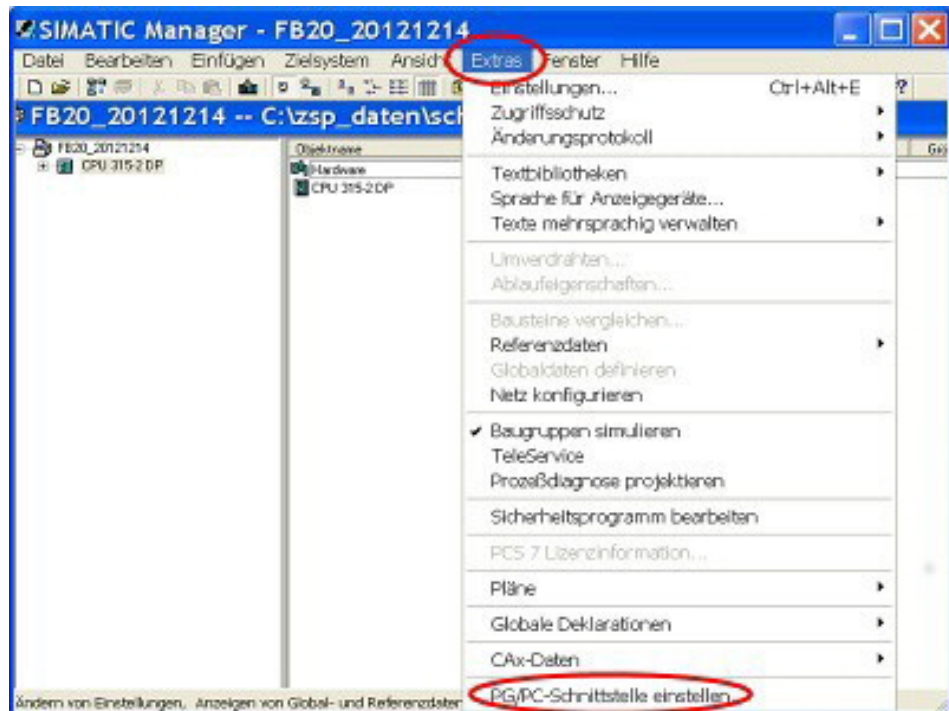


Abb. 42

1 Anschluss SPS am PC

In das Programmfenster SIMATIC Manager wechseln.

2 Über **Extras PG/PC Schnittstelle** die entsprechende Kommunikationshardware auswählen. Diese ist entsprechend einzurichten.

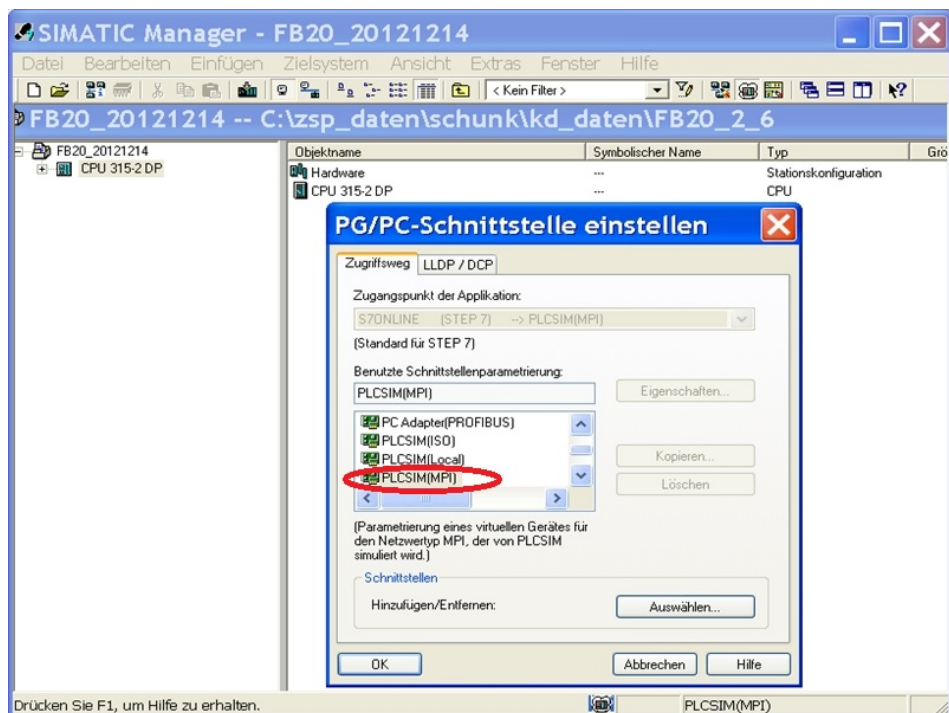


Abb. 43

3 Kommunikationshardware auswählen

Im Beispiel wird die Schnittstelle **PLCSIM(MPI)** ausgewählt.
Die richtige Schnittstelle auszuwählen, die an ihren PC/SPS vorhanden ist.

Der PC/SPS sollte verbunden sein.

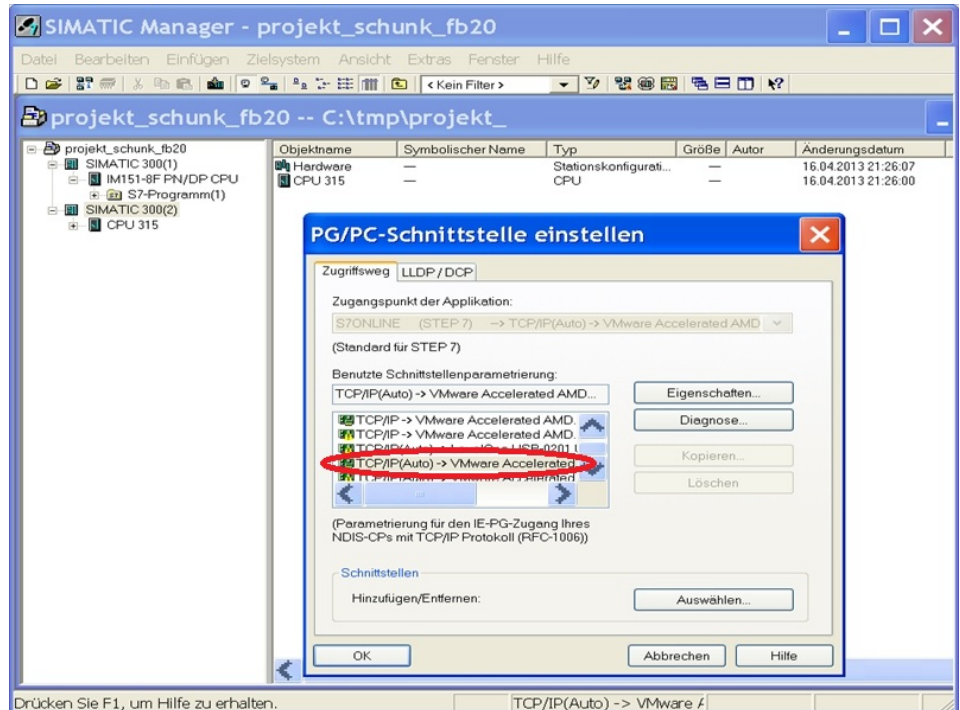


Abb. 44

4 Besteht eine **Verbindung** über **Ethernet**, wie in diesem Beispiel, wird diese Schnittstelle vorgeschlagen:

TCP/IP(Auto) -> VMware Accelerated

Es können div. **Einstellungen** vorgenommen werden. SCHUNK **empfiehlt**, die **Standard** Einstellungen beizubehalten. Sollte keine Verbindung zur Steuerung aufgebaut werden können, müssen sämtliche Einstellungen überprüft werden.

Hardwarekonfiguration übertragen

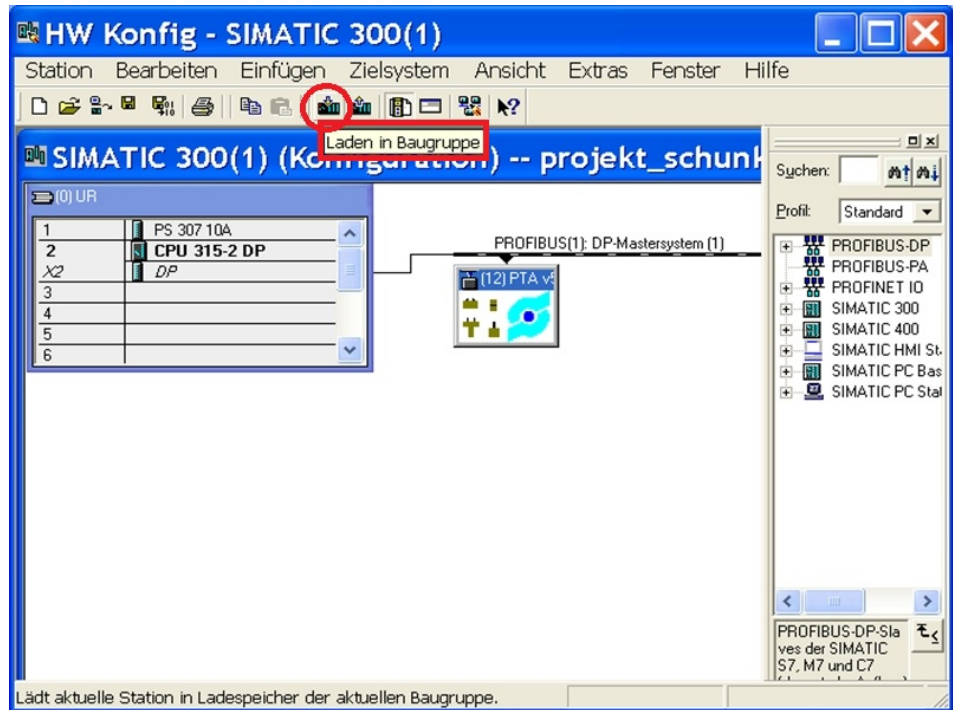


Abb. 45

1 Hardwarekonfiguration in Baugruppe laden

ACHTUNG: Die SPS muss sich in "Stop" befinden.

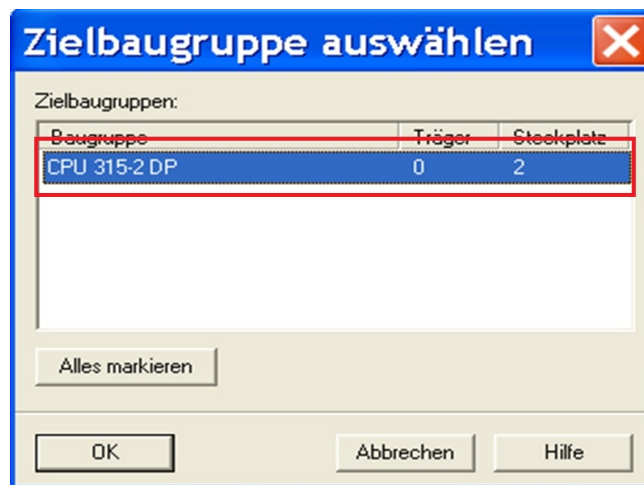


Abb. 46

2 ZIELBAUGRUPPE auswählen

Baugruppe auswählen und mit "OK" bestätigen.

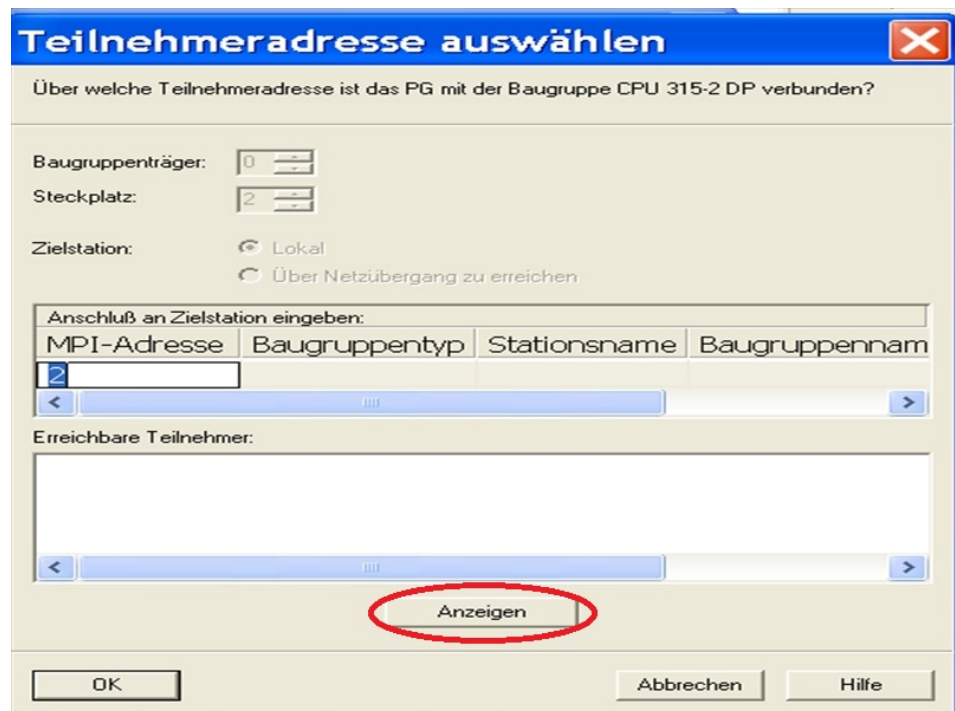


Abb. 47

3 Erreichbare Teilnehmer

Teilnehmer werden durch klicken auf dem Button "Anzeigen" ermittelt.

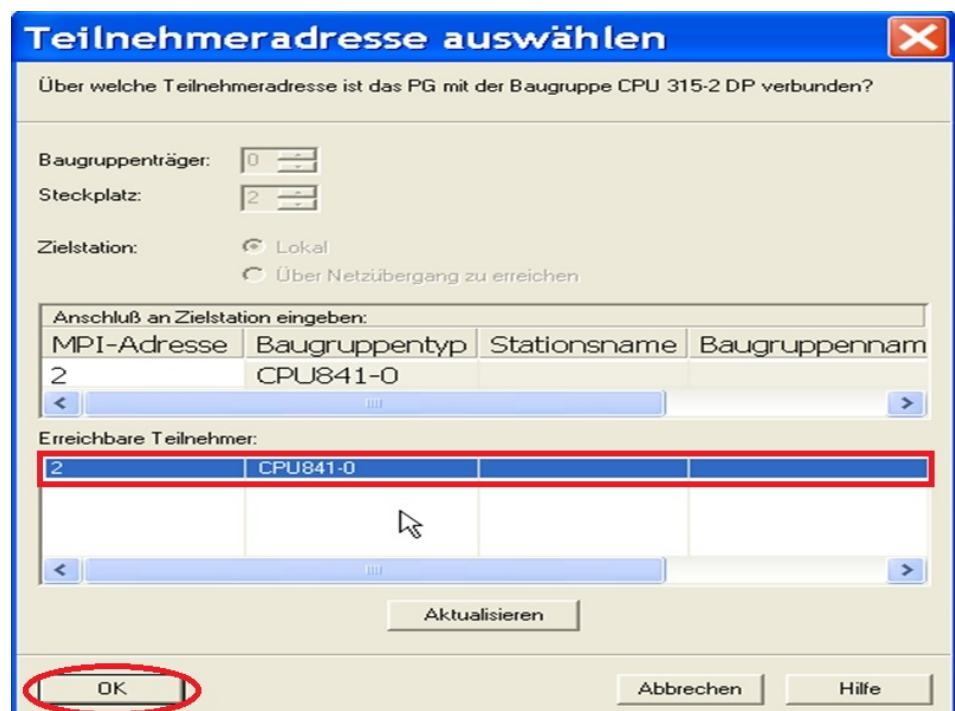


Abb. 48

4 Teilnehmeradresse auswählen

Ermittelten Teilnehmer markieren und mit "OK" bestätigen.

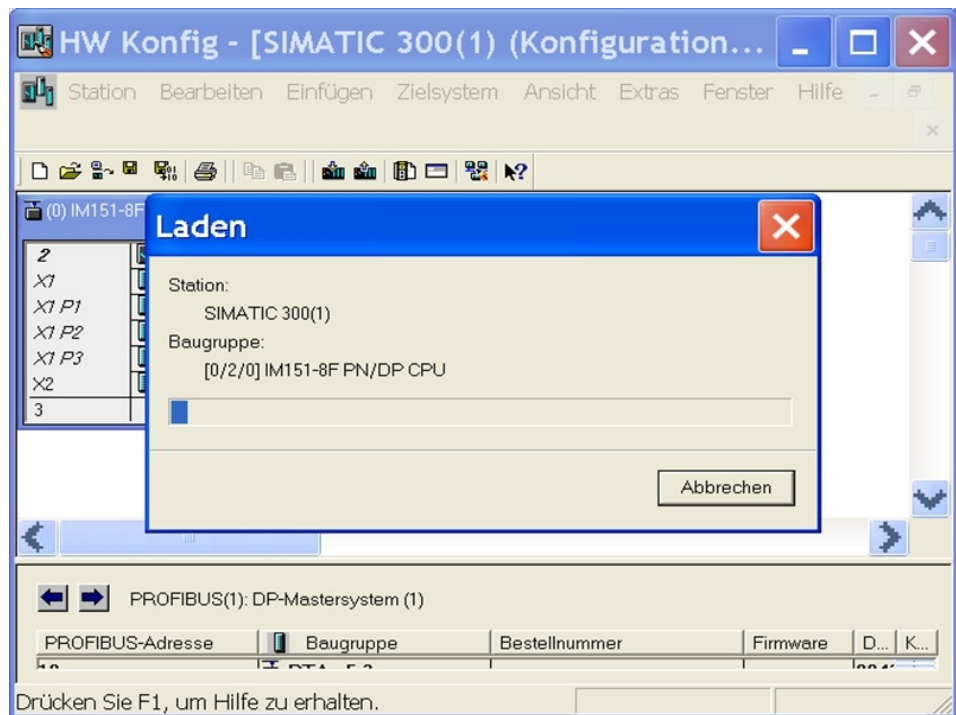


Abb. 49

5 **Hardwarekonfiguration** wird in die SPS **übertragen**.

Sollte nachgefragt werden, ob die SPS/CPU in Stop gesetzt werden sollte, befand sich die SPS vorher nicht in Stop oder es wurde die falsche Steuerung ausgewählt. Evtl. ist die Übertragung abubrechen.

In diesem Beispiel wird davon ausgegangen, dass sich in der SPS noch kein SPS Programm befindet.

Wenn nun die SPS auf Run geschaltet wird und alles fehlerfrei ist, inkl. der angeschlossenen Hardware, sollte das System ohne Störung in Run gehen und bleiben.

Software einspielen In dem Musterprojekt sind alle Bausteine vorhanden die wichtig sind, um das Modul über die SPS anzusteuern. Bei der nun folgenden Beschreibung wird von folgenden Voraussetzungen ausgegangen:

- Muster Projekt von Schunk ist entpackt.
- Das oben beschriebene Projekt wurde erstellt und funktioniert.
- In den erstellten Projekten sind keine weiteren Daten enthalten und wurden gesichert.

Umkopieren Grundprogramm

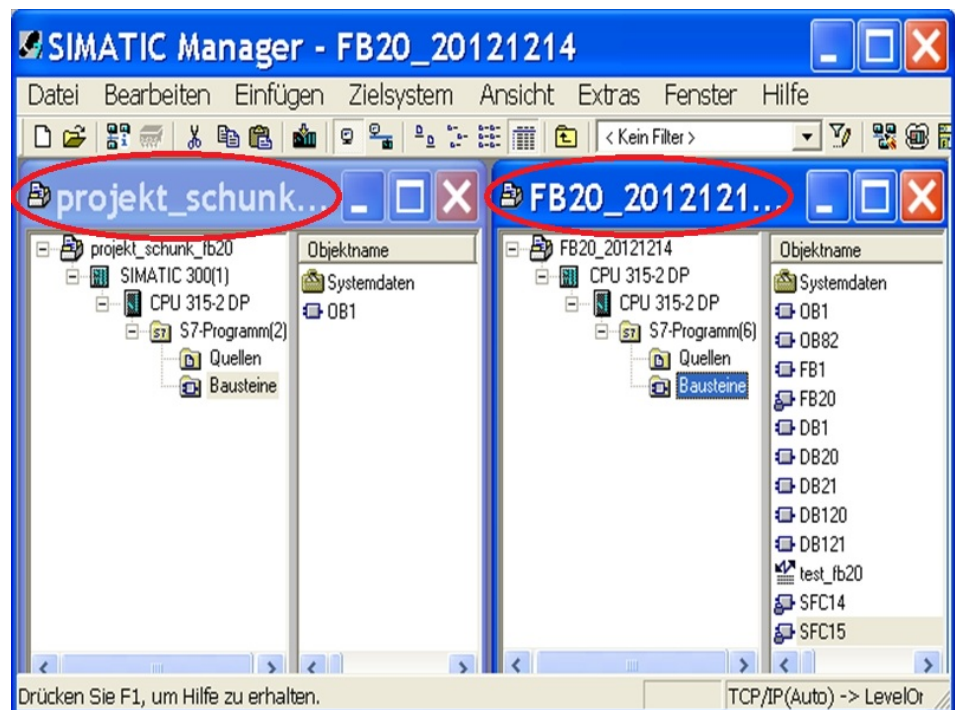


Abb. 50

1 Projekte öffnen

Erstellte Projekte und **Musterprojekte** von Schunk sind **geöffnet** und die Fenster sind nebeneinander angeordnet.

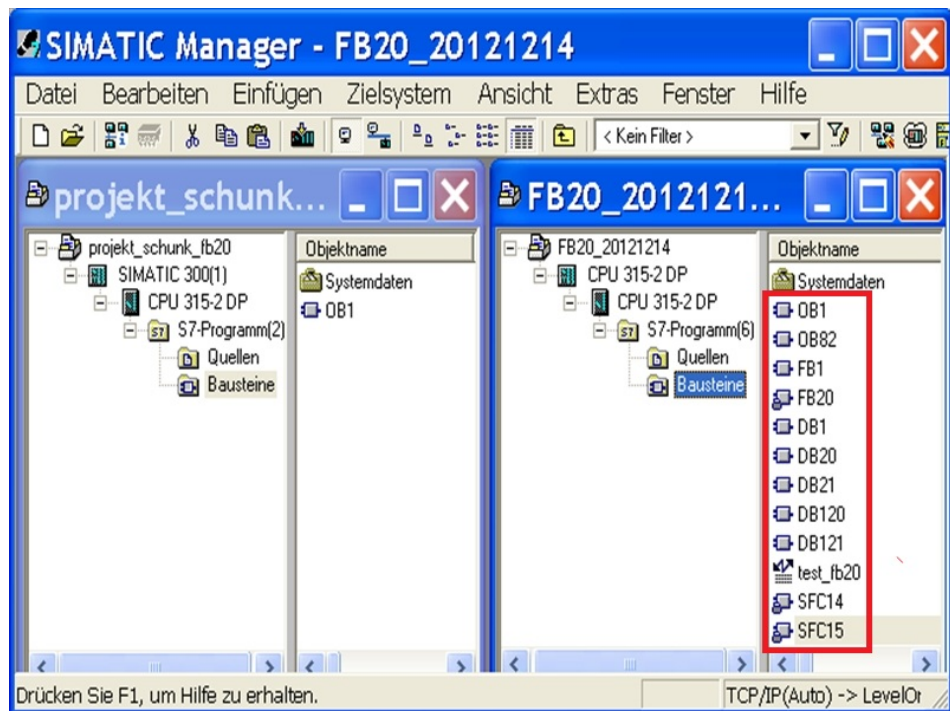


Abb. 51

2 Bausteine kopieren

Bausteine OB1 bis SFC15 im Musterprojekt Schunk (rechtes Fenster) markieren und per Drag and Drop in das aktuelle Projekt ziehen. Dabei kann der vorhandene OB1 überschrieben werden.

⇒ **ACHTUNG:** Kopieren **ohne Systemdaten**.

Beim Übertragen der Systemdaten kann es zu **Schwierigkeiten** kommen, da diese **speziell** für die jeweilige **Hardwarekonfiguration** erzeugt wird.

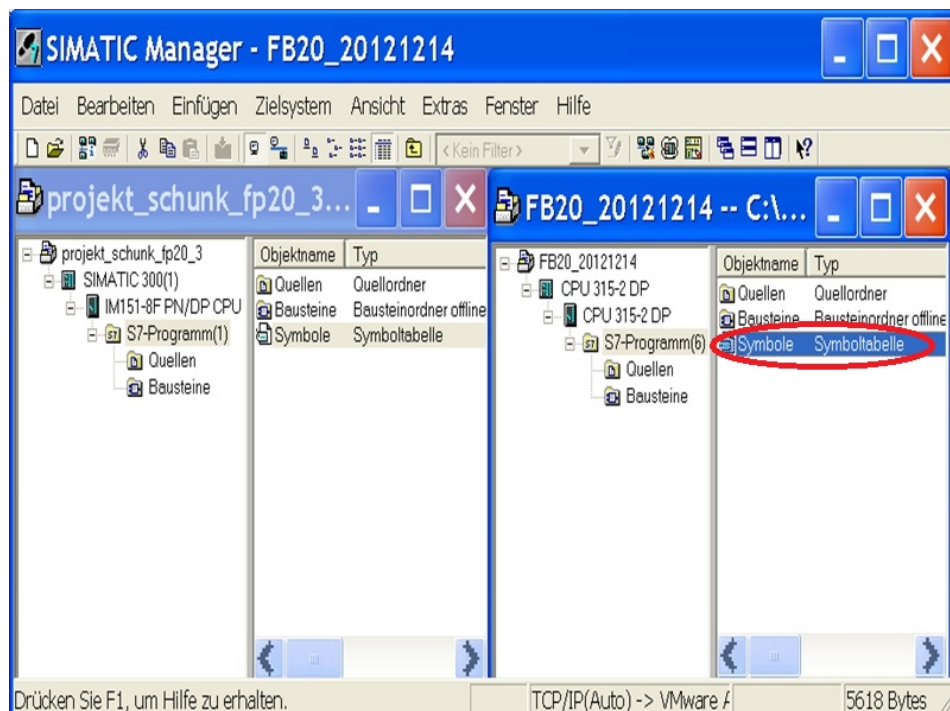


Abb. 52

- 3 Symbolik-Ordner umkopieren. Dabei kann der vorhandene Symbolordner überschrieben werden, wenn dieser nicht bearbeitet wurde. Ansonsten könnten Daten verloren gehen!

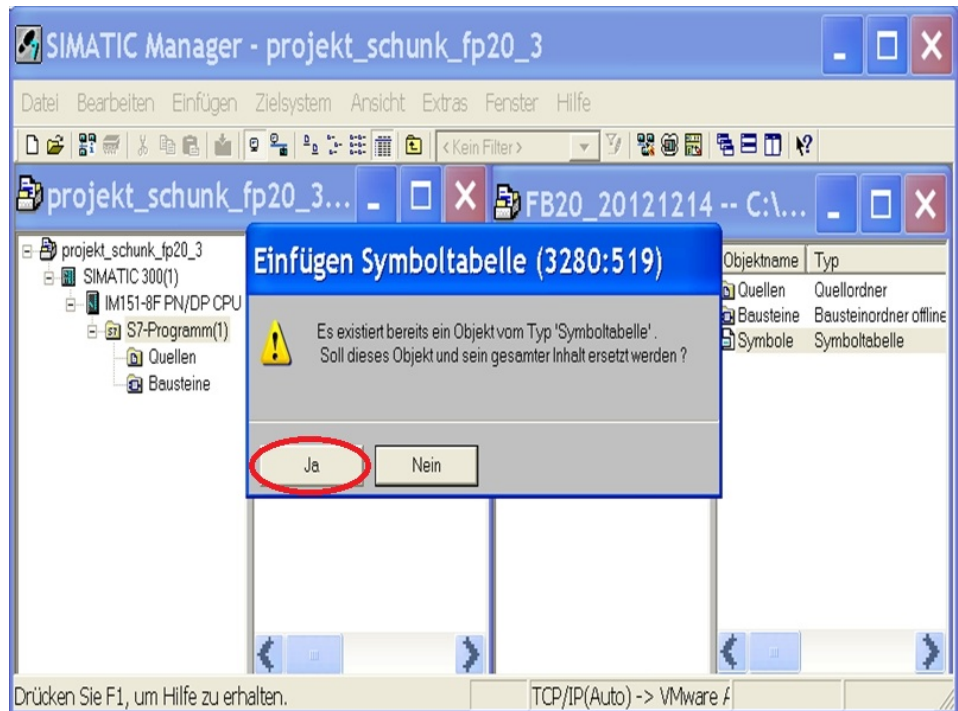


Abb. 53

- 4 Objekt und gesamten Inhalt ersetzen.

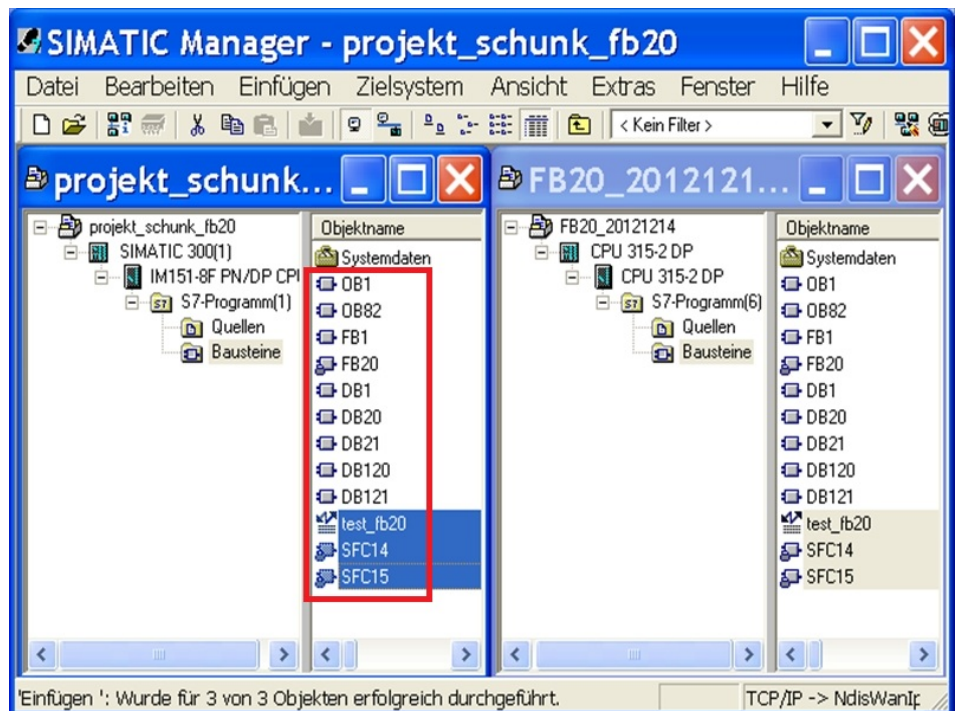


Abb. 54

- 5 **Das Musterprojekt schließen**

Die Bausteine wurden kopiert.

⇒ Das Musterprojekt kann nun geschlossen werden.

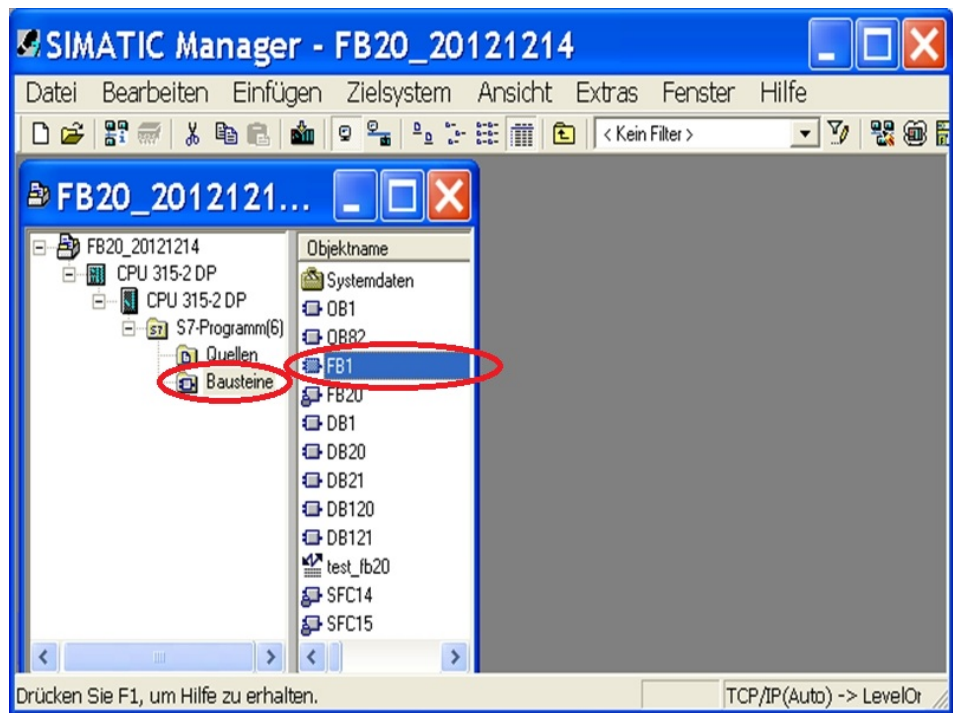


Abb. 55

- 6 **Anpassungen im SPS Programm**
Baustein **FB1** vom Projekt öffnen.

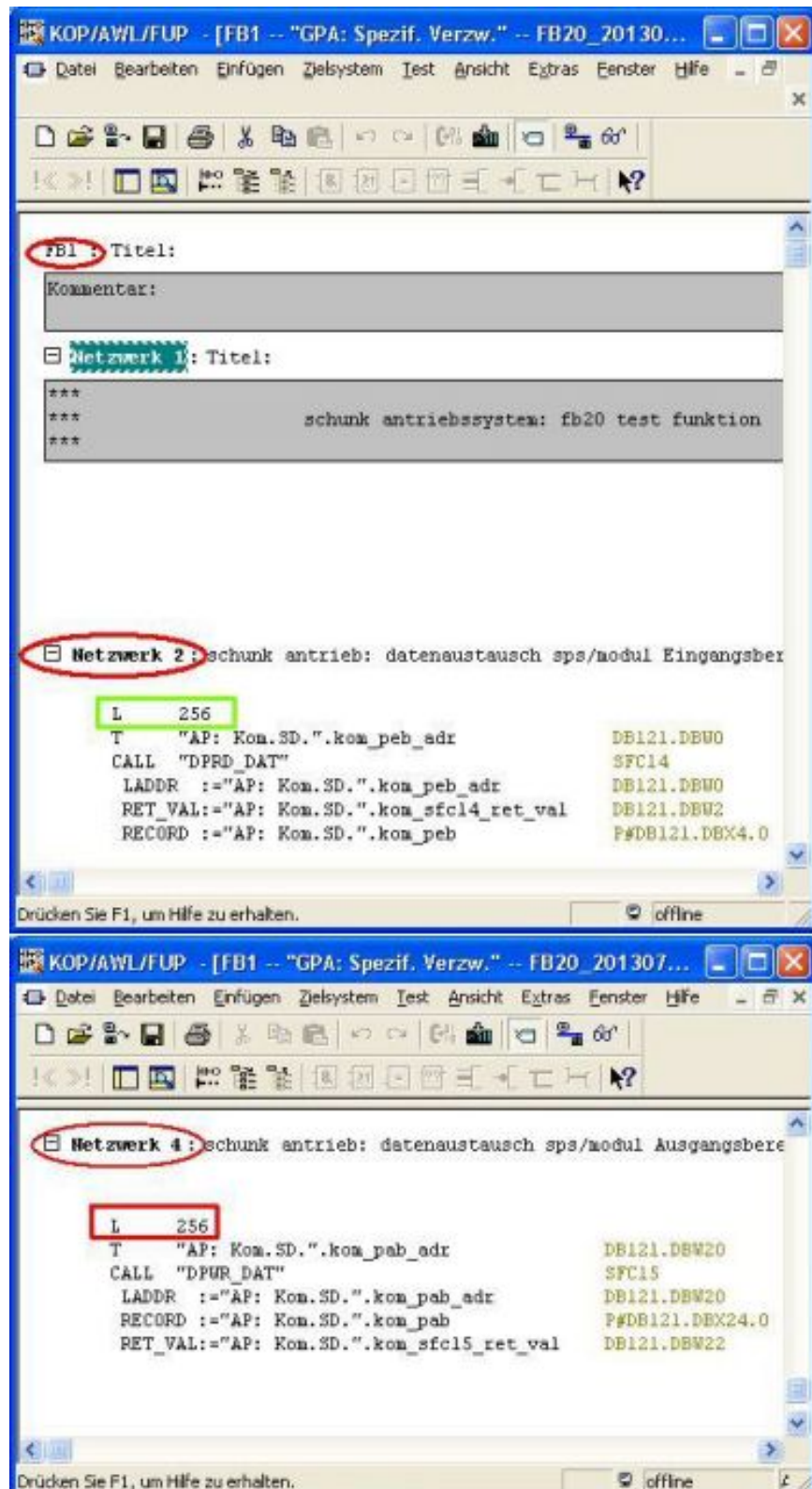


Abb. 56

7 Adressen anpassen

Die **Adressierungen** der **Ein- und Ausgänge**, die bei der Hardwarekonfiguration am SCHUNK-Modul eingestellt wurden, sind im **FB1** Netzwerk2 und Netzwerk4 einzustellen.

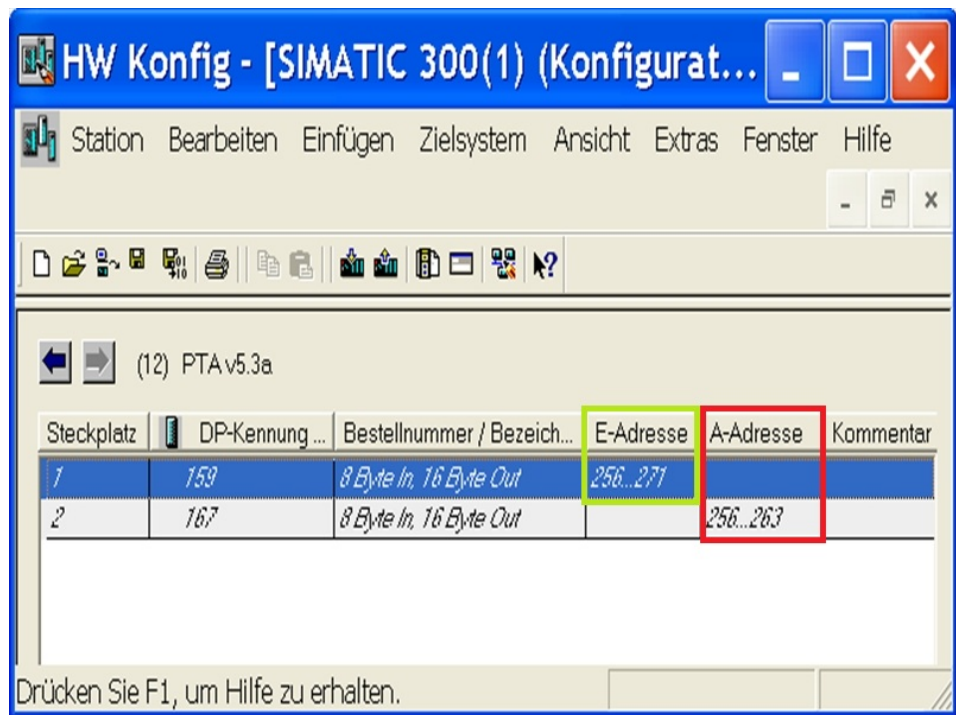


Abb. 57

8 Der KOP/AWL/FUP Editor kann geschossen werden.

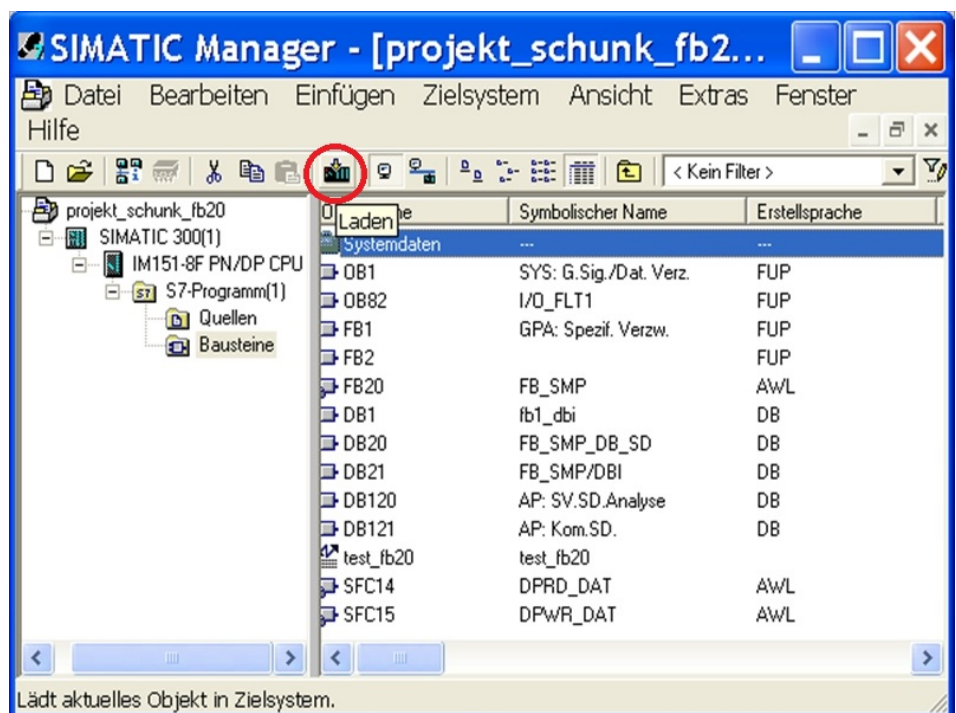


Abb. 58

9 Bausteine in die CPU laden

ACHTUNG: CPU auf Status **STOP** setzen um Störungen und unkontrollierte Bewegungen zu vermeiden. Dazu kann z.B. der RUN/STOP Schalter an der CPU auf **STOP** gestellt werden. Die Bausteine von OB1 bis DB121 sind zu markieren. Mit dem Anklicken des Icon in der Menüleiste **Symbol**, wird das Programm übertragen.



Abb. 59

- 10 **ACHTUNG:** Sollten zu kopierende Bausteine in der CPU vorhanden sein, werden diese nach bestätigen der Meldung überschrieben und können nicht wiederhergestellt werden.

Die CPU sollte kein Programm enthalten (Lauf- Voraussetzung): Bevor diese Meldung bestätigt wird, sollten Sie sich ganz sicher sein, ob Sie mit der richtigen SPS verbunden sind und ob wirklich die Bausteine überschrieben werden sollen.

- ⇒ **Wenn sicher gestellt ist, dass durch eine Bewegung der Baugruppen kein Schaden entsteht, kann die SPS auf Run geschaltet werden.** Sollte alles korrekt gemacht worden sein, müsste das System ohne Störung hochlaufen und im Run- Modus bleiben. Ansonsten sollten die Fehler beseitigt werden.

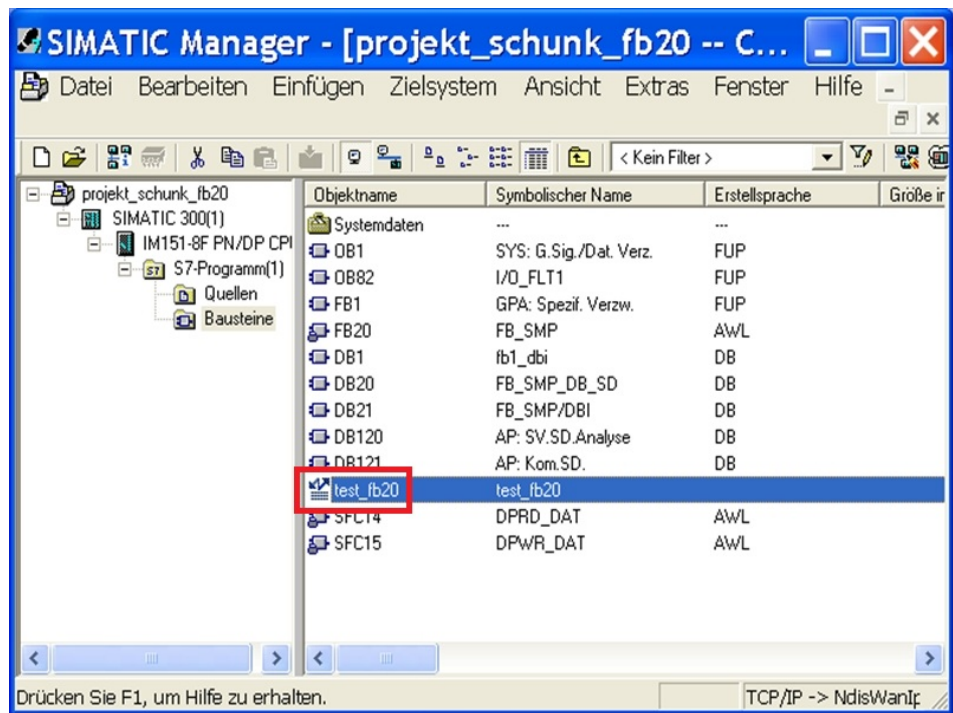
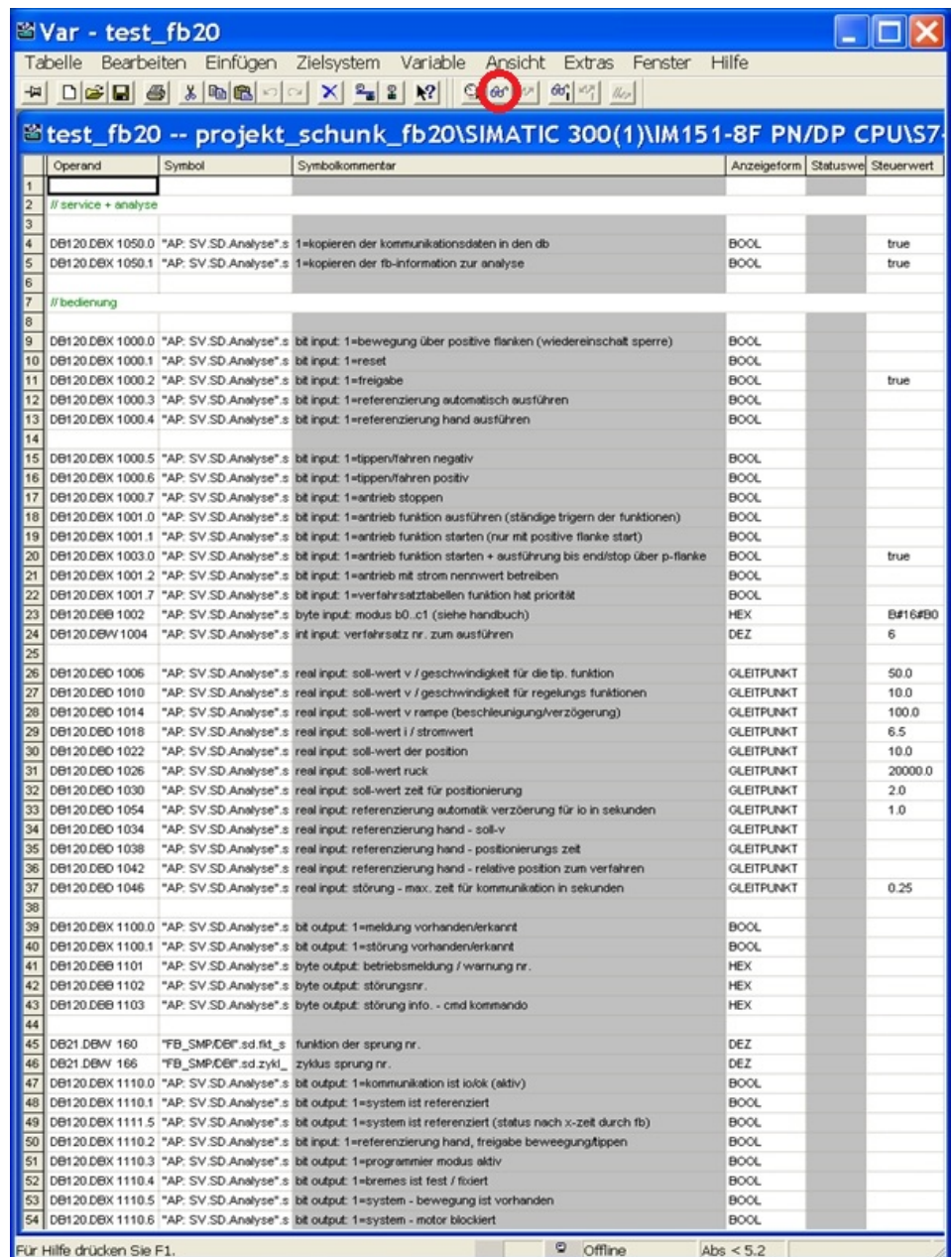


Abb. 60

11 Variablentabelle mit Doppelklick öffnen


Über die **Variablentabelle** kann das **FB20/Modul** von **Hand angesteuert** werden. Dabei ist vorher sicher zu stellen, dass durch eine Bewegung des Moduls kein Schaden entstehen kann.

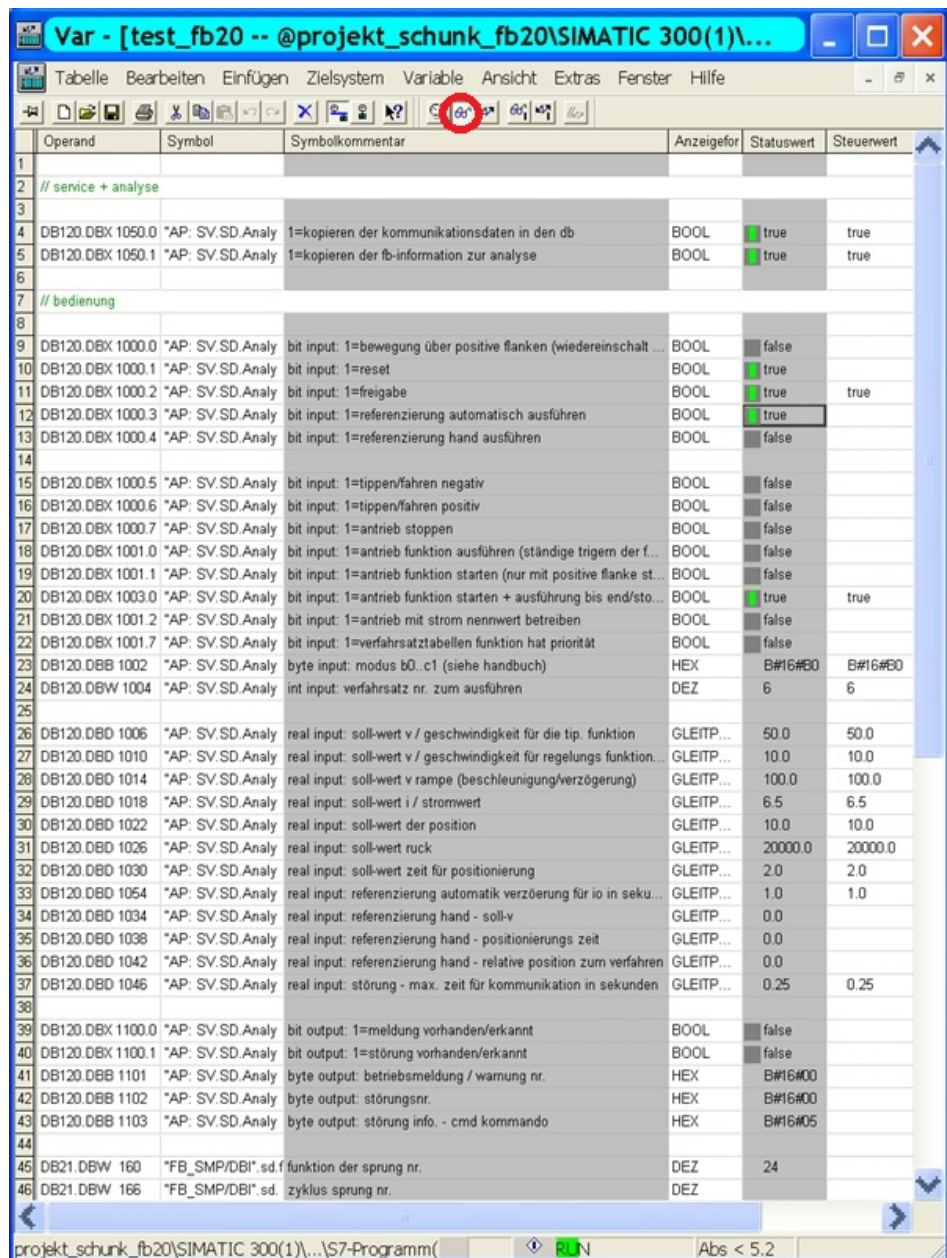


Operand	Symbol	Symbolkommentar	Anzeigeform	Statuswert	Steuerwert
1					
2	// service + analyse				
3					
4	DB120.DBX 1050.0	"AP: SV_SD_Analyse".s	1=kopieren der kommunikationsdaten in den db	BOOL	true
5	DB120.DBX 1050.1	"AP: SV_SD_Analyse".s	1=kopieren der fb-information zur analyse	BOOL	true
6					
7	// bedienung				
8					
9	DB120.DBX 1000.0	"AP: SV_SD_Analyse".s	bit input: 1=bewegung über positive flanken (wiedereinschalt sperre)	BOOL	
10	DB120.DBX 1000.1	"AP: SV_SD_Analyse".s	bit input: 1=reset	BOOL	
11	DB120.DBX 1000.2	"AP: SV_SD_Analyse".s	bit input: 1= freigabe	BOOL	true
12	DB120.DBX 1000.3	"AP: SV_SD_Analyse".s	bit input: 1=referenzierung automatisch ausführen	BOOL	
13	DB120.DBX 1000.4	"AP: SV_SD_Analyse".s	bit input: 1=referenzierung hand ausführen	BOOL	
14					
15	DB120.DBX 1000.5	"AP: SV_SD_Analyse".s	bit input: 1=tippen/fahren negativ	BOOL	
16	DB120.DBX 1000.6	"AP: SV_SD_Analyse".s	bit input: 1=tippen/fahren positiv	BOOL	
17	DB120.DBX 1000.7	"AP: SV_SD_Analyse".s	bit input: 1=antrieb stoppen	BOOL	
18	DB120.DBX 1001.0	"AP: SV_SD_Analyse".s	bit input: 1=antrieb funktion ausführen (ständige triggern der funktionen)	BOOL	
19	DB120.DBX 1001.1	"AP: SV_SD_Analyse".s	bit input: 1=antrieb funktion starten (nur mit positive flanke start)	BOOL	
20	DB120.DBX 1003.0	"AP: SV_SD_Analyse".s	bit input: 1=antrieb funktion starten + ausführung bis end/stop über p-flanke	BOOL	true
21	DB120.DBX 1001.2	"AP: SV_SD_Analyse".s	bit input: 1=antrieb mit strom nennwert betreiben	BOOL	
22	DB120.DBX 1001.7	"AP: SV_SD_Analyse".s	bit input: 1=verfahrenstabelle funktion hat priorität	BOOL	
23	DB120.DBB 1002	"AP: SV_SD_Analyse".s	byte input: modus b0..c1 (siehe handbuch)	HEX	B#16#B0
24	DB120.DBW 1004	"AP: SV_SD_Analyse".s	int input: verfahrenszahl nr. zum ausführen	DEZ	6
25					
26	DB120.DBD 1006	"AP: SV_SD_Analyse".s	real input: soll-wert v / geschwindigkeit für die tp. funktion	GLEITPUNKT	50.0
27	DB120.DBD 1010	"AP: SV_SD_Analyse".s	real input: soll-wert v / geschwindigkeit für regelungs funktionen	GLEITPUNKT	10.0
28	DB120.DBD 1014	"AP: SV_SD_Analyse".s	real input: soll-wert v rampe (beschleunigung/verzögerung)	GLEITPUNKT	100.0
29	DB120.DBD 1018	"AP: SV_SD_Analyse".s	real input: soll-wert i / stromwert	GLEITPUNKT	6.5
30	DB120.DBD 1022	"AP: SV_SD_Analyse".s	real input: soll-wert der position	GLEITPUNKT	10.0
31	DB120.DBD 1026	"AP: SV_SD_Analyse".s	real input: soll-wert ruck	GLEITPUNKT	20000.0
32	DB120.DBD 1030	"AP: SV_SD_Analyse".s	real input: soll-wert zeit für positionierung	GLEITPUNKT	2.0
33	DB120.DBD 1054	"AP: SV_SD_Analyse".s	real input: referenzierung automatik verzögerung für io in sekunden	GLEITPUNKT	1.0
34	DB120.DBD 1034	"AP: SV_SD_Analyse".s	real input: referenzierung hand - soll-v	GLEITPUNKT	
35	DB120.DBD 1038	"AP: SV_SD_Analyse".s	real input: referenzierung hand - positionierungs zeit	GLEITPUNKT	
36	DB120.DBD 1042	"AP: SV_SD_Analyse".s	real input: referenzierung hand - relative position zum verfahren	GLEITPUNKT	
37	DB120.DBD 1046	"AP: SV_SD_Analyse".s	real input: störung - max. zeit für kommunikation in sekunden	GLEITPUNKT	0.25
38					
39	DB120.DBX 1100.0	"AP: SV_SD_Analyse".s	bit output: 1=meldung vorhanden/erkannt	BOOL	
40	DB120.DBX 1100.1	"AP: SV_SD_Analyse".s	bit output: 1=störung vorhanden/erkannt	BOOL	
41	DB120.DBB 1101	"AP: SV_SD_Analyse".s	byte output: betriebsmeldung / warnung nr.	HEX	
42	DB120.DBB 1102	"AP: SV_SD_Analyse".s	byte output: störungsnr.	HEX	
43	DB120.DBB 1103	"AP: SV_SD_Analyse".s	byte output: störung info. - cmd kommando	HEX	
44					
45	DB21.DBW 160	"FB_SMP/DBI".sd.fkt_nr.	funktion der sprung nr.	DEZ	
46	DB21.DBW 166	"FB_SMP/DBI".sd.zykl.	zyklus sprung nr.	DEZ	
47	DB120.DBX 1110.0	"AP: SV_SD_Analyse".s	bit output: 1=kommunikation ist lokal (aktiv)	BOOL	
48	DB120.DBX 1110.1	"AP: SV_SD_Analyse".s	bit output: 1=system ist referenziert	BOOL	
49	DB120.DBX 1111.5	"AP: SV_SD_Analyse".s	bit output: 1=system ist referenziert (status nach x-zeit durch fb)	BOOL	
50	DB120.DBX 1110.2	"AP: SV_SD_Analyse".s	bit input: 1=referenzierung hand, freigabe bewegungstippen	BOOL	
51	DB120.DBX 1110.3	"AP: SV_SD_Analyse".s	bit output: 1=programmier modus aktiv	BOOL	
52	DB120.DBX 1110.4	"AP: SV_SD_Analyse".s	bit output: 1=bremse ist fest / fixiert	BOOL	
53	DB120.DBX 1110.5	"AP: SV_SD_Analyse".s	bit output: 1=system - bewegung ist vorhanden	BOOL	
54	DB120.DBX 1110.6	"AP: SV_SD_Analyse".s	bit output: 1=system - motor blockiert	BOOL	

Abb. 61

12 Variable beobachten

Mit Variable Tabelle online gehen, über Symbol  **Brille** aktivieren.



Operand	Symbol	Symbolkommentar	Anzeigeform	Statuswert	Steuwert
1					
2	// service + analyse				
3					
4	DB120.DBX 1050.0	*AP: SV.SD.Analy 1=kopieren der kommunikationsdaten in den db	BOOL	true	true
5	DB120.DBX 1050.1	*AP: SV.SD.Analy 1=kopieren der fb-information zur analyse	BOOL	true	true
6					
7	// bedienung				
8					
9	DB120.DBX 1000.0	*AP: SV.SD.Analy bit input: 1=bewegung über positive flanken (wiedereinschalt ...	BOOL	false	
10	DB120.DBX 1000.1	*AP: SV.SD.Analy bit input: 1=reset	BOOL	true	
11	DB120.DBX 1000.2	*AP: SV.SD.Analy bit input: 1= freigabe	BOOL	true	true
12	DB120.DBX 1000.3	*AP: SV.SD.Analy bit input: 1=referenzierung automatisch ausführen	BOOL	true	
13	DB120.DBX 1000.4	*AP: SV.SD.Analy bit input: 1=referenzierung hand ausführen	BOOL	false	
14					
15	DB120.DBX 1000.5	*AP: SV.SD.Analy bit input: 1=tippen/fahren negativ	BOOL	false	
16	DB120.DBX 1000.6	*AP: SV.SD.Analy bit input: 1=tippen/fahren positiv	BOOL	false	
17	DB120.DBX 1000.7	*AP: SV.SD.Analy bit input: 1=antrieb stoppen	BOOL	false	
18	DB120.DBX 1001.0	*AP: SV.SD.Analy bit input: 1=antrieb funktion ausführen (ständige triger der f...	BOOL	false	
19	DB120.DBX 1001.1	*AP: SV.SD.Analy bit input: 1=antrieb funktion starten (nur mit positive flanke st...	BOOL	false	
20	DB120.DBX 1003.0	*AP: SV.SD.Analy bit input: 1=antrieb funktion starten + ausführung bis end/sto...	BOOL	true	true
21	DB120.DBX 1001.2	*AP: SV.SD.Analy bit input: 1=antrieb mit strom nenwert betreiben	BOOL	false	
22	DB120.DBX 1001.7	*AP: SV.SD.Analy bit input: 1=verfahrenstabelle funktion hat priorität	BOOL	false	
23	DB120.DBB 1002	*AP: SV.SD.Analy byte input: modus b0_c1 (siehe handbuch)	HEX	B#16#B0	B#16#B0
24	DB120.DBW 1004	*AP: SV.SD.Analy int input: verfahrenszahl nr. zum ausführen	DEZ	6	6
25					
26	DB120.DBD 1006	*AP: SV.SD.Analy real input: soll-wert v / geschwindigkeit für die tip. funktion	GLEITP...	50.0	50.0
27	DB120.DBD 1010	*AP: SV.SD.Analy real input: soll-wert v / geschwindigkeit für regelungs funktion...	GLEITP...	10.0	10.0
28	DB120.DBD 1014	*AP: SV.SD.Analy real input: soll-wert v rampe (beschleunigung/verzögerung)	GLEITP...	100.0	100.0
29	DB120.DBD 1018	*AP: SV.SD.Analy real input: soll-wert i / stromwert	GLEITP...	6.5	6.5
30	DB120.DBD 1022	*AP: SV.SD.Analy real input: soll-wert der position	GLEITP...	10.0	10.0
31	DB120.DBD 1026	*AP: SV.SD.Analy real input: soll-wert ruck	GLEITP...	20000.0	20000.0
32	DB120.DBD 1030	*AP: SV.SD.Analy real input: soll-wert zeit für positionierung	GLEITP...	2.0	2.0
33	DB120.DBD 1054	*AP: SV.SD.Analy real input: referenzierung automatik verzögerung für io in seku...	GLEITP...	1.0	1.0
34	DB120.DBD 1034	*AP: SV.SD.Analy real input: referenzierung hand - soll-v	GLEITP...	0.0	
35	DB120.DBD 1038	*AP: SV.SD.Analy real input: referenzierung hand - positionierungs zeit	GLEITP...	0.0	
36	DB120.DBD 1042	*AP: SV.SD.Analy real input: referenzierung hand - relative position zum verfahren	GLEITP...	0.0	
37	DB120.DBD 1046	*AP: SV.SD.Analy real input: störung - max. zeit für kommunikation in sekunden	GLEITP...	0.25	0.25
38					
39	DB120.DBX 1100.0	*AP: SV.SD.Analy bit output: 1=meldung vorhanden/erkannt	BOOL	false	
40	DB120.DBX 1100.1	*AP: SV.SD.Analy bit output: 1=störung vorhanden/erkannt	BOOL	false	
41	DB120.DBB 1101	*AP: SV.SD.Analy byte output: betriebsmeldung / warnung nr.	HEX	B#16#00	
42	DB120.DBB 1102	*AP: SV.SD.Analy byte output: störungsnr.	HEX	B#16#00	
43	DB120.DBB 1103	*AP: SV.SD.Analy byte output: störung info. - cmd kommando	HEX	B#16#05	
44					
45	DB21.DBW 160	*FB_SMP/DBI".sd.f funktion der sprung nr.	DEZ	24	
46	DB21.DBW 166	*FB_SMP/DBI".sd. zyklus sprung nr.	DEZ		

Abb. 62

13 Variable steuern

Die Seite bis zum Ende herunter scrollen über das Symbol



Variable steuern. Herunter scrollen ist **wichtig**, damit alle **Werte übernommen** werden.

Dazu kann es nötig werden, dass mehrmals die Steuerfunktion ausgeführt werden muss oder dass bereits gesteuerte Vorgaben/Signale/Daten gelöscht werden.

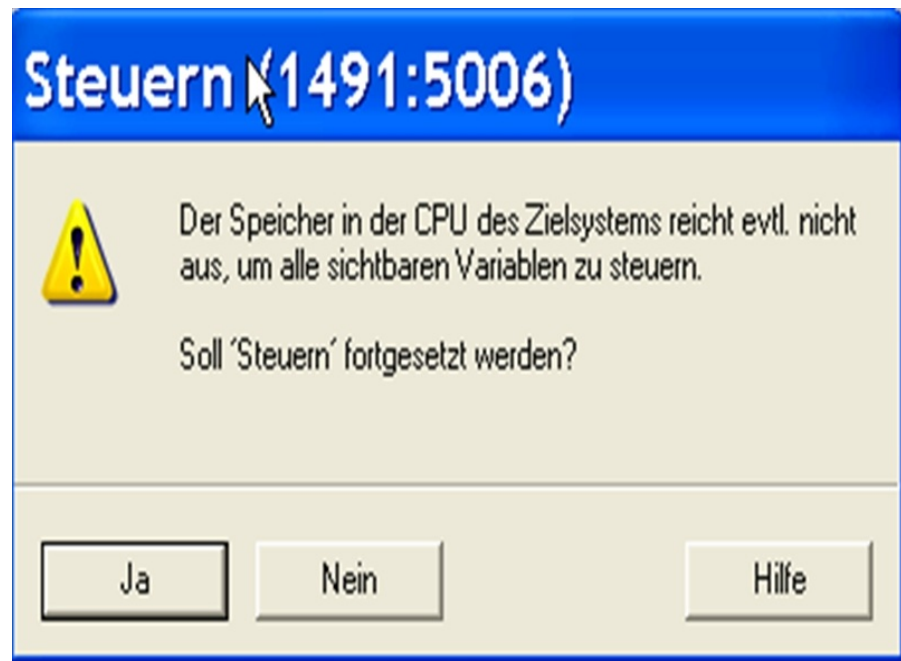


Abb. 63

- 14 Meldungen die nach dem Start Variable Steuern erscheinen, mit "Ja" bestätigen.

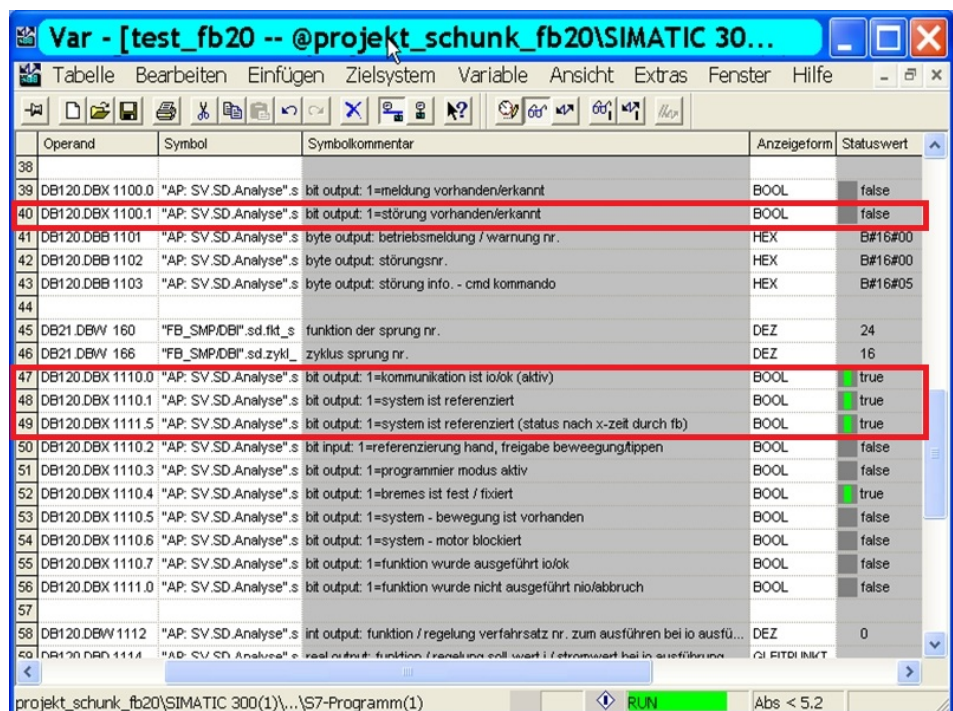


Abb. 64

15 Antriebsmodul ansteuern

Statuswert der Variablen in Zeile 37 muss größer Zykluszeit gewählt werden.

Im Schunk Test ist der Wert auf 0.25 eingestellt.

- Setzen der Variablen auf true (Strg und 1)

- Setzen der Variablen auf false (Strg und 0)

Zeile 10 (Reset): Kurz auf 1/0 setzen.

Zeile 12 (Referenzieren): Auf 1 setzen, bis das Modul referenziert ist. Siehe Zeile 48/49 sind dann beide auf True.

Antrieb kann durch **Setzen** der Variablen auf Statuswert **true** in **Zeile 15/16** nach **Links** oder **Rechts** gedreht werden. Die **Geschwindigkeit** kann durch **Verändern** des Wertes der **Variablen** in **Zeile 26** erhöht / verringert werden. Die „**Kraft**“ kann durch **verändern** des Wertes der **Variablen** in **Zeile 29** erhöht/verringert werden.

Dies sind nur einige Beispiele. Diese Variablen-tabelle dient zum Testen/Probieren der Funktionen vom Modul.

Es ist stets darauf zu achten, dass kein Schaden oder keine Verletzungen entstehen können.

4 FB20 im SPS Programm einbinden

In diesem Kapitel wird beschrieben, wie der FB20 im SPS Programm verwendet werden kann.

4.1 FB20 und die Datenbausteine

Der FB20 benötigt einen Dateninstanz-Baustein.

Dabei kann ein einzelner DB pro FB20-Aufruf zugewiesen werden, wie es im Beispielprojekt zu sehen ist.

Weil der FB20 multi-instanzfähig ist, können die Daten auch im statischen Bereich angelegt werden.

4.2 Einstellung der Schnittstelle/Kommunikation

Die Kommunikation zwischen SPS und Antrieb erfolgt über die E/A Daten. Weil am FB20 die Adressen der Daten von der Kommunikation über Pointer angegeben werden, können die Daten im E/A Bereich, Phereperiebereich, Merkerbereich oder im Datenbaustein liegen.

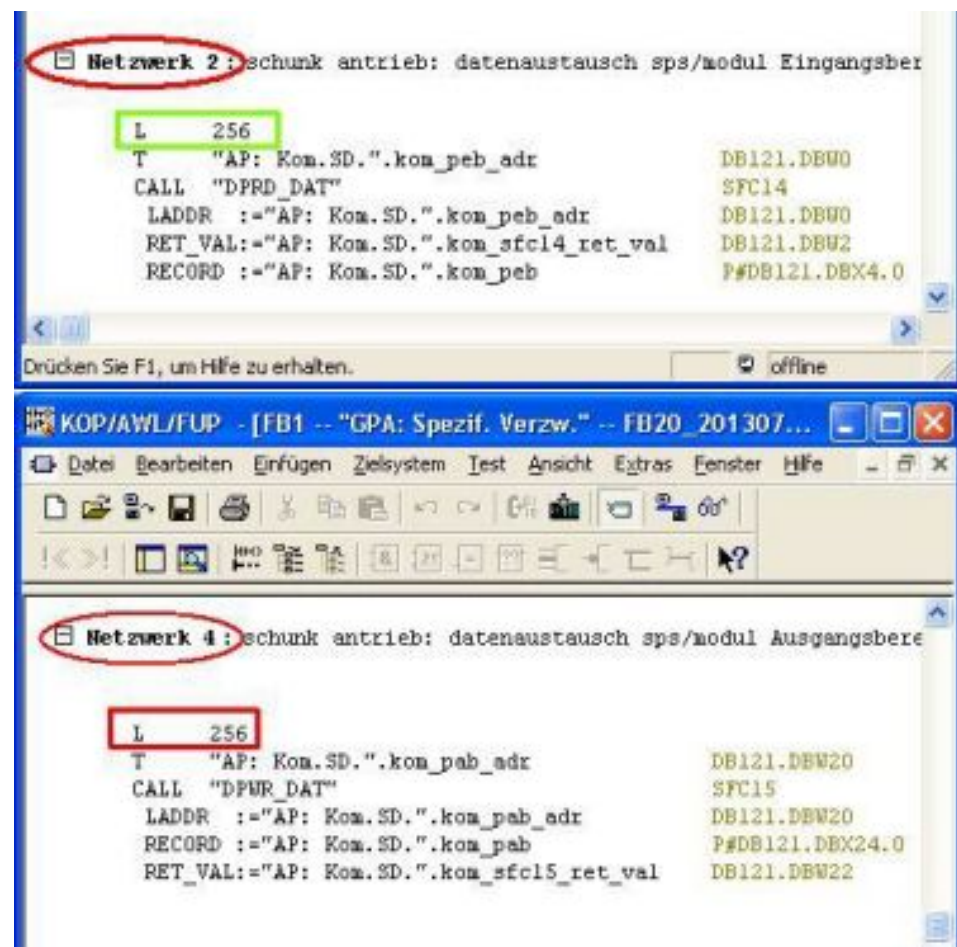


Abb. 65

Damit sichergestellt wird, dass die Datenkonsistenz erhalten bleibt, sollten die Daten über die SFC14 / SFC15 gelesen bzw. geschrieben werden. Entsprechend müssen die Adressen in Merker oder Datenbaustein zwischengespeichert werden, um dann die Adressen an den FB20 zu übergeben.

SCHUNK empfiehlt, die Anfangsbytes der Eingänge mit anzugeben, damit dies in der Querverweisliste erscheint und somit eine bessere Softwarewartung möglich ist.

4.3 Datenbaustein für die Service- bzw. Systemdaten

db_sys_sd

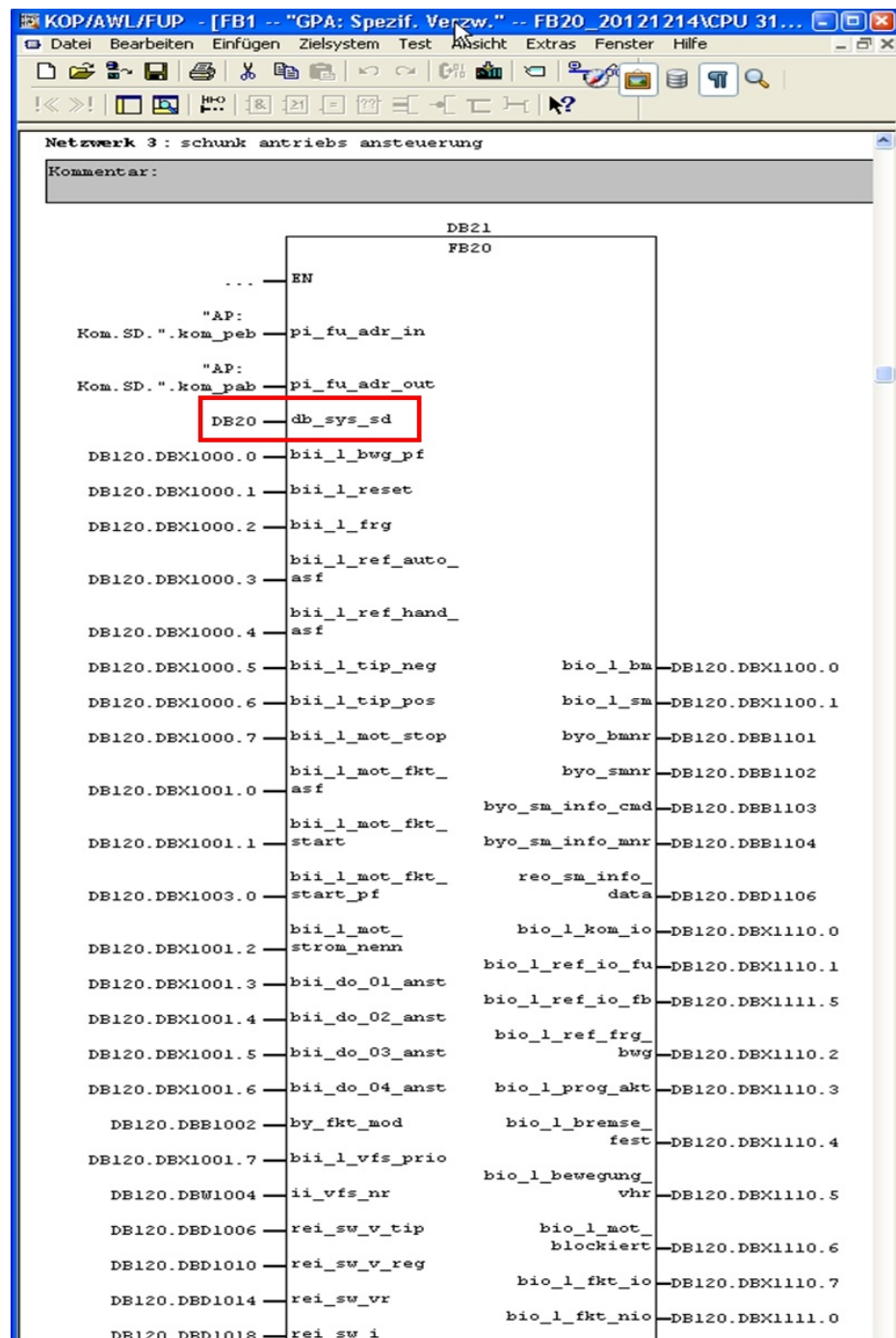


Abb. 66

Der DB20 dient als Zwischenspeicher der Kommunikationssignale und Eckdaten vom Antrieb, sowie Informationen vom FB20. Diese Signale sind für den Service wichtig, können auch vom Anwenderprogramm verwendet werden.

Damit die Daten aktualisiert werden, ist am Eingang **dii_sf_cod** der **Wert L#50331648** einzutragen. Dadurch werden in den DB die aktuellen Signale und Daten eingetragen.

Der **Nachteil** liegt darin, dass es zu Lasten der Zykluszeit geht. Diese Option sollte daher **nur genutzt werden, wenn sie unbedingt erforderlich** ist.

4.4 Zeitwert für den FB20

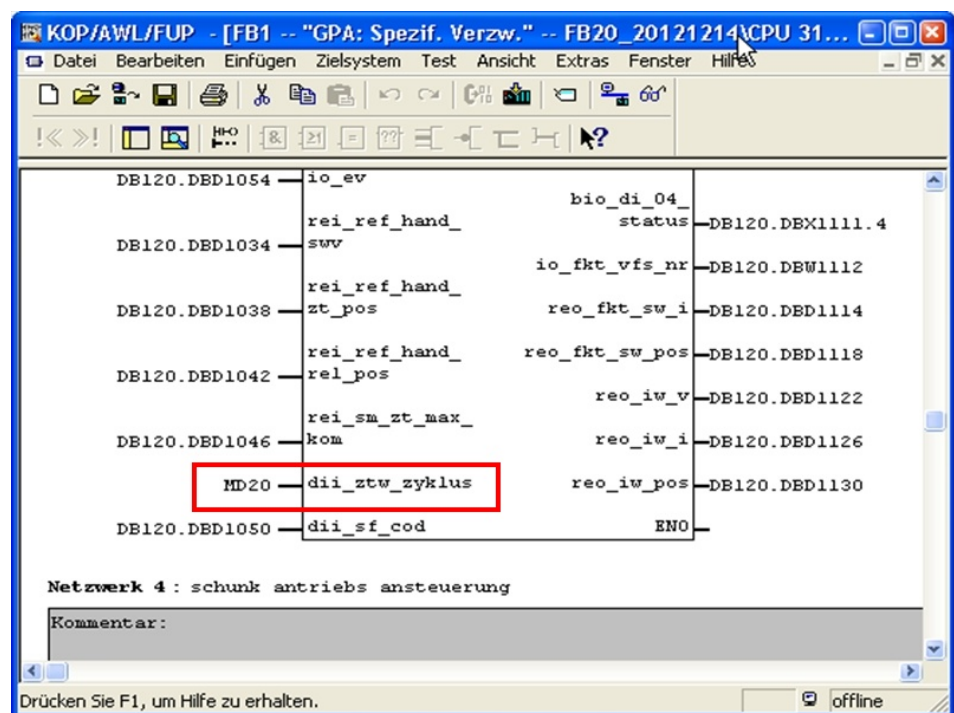


Abb. 67

- 1 Im FB werden einige Zeitfunktionen benötigt. Dazu muss am FB Eingang **dii_ztw_zyklus** die aktuelle SPS Zykluszeit angeschalten werden.
- 2 Sollte der FB durch einen Weckalarm aufgerufen werden, ist der Zeitwert vom Weckalarm einzutragen.

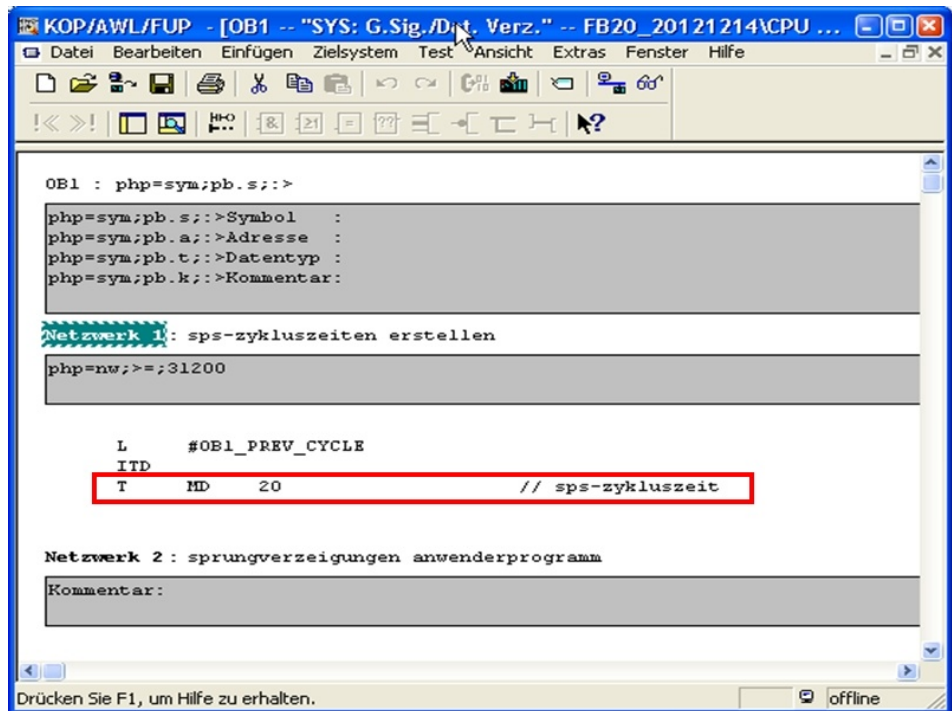


Abb. 68

- 3 Im Beispielprojekt wird das Merkerbyte **MB20** im OB1 erstellt. Wenn der Zeitwert nicht an den FB20 übergeben wird, kann das zu Problemen führen.

4.5 Warnungen / Störungen anzeigen

Für die Anzeige von Warnungen und Störungen sind mehrere Signale / Daten vorgesehen, die bei der schnellen Störungsbeseitigung helfen.

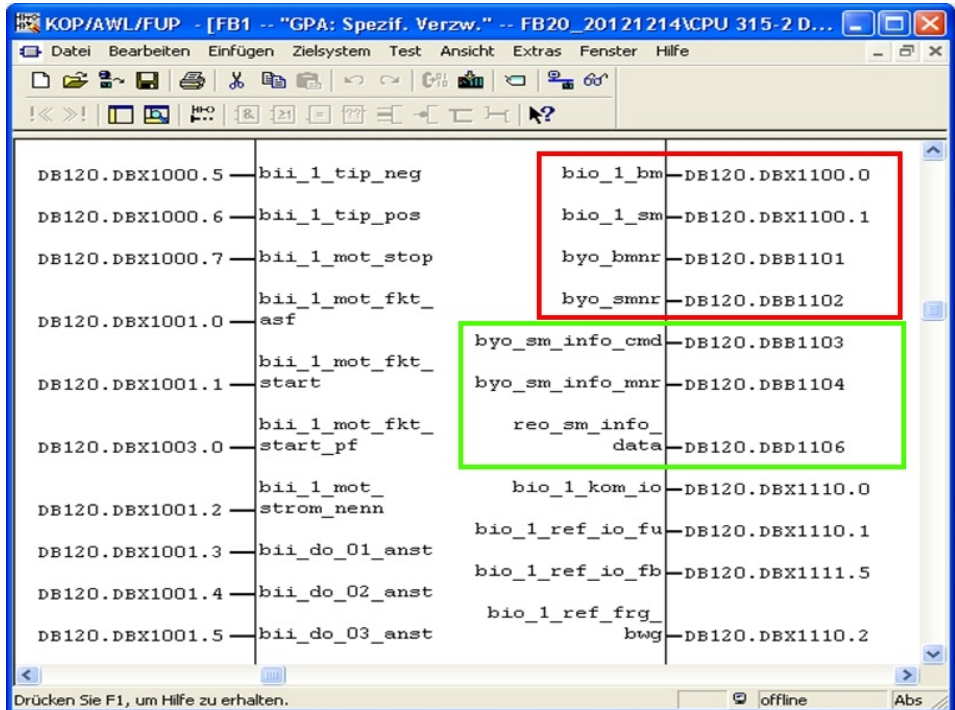


Abb. 69

- Signale**
- **bio_1_bm** = Betriebsmeldung / Warnung.
Wenn der Ausgang 1/True hat, liegt eine Warnung vor.
 - **byo_bmnr** = Kodierte Meldung der Warnung.
Das Handbuch "Motion Tool Schunk" enthält für die Bedeutung der Werte eine genauere Beschreibung.
 - **bio_1_sm** = Störung.
Wenn der Ausgang 1/True ist, liegt eine Störung vor.
 - **byo_smnr** = Kodierte Meldung der Störung.
Das Handbuch "Motion Tool Schunk" enthält für die Bedeutung der Werte eine genauere Beschreibung.

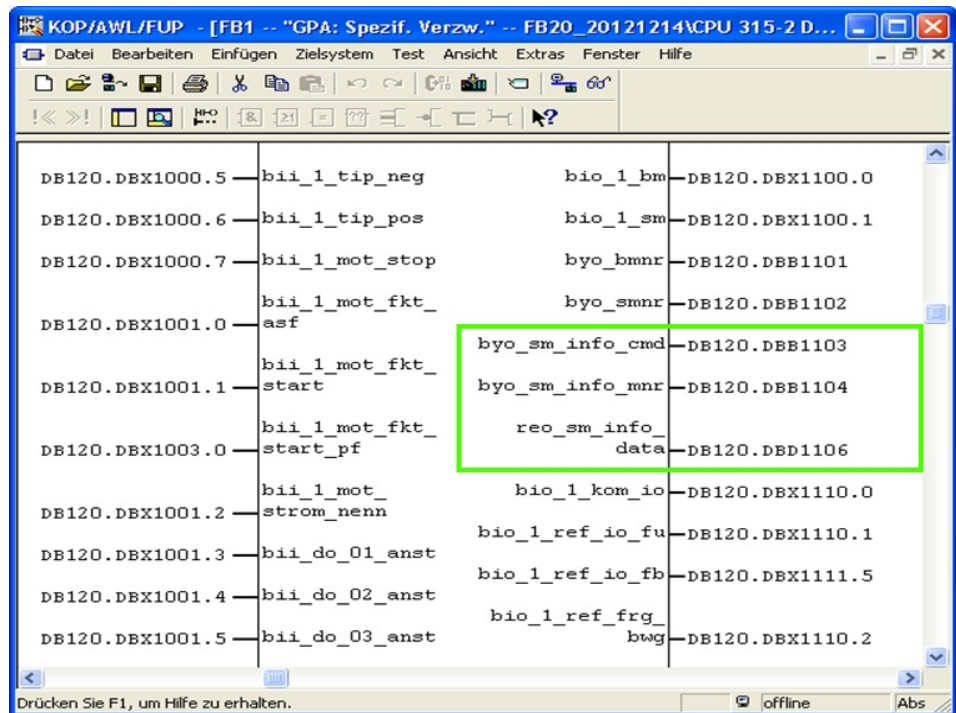


Abb. 70

Der FB sendet regelmäßig das Kommando cmd (hex.96) aus, dass weitere Diagnosedaten über die Ausgänge **byo_sm_info_cmd**, **byo_sm_info_mnr**, **reo_sm_info_data** anzeigt.

Das Handbuch "Motion Tool Schunk" enthält für das Kommando und dessen Ausgabewerte eine genauere Beschreibung.

4.6 Weitere Einstellungen der Kommunikation

Die Kommunikation zwischen SPS und Antrieb wird ständig überprüft.

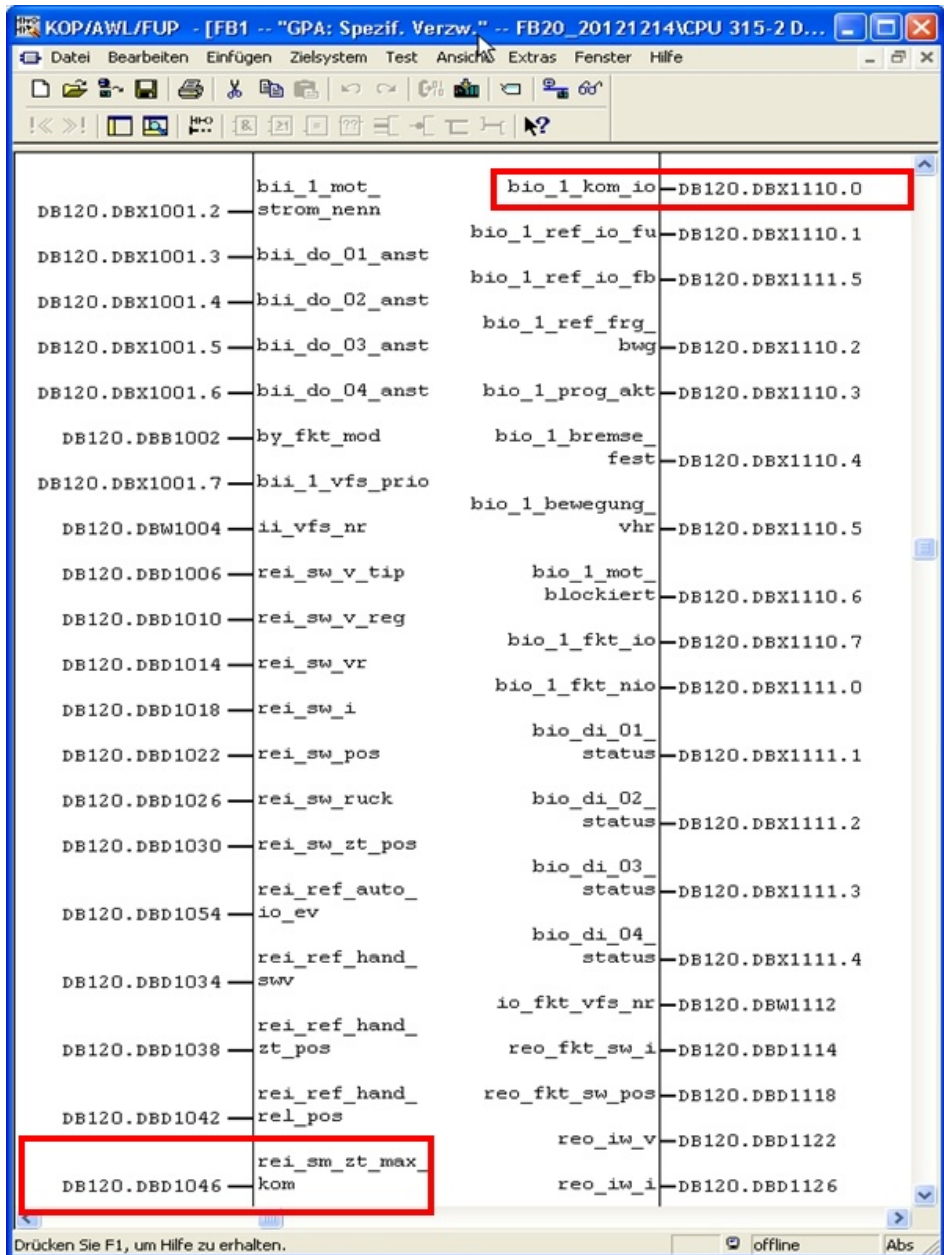


Abb. 71

- **bio_1_kom_io** = Status Kommunikation.
Die Kommunikation ist OK, wenn ein 1/True Signal vorhanden ist. Sollte dies nicht der Fall sein, muss die Hardwarekonfiguration überprüft werden.
 - Einstellungen der E / A Adressen,
 - Einstellungen am SFC14/15 und an den Adresspointern mit Zykluszeitwert.
 - Desweiteren müssen der Profibus und die Elektrik des Antriebs überprüft werden.
- **rei_sm_zt_max_kom** = Kommunikation Kontrollzeit
Daran sollte die maximal zulässige Kommunikationszeit eingetragen werden. Wenn die Zeit zu klein oder zu groß eingestellt ist, können zu viele Störungen entstehen, bzw. eine Kommunikation/Störung zu spät erkannt werden. Ein Zeitwert von 0,3 Sekunden wäre ein guter Wert.

4.7 Statussignal, Rückmeldungen

Der Antrieb gibt verschiedene Statussignale und Rückmeldungen an die Steuerung, die über den FB20 direkt ausgegeben werden.

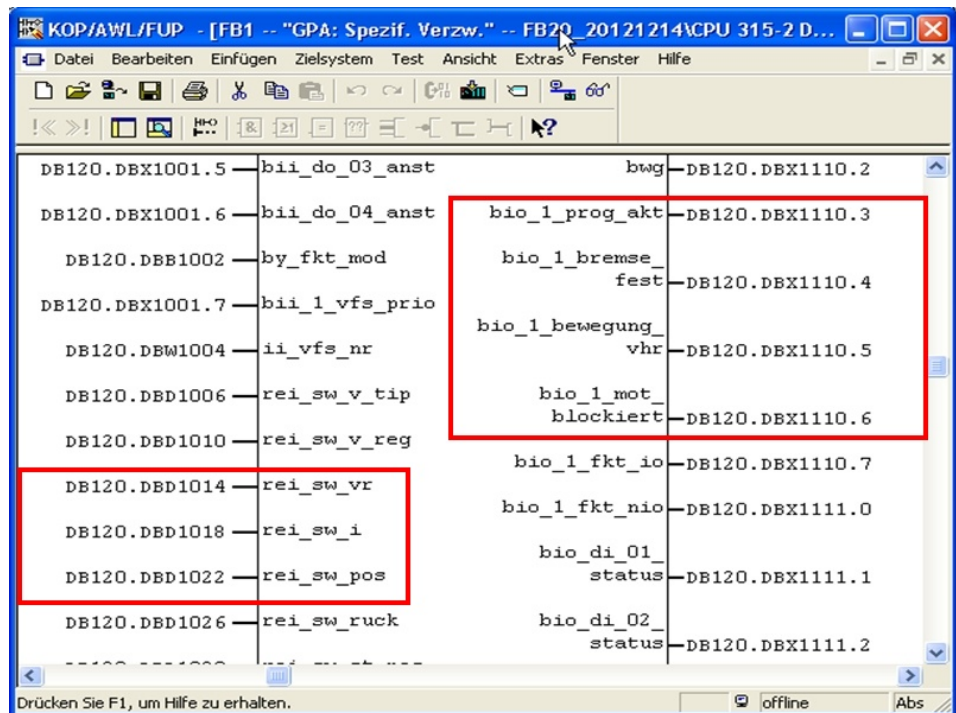


Abb. 72

- **bio_1_prog_akt = 1** Programm aktiv
- **bio_1_bremse_fest = 1** Bremse fest. Antrieb kann von Hand nicht gedreht werden, wenn Bremse vorhanden.
- **bio_1_bewegung_vhr = 1** Motor von Antrieb ist in Bewegung
- **bio_1_mot_blockiert = 1** Motor ist blockiert. Dieses Signal wird häufig beim Greifen verwendet.

4.8 Digitale Ein-/Ausgänge am Modul

Einige Antriebsmodule haben auf der Elektronikplatine Digitale E/A, welche über den FB20 gesteuert werden können.

Der FB20 bereitet die Signale vom Modul entsprechend auf, sodass ein 0/False Signal entsprechend 0Volt am Eingang bzw. Ausgang vorhanden ist, bzw. beim 1/True Signal 24Volt anliegt.

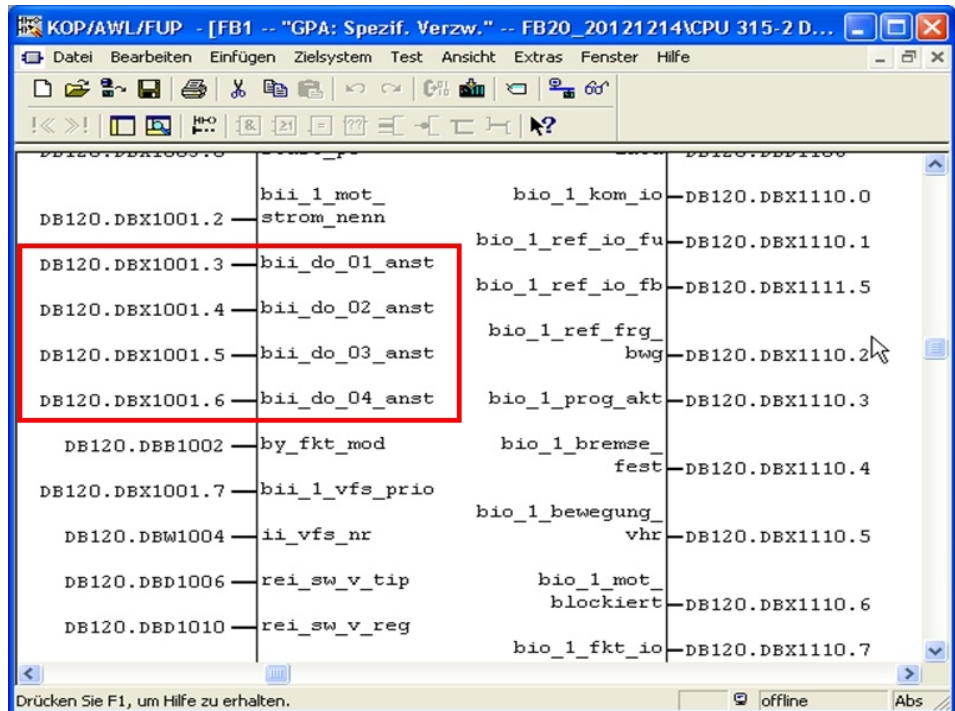


Abb. 73

Über diese 4 Eingangssignale (**bii_do_01_anst**, **bii_do_02_anst**, **bii_do_03_anst**, **bii_do_04_anst**) am FB20 können auf der Platine vorhandene Ausgänge entsprechend angesteuert werden.

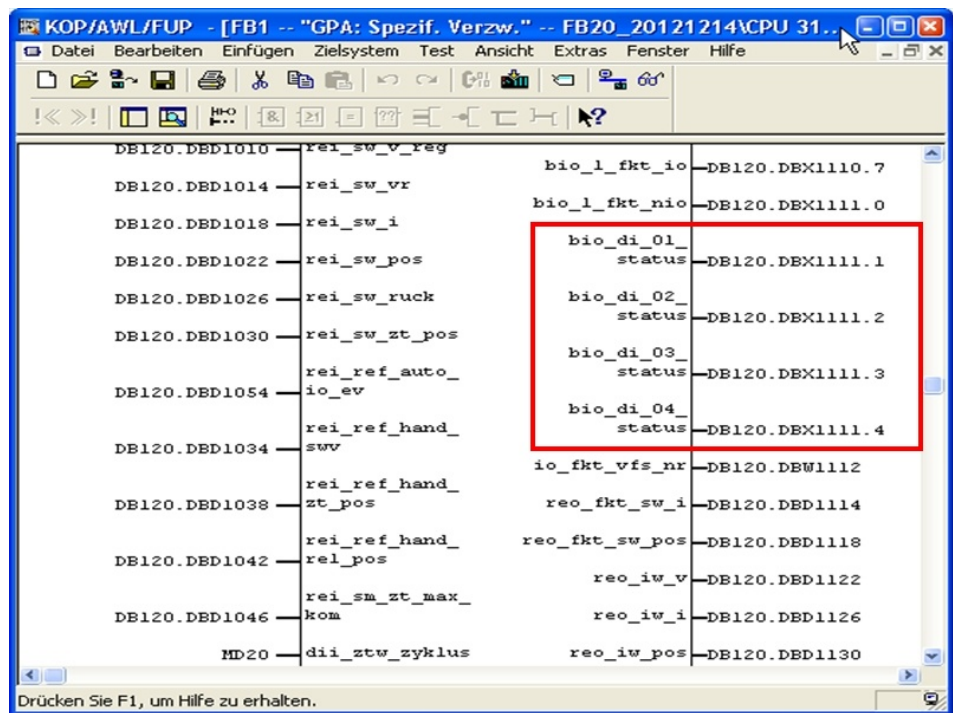


Abb. 74

Über die 4 Ausgangssignale (**bio_di_01_status**, **bio_di_02_status**, **bio_di_03_status**, **bio_di_04_status**) vom FB20 zeigen diese den Status der digitalen Eingänge der Platine an.

Die E/A's vom Modul werden regelmäßig vom FB20 nach X Kommandos abgefragt.

Weil wichtigere Funktionen häufiger zum Modul gesendet/abgefragt, bzw. angesteuert werden, kann es mehr SPS - Zyklus-Zeit dauern, bis die Eingänge/Ausgänge bearbeitet werden. Daher sind diese Signale nur für Zeiten unkritischer Anwendungen geeignet.

Daher sind diese Signale nur für Zeiten unkritischer Anwendungen geeignet.

4.9 Allgemeine Signale und Daten

Einige Signale sind für mehrere Funktionen zuständig, und für den Betrieb des Antriebs wichtig.

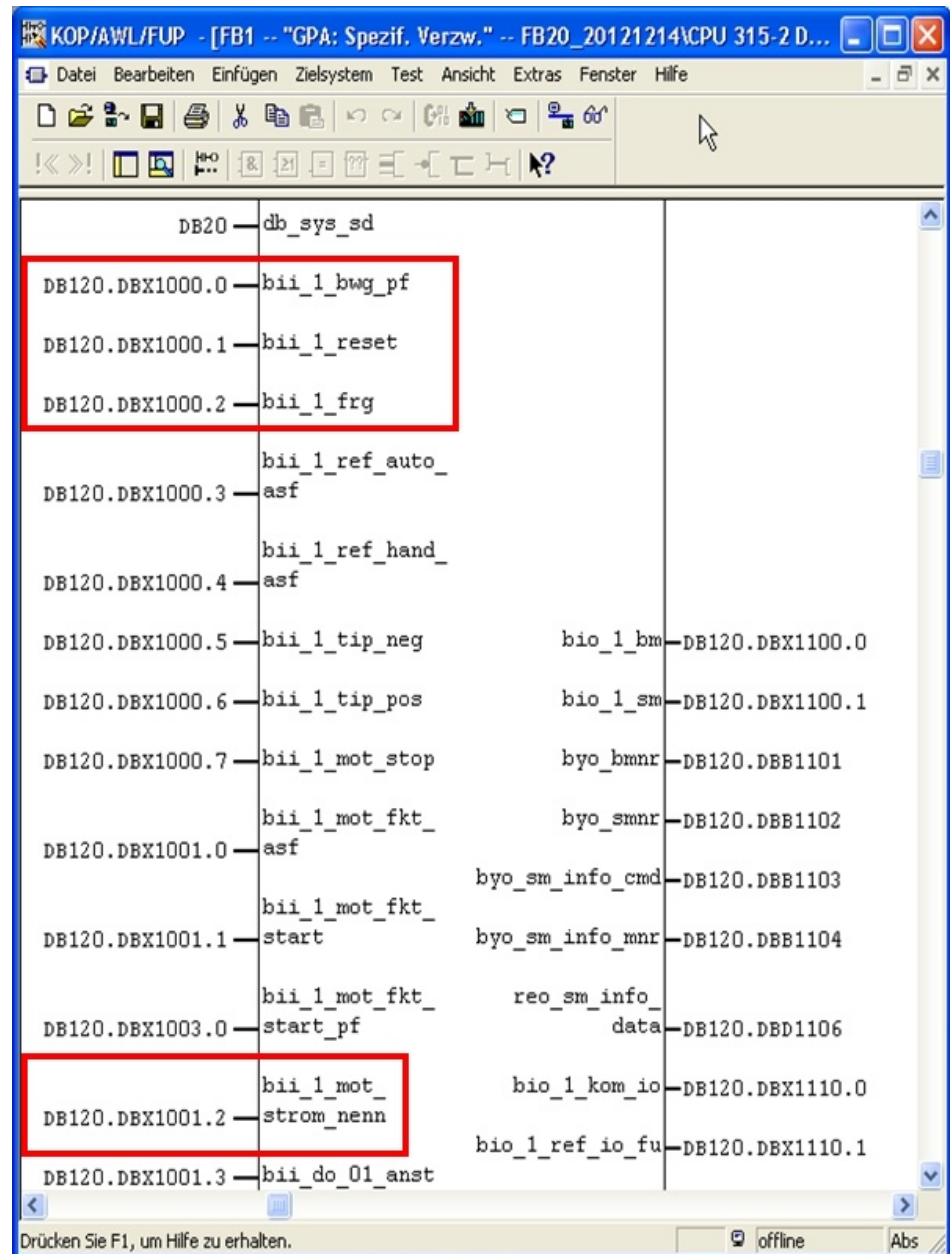


Abb. 75

- **bii_1_frg** = Freigabe
Bei 0/False Signal wird ein Notstopp ausgeführt, was zu einer Störung führt und quittiert werden muss.
- **bii_1_reset** = Reset
Beim Wechsel von 0/False auf 1/True wird die Störung quittiert / zurückgesetzt, wobei der Antrieb gestoppt wird.
Störungen können nur quittiert werden, wenn diese zuvor behoben worden sind.
- **bii_1_bwg_pf** = Bewegung über P- Flanke
Bei einem 1/True Signal wird nur eine Bewegung ausgeführt, bei einer positiven Flanke (Signalwechsel von 0 auf 1), an den entsprechenden Bewegungsfunktionen (z.B. Tippen). Dadurch soll gewährleistet werden, dass der Antrieb nach dem er angehalten wurde, erst wieder angesteuert werden kann, sobald eine positive Flanke (Signalwechsel von 0 auf 1) erkannt wird. So kann ein selbstständiger Anlauf unterbunden werden (z.B. beim Tippen).
- **bii_mot_strom_nenn** = Motornennstrom
Der Motorstrom kann über den Wert **rei_sw_i** begrenzt werden. Bei einem 1/True Signal am Eingang wird der Motor mit seinem Nennstromwert ohne Rücksicht auf Sollwert Vorgabe, gesetzt.

4.10 Referenzieren Auto

Bevor eine Positionierung ausgeführt werden kann, muss die Achse referenziert sein.

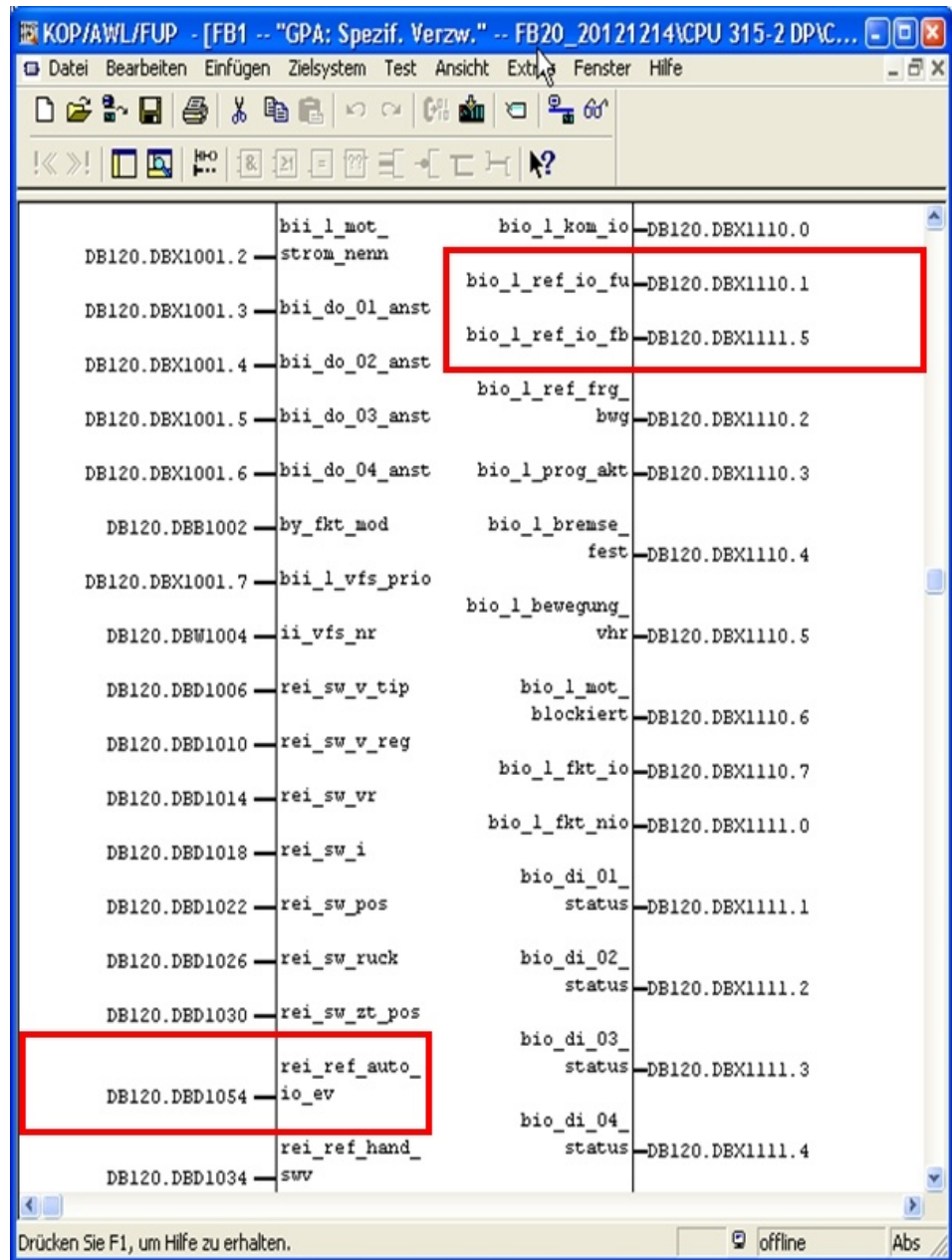


Abb. 76

- **bii_1_ref_auto_asf** = Referenzieren Auto
Wenn ein 1/True Signal anliegt wird die Referenzierung gestartet, auch wenn die Achse referenziert ist. Erscheint während der Referenzierung das Signal 0/False, geht die Achse in stopp und die Referenzierung muss erneut durchgeführt werden.
- **bio_1_ref_io_fu** = Status Referenzierung Antrieb
Wenn die Achse in der Referenzierung ausgeführt wird, erscheint als Ausgabe 1/True.
- **bio_1_ref_io_fb** = Status Referenzierung FB
Nachdem der Antrieb referenziert ist (**bio_1_ref_io_fb=1**), kann nach interner Einstellung des Antriebs noch eine Bewegung erfolgen (z.B. um **X Millimeter** zum Referenzpunkt zufahren). Erst wenn der Antrieb nach **X Zeit** (Einstellung über **rei_ref_auto_io_ev**) für **X Zeit** still steht, wird ein 1/True Signal ausgegeben.

4.11 Referenzierung Hand

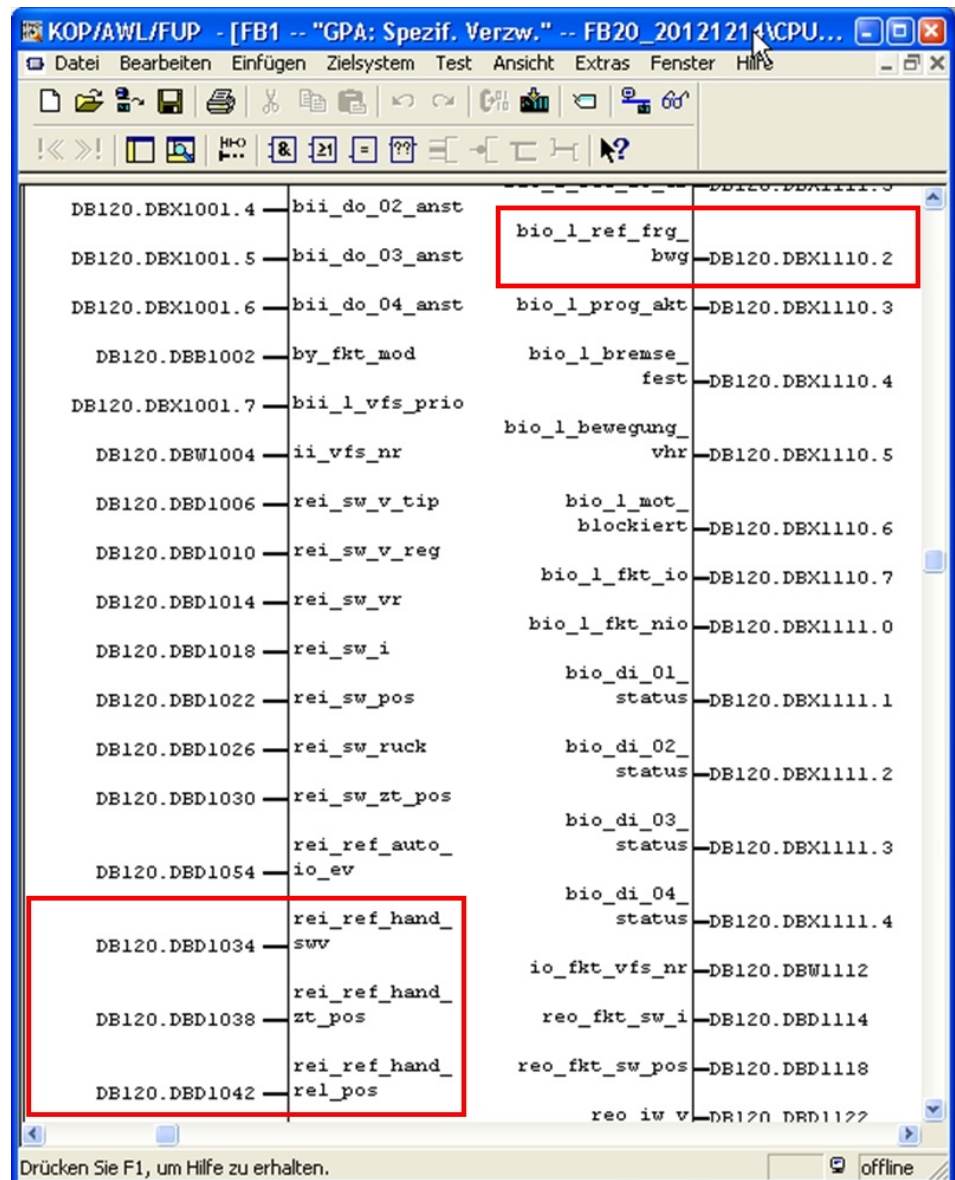


Abb. 77

Diese Funktion ist in der Software deaktiviert und hat keine Funktion.

4.12 Antrieb tippen

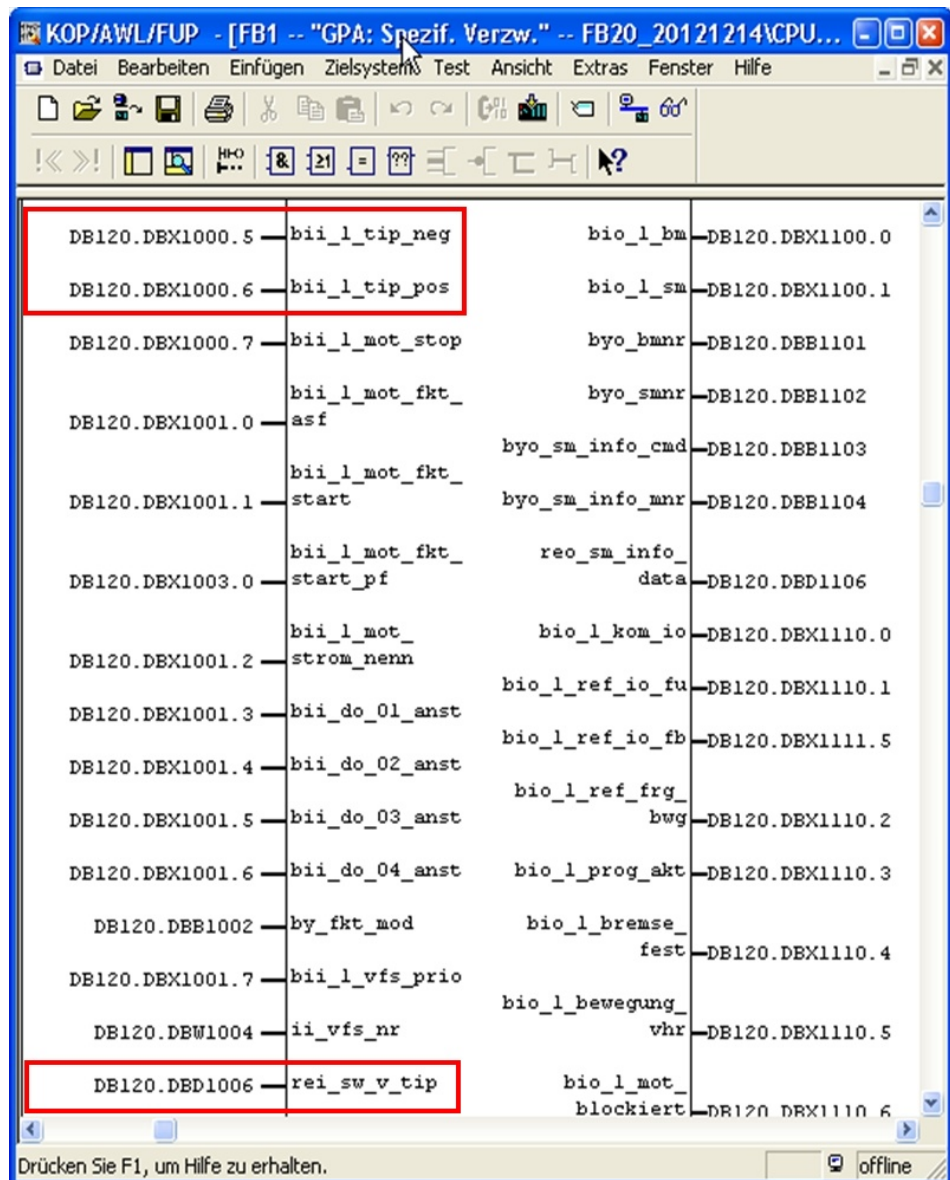


Abb. 78

- **bii_1_tip_neg** und **bii_1_tip_pos**

Es gibt die Möglichkeit, den Antrieb zu tippen, auch wenn der Antrieb nicht referenziert ist. Sobald der Antrieb entsprechend freigegeben ist kann der Antrieb entsprechend verfahren werden.

- **rei_sw_v_tip**

Geschwindigkeitsvorgabe für die Tippfunktion.

4.13 Allgemeine Vorgaben

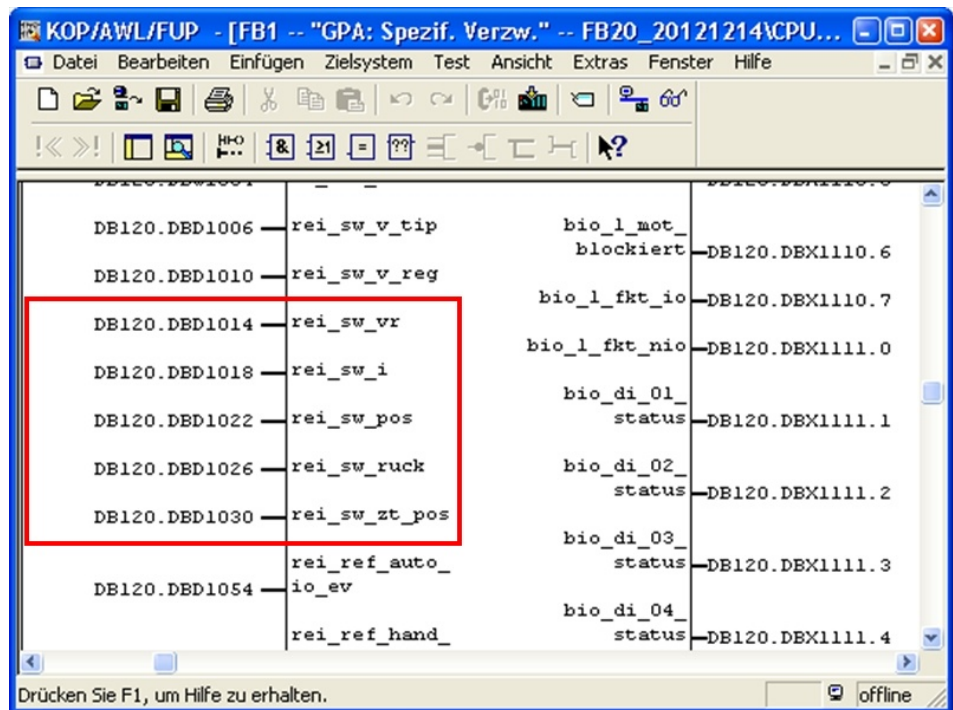


Abb. 79

- **rei_sw_vr**
Einstellung der Rampe für Beschleunigung / Verzögerung.
- **rei_sw_i**
Vorgabe des maximalen Stromes. Ein Stromwert kleiner 0,001 sollte nicht vorgenommen werden.
- **rei_sw_ruck**
Sollwertvorgabe für den "Ruck". Näheres siehe Handbuch.
- **rei_sw_zt_pos**
Zeitvorgabe für das Positionieren, wenn entsprechend die Regelungsart verwendet wird. Der Antrieb errechnet seine Geschwindigkeit selbst, um die Position in der vorgegebenen Zeit zu erreichen.

Die oben beschriebenen Vorgaben werden regelmäßig in den Antrieb übertragen. Näheres siehe Handbuch.

4.14 Regelungsfunktionen

Bevor eine Regelungsfunktion ausgeführt wird, werden eventuell geänderte Werte vor dem Start der Regelungsfunktionen in den Antrieb übertragen.

Für den Start der Regelungsfunktionen gibt es drei Möglichkeiten. Voraussetzung ist, dass der Antrieb betriebsbereit (referenziert und keine Störung vorhanden) ist.

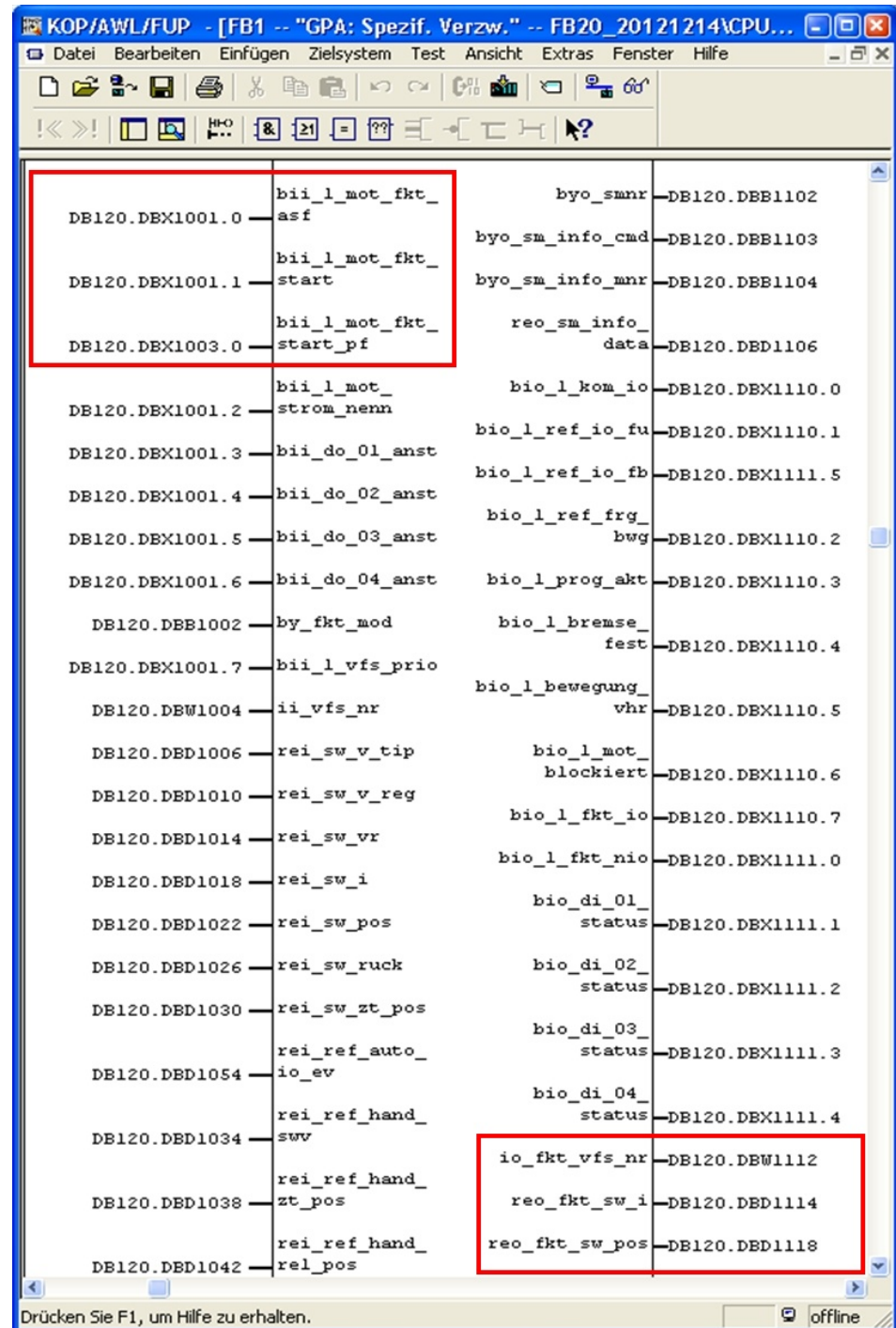


Abb. 80

- **bii_1_mot_fkt_asf**

Wenn hier eine 1/True anliegt wird die Regelungsfunktion mit den aktuellen Einstellungen ausgeführt. Sollte sich ein Wert ändern, wie z.B. Regelungsfunktionen oder eine Positionsvorgabe wird die Regelungsfunktion automatisch gestartet. Bei einem Signalwechsel von 1/True auf 0/False wird der Antrieb gestoppt.

- **bii_1_mot_fkt_start**

Mit der positiven Flanke (Signalwechsel von 0/False auf 1/True) wird die Regelung gestartet. Bei einem Signalwechsel von 1/True auf 0/False wird der Antrieb gestoppt.

- **bii_1_mot_fkt_start_pf**

Dieses Signal bildet die gleiche Funktion ab wie vom FB10 das Eingangssignal **I_CMD_START_POS**. Das bedeutet, dass die Regelungsfunktion mit dem Signalwechsel von 0/False auf 1/True gestartet wird und erst wieder gestoppt wird, wenn die Funktion ausgeführt / abgeschlossen ist, oder durch z.B. Störung, Stopp Signal, Signalfreigabeentzug gestoppt wird.

- **io_fkt_vfs_nr, reo_fkt_sw_i, reo_fkt_sw_pos**

Wenn die Regelungsfunktionen abgeschlossen werden konnten, werden an den Ausgängen die Werte ausgegeben, die beim Start der Regelungsfunktion anstanden.

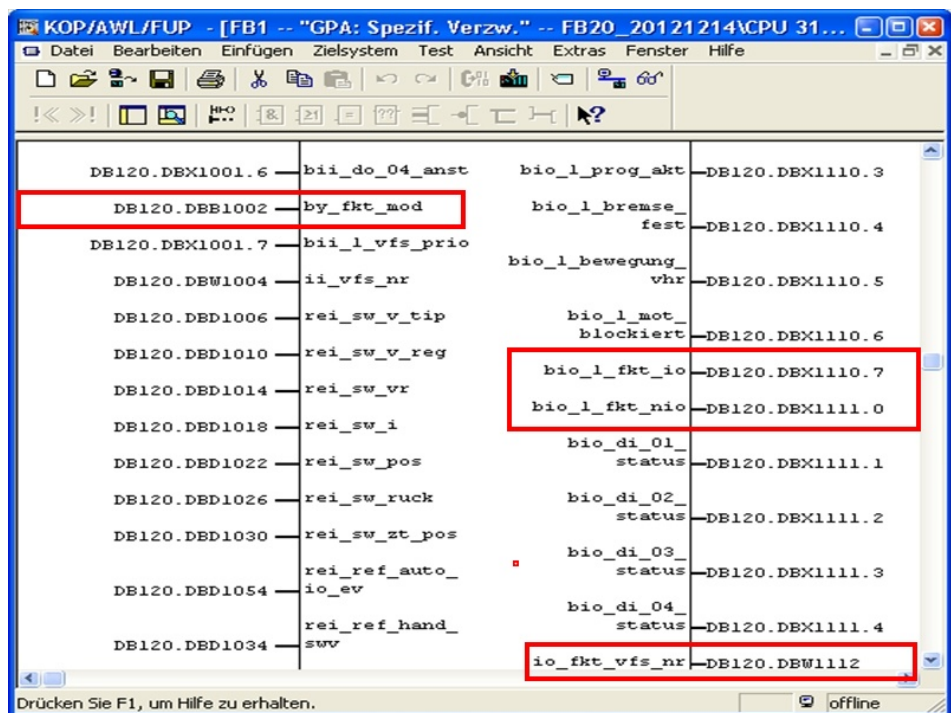


Abb. 81

- **bio_1_fkt_io** und **bio_1_fkt_nio**
Diese beiden Signale geben den Status vor der Regelungsfunktion aus.
- **bio_1_fkt_io**
Es liegt eine 1/True an, wenn die Regelungsfunktion erfolgreich ausgeführt wurde.
- **bio_1_fkt_nio**
Es wird eine 1/True ausgegeben, wenn die Funktion nicht ausgeführt wurde, bzw. abgebrochen wurde.
- **by_fkt_mod** = Regelungsart
Vorgabe der entsprechenden cmd Befehlsnummer, zum Ausführen der Regelungsart. Folgende Werte sind gültig:
 - CMD (Hex.b0): move pos
 - CMD (Hex.b1): move pos time
 - CMD (Hex.b3): move cure
 - CMD (Hex.b5): move vel
 - CMD (Hex.b7): move grip
 - CMD (Hex.b8): move pos rel
 - CMD (Hex.b9): move pos time rel
 - CMD (Hex.c1): exe phrase

Weitere Informationen zu den CMD Kommandos enthält das Handbuch "Motion Tool Schunk".

Eine besondere Funktion wird mit dem Wert **by_fkt_mod = #16#ff** gestartet. Diese Sonderfunktion ermöglicht es, den Antrieb über die Geschwindigkeitsvorgabe und den Stromwert zu steuern. Dabei werden die Vorgaben von **rei_sw_v_reg** und **rei_sw_i** immer wieder aktualisiert.

4.15 Verfahrssatztabelle

Mit dem MTS Tool kann eine Verfahrssatztabelle im Antrieb hinterlegt werden. In diese Verfahrssatztabelle können Funktionen eingetragen werden, die mit der Ausführung im Widerspruch zu den Funktionalitäten des FB20 stehen. Daher gibt es die Möglichkeit den Antrieb nur über die Verfahrssatztabelle zu fahren.

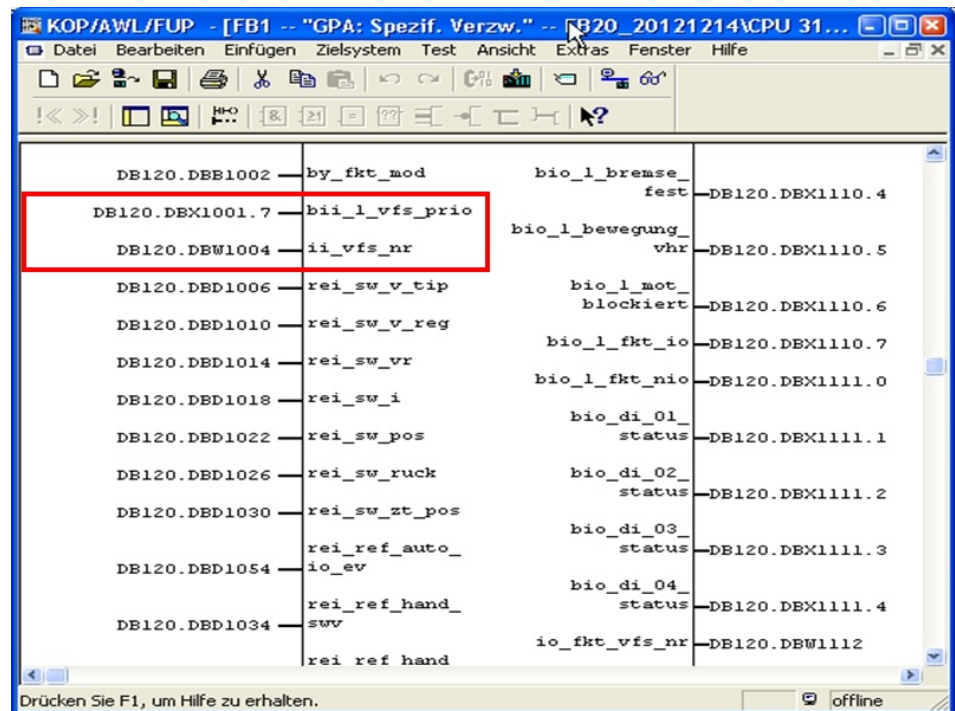


Abb. 82

- **bii_1_vfs_prio** = Verfahrssatztabelle Priorität
Wenn an diesem Eingang 1/True Signal anliegt, kann der Antrieb nur noch über die Verfahrssatztabelle gesteuert werden. NotAus und Stop bleiben weiterhin aktiv. Die Störungsquittierung ist ebenfalls deaktiviert.
- **ii_vfs_nr** = Verfahrssatztabellennummer
Hier wird die Verfahrssatztabellennummer angegeben, die ausgeführt werden soll.

5 Unterschiede zwischen FB10 / FB20

In diesem Kapitel wird die Beschaltung vom FB10, Vorgänger Baustein zur Ansteuerung der Antriebsmodule und dem aktuellen FB20 gegenüber gestellt. So entsteht ein Überblick der Gemeinsamkeiten der FB's, sowie eine Darstellung der Neuerungen am FB20. Mit diesen Informationen kann schnell vom FB10 auf den FB20 gewechselt werden.

5.1 FB Beschaltung: Input + In-/Output

FB10/20 Beschaltung: Input + In-/Output		
FB10	FB20	Bemerkung
I_ADDR	pi_fu_adr_in pi_fu_adr_out	Die Vorgabe der Adresse des Moduls wird nun über Pointer gemacht. Dies hat folgende Vorteile: - Die E/A-Adressen können unterschiedlich sein. - Eintrag in der Querverweisliste. - Flexibler, weil der komplette Adress-Bereich der CPU genutzt werden kann.
---	db_sys_sd	Dieser DB enthält weitere Signale/Daten, die für den Anwender und Service genutzt werden kann.
I_FL_CYCLE_BYTE	---	Die Blinker werden nicht mehr benötigt.
---	bii_1_bwg_pf	Mit diesem Signal wird fest gelegt, ob ein Start der Bewegung nur durch ein positives Flankensignal erfolgen kann, wenn der Antrieb gestoppt wurde (z.B. durch eine Störung oder Stop Funktion). Dieses Signal hat einen Einfluss auf folgende Signale: - bii_1_tip_neg - bii_1_tip_pos - bii_1_mot_fkt_asf - bii_1_vfs_prio
I_ACK	bii_1_reset	Quittierung einer Störung, bzw. zurücksetzen von internen FB Signale/Daten.

FB10/20 Beschaltung: Input + In-/Output		
FB10	FB20	Bemerkung
I_RESEND_CMD	---	Wird nicht mehr verwendet, um ein selbständiges/unerwartetes Anlaufen zu vermeiden.
I_CONT_POS_FEEDB	---	Die Positionsangabe, sowie weitere Signale/Daten werden in Intervallen aktualisiert.
I_ENABLE	bii_1_frg	Freigabe der Bewegung. Bei 0 Signal wird ein Not-Stop ausgeführt, bis dieser bestätigt wurde oder der Antrieb steht.
I_CMD_REF_MODUL	bii_1_ref_auto_asf bii_1_ref_hand_asf	Erweiterung der Referenzierungsarten wurde vorgesehen. Eine automatische, die der Funktion (I_CMD_REF_MODUL) entspricht, sowie eine Handreferenzierung als Reservesignal.
I_CMD_JOG_PLUS	bii_1_tip_neg	Funktionalitäten wurden verbessert. So ist ein direktes Umschalten, sowie die Änderung der Geschwindigkeitsvorgabe während der Ausführung möglich.
I_CMD_JOG_MINUS	bii_1_tip_pos	Funktionalitäten wurden verbessert. So ist ein direktes Umschalten, sowie die Änderung der Geschwindigkeitsvorgabe während der Ausführung möglich.
I_CMD_STOP_MOTION	bii_1_mot_stop	Funktionalität wurde verbessert. Wenn einmal das Stoppsignal am FB anlag (auch wenn es nur für einen SPS-Zyklus vorhanden war), wird die Stop Funktion ausgeführt, bis keine Bewegung mehr vorhanden ist und die Bremse den Motor fest hält.
I_CMD_START_POS	bii_1_mot_fkt_asf bii_1_mot_fkt_start	Start der Motor Funktion/Regelung. Es wurden zwei Arten der Funktionsstarts der CMD Kommandos eingebunden, um so den unterschiedlichen Bedürfnissen in der SPS- Programmierung gerecht zu werden. Beim Signal 'bii_1_mot_fkt_start' wird die Funktion bei einer positiven Flanke gestartet, während beim Signal 'bii_1_mot_fkt_asf' immer ein Start der Regelungsfunktion ausgeführt wird,

FB10/20 Beschaltung: Input + In-/Output		
FB10	FB20	Bemerkung
		sobald neue Werte anliegen. Dies ist nützlich beim Automatik-, bzw. Schrittkettenbetrieb, weil nur die neuen Werte geändert werden müssen ohne ein Signalwechsel über mehrere SPS-Zyklen zu erstellen. Dabei wird die Regelungsfunktion nur so lange ausgeführt, wie am Signal 'bii_1_mot_fkt_start' bzw. 'bii_1_mot_fkt_asf' 1/True ist.
	bii_1_mot_fkt_start_pf	Dieses Signal hat die gleichen Funktionalitäten wie das 'bii_1_mot_fkt_start' Signal, nur mit dem Unterschied, dass es nicht anliegen bleiben muss. Damit soll die gleiche/ähnliche Funktionalität nachgebildet werden, wie das vom alten FB 'I_CMD_START_POS'
	bii_1_mot_strom_nenn	Diese Signal wurden neu eingefügt, damit ein einfaches Umschalten auf den Nennstrom des Moduls erfolgen kann, welches z.B. bei der Tippfunktion nützlich ist.
---	bii_do_01_anst	Die Ausgänge des Antriebs werden entsprechend des Signalzustands der im FB anliegt, intervallmäßig angesteuert.
	bii_do_02_anst	
	bii_do_03_anst	
	bii_do_04_anst	
I_POS_MODE	by_fkt_mod	Angabe der Regelungsfunktion von xB0..C1. Als neue Sonderfunktion ist das Kommando xff hinzugekommen, um den Antrieb nur nach dem Soll-Wert V/I anzusteuern.
---	bii_1_vfs_prio	Priorität hat die Funktion der Verfahrtsatzausführung. Nur Stop und Not-Halt sind noch gültig.
I_POS_SEQUENCE	ii_vfs_nr	Vorgabe Verfahrtsatz Nr.. Verfahrtsätze mit ACK,Referenzierung oder ähnlichen Funktionen können nicht ausgeführt werden, wenn diese

FB10/20 Beschaltung: Input + In-/Output		
FB10	FB20	Bemerkung
		im Widerspruch der Funktionalitäten des FB20 stehen.
I_POS_VELOCITY	rei_sw_v_tip	Für die Betriebsarten Hand- bzw. Automatik in den Anlagen, wurden in dem neuen FB nun zwei Geschwindigkeitsvorgaben gemacht werden, die eine Vereinfachung für den SPS- Programmierer darstellen.
	rei_sw_v_reg	
I_POS_ACCELERATION	rei_sw_vr	Beschleunigungswert
I_POS_CURRENT	rei_sw_i	Stromvorgabe
I_POS_POSITION	rei_sw_pos	Positionsvorgabe
I_POS_JERK	rei_sw_ruck	Beschleunigungsruck
I_POS_TIME	rei_sw_zt_pos	Zeit für die Positionierung (zu Befehl "MovPosTime")
---	rei_ref_hand_svv	Referenzierung: Geschwindigkeits- Soll-Wert (Reserve)
---	rei_ref_hand_zt_pos	Referenzierung: Zeit -Positionierung (Reserve)
---	rei_ref_hand_rel_pos	Referenzierung: Relative Position (Reserve)
I_RES_TIME	---	Entfällt auf Grund der Struktur des FB's
I_TIMER_WDOG	rei_sm_zt_max_kom	Zeit max. für Kommunikation Überwachung.
---	dit_ztw_zyklus	SPS-Zykluszeit für Zeitfunktionen
---	dii_sf_cod	Service Signale für Analysefunktionalität

5.2 FB Beschaltung: Output

FB10/20 Beschaltung: Output		
FB10	FB20	Bemerkung
M_WARNING	bio_1_bm	Status Signal Warnung vom Antriebsmodul wird ausgegeben.
M_ALARM	bio_1_sm	Status Signal Störung vom Antriebsmodul wird ausgegeben.
---	byo_bmnr	Warnung Info die über den Rückgabewert CMD x89 angegeben wurde. Wenn das Statussignal Warnung nicht anliegt, wird nach einiger Zeit (max. Kommunikation) der Wert zurückgesetzt.
M_ERROR	byo_smnr	Störung Info die über den Rückgabewert CMD x88 angegeben wurde. Wenn das Statussignal Störung nicht anliegt, wird nach einiger Zeit (max. Kommunikation) der Wert zurückgesetzt.
---	byo_sm_info_cmd	Diese Meldung, Detail Information wird intervallmäßig ausgelesen, über den CMD x96 Commando und die Daten über die drei Outputwerte ausgegeben.
---	byo_sm_info_mnr	
---	reo_sm_info_data	
---	bio_1_kom_io	Statussignal vom FB dass die Kommunikation mit dem Antrieb in Funktion ist.
M_REFERENCED	bio_1_ref_io	Statussignal vom Antriebsmodul, dass es referenziert ist.
---	bio_1_ref_io_fb	Verzögertes Statussignal vom FB, dass die Referenzierung abgeschlossen ist, nach dem keine Bewegungen mehr vorhanden sind.
	bio_1_ref_frg_bwg	Statussignal für die Handreferenzierung (Reserve)
---	bio_1_ref_frg_bwg	Bei Handreferenzierung wird das Signal auf

FB10/20 Beschaltung: Output		
FB10	FB20	Bemerkung
		1 gesetzt, für die Anzeige, dass der Antrieb nun verfahren werden kann.
M_PROG_ACTIVE	bio_1_prog_akt	Statussignal vom Antriebsmodul.
M_BRAKE_ACTIVE	bio_1_bremse_fest	Statussignal vom Antriebsmodul.
M_IN_MOTION	bio_1_bewegung_vhr	Statussignal vom Antriebsmodul.
M_MOTION_BLOCKED	bio_1_mot_blockiert	Statussignal vom Antriebsmodul.
M_POS_REACHED	bio_1_fkt_io	Bei Priorität Verfahrssatz wird das Signal durchgereicht, welches über 'bio_1_fkt_io' ausgegeben wird.
	bio_1_fkt_nio	Bei normaler Funktionsausführung, CMD B0..C1 wird zwischen Ausführung IO / NIO unterschieden. NIO wird dann gesetzt, wenn während der Ausführung die Funktion abgebrochen wird.
---	bio_di_01_status	Statussignal des digitalen Eingangs vom Antriebsmodul.
	bio_di_02_status	
	bio_di_03_status	
	bio_di_04_status	
--	io_fkt_vfs_nr	Wenn die Regelungsfunktion erfolgreich abgeschlossen wurde, werden die Werte mit ausgegeben, welche beim Start der Regelungsstartfunktion am FB anlagen. Damit kann der Programmierer zusätzlich im Programm die Werte auswerten, z.B. für die Weiterschaltbedingung der Schrittketten.
	reo_fkt_sw_i	
	reo_fkt_sw_pos	
M_ACT_VELO	reo_iw_v	Aktuelle Geschwindigkeit.
M_ACT_CURR	reo_iw_i	Aktueller Strom
M_ACT_POS	reo_iw_pos	Aktuelle Position

6 FB CMD Kommandos

Verwendete CMD Kommandos im FB	
CMD	Bemerkung
x80 Konfiguration auslesen	Einige Grunddaten werden ausgelesen, um diese im FB zu verwenden, die die Begrenzung von den Vorgabewerten darstellen.
x8B Störung quittieren	
x90 Not-Stop	
x91 Stop	
x92 Referenzierung	
x95 get state	
x96 get detailed error info	
x97 Referenzierung Hand	Vorbereitet und vorerst deaktiviert, bis die Probleme auf der Antriebsseite behoben wurde.
xA0 Geschwindigkeitswert setzen	Wird intervallmäßig bearbeitet.
xA1 Beschleunigungswert setzen	Wird intervallmäßig bearbeitet.
xA2 Ruckwert setzen	Wird intervallmäßig bearbeitet.
xA3 Strom wert setzen	Wird intervallmäßig bearbeitet.
xA4 Zeitwert setzen	Wird intervallmäßig bearbeitet.

Verwendete CMD Kommandos im FB	
CMD	Bemerkung
xB0 move pos	
xB1 move pos time	
xB3 move cure	
xB5 Geschwindigkeits- fahrt	
xB7 move grip	
xB8 move pos rel	
xB9 Move pos time rel	
xC1 exe phrase	Nicht alle Funktionalitäten können von dieser Verfahrstabelle genutzt werden (z.B. Referenzierung). Alternativ ist die Priorität der Verfahrstabelle (bii_1_vfs_prio) zu aktivieren.
xE1 DI/DO le- sen/schreiben	Das Lesen/Schreiben wird zyklisch bearbeitet, neben vielen weiteren Kommandos. Dadurch ist die Aktualisierungszeit langsam.

7 Versionshinweise

Version	Datum	Bemerkung
		Empfehlung Die Kommunikation oder der Datenaustausch sollte über die SFC14/15 erfolgen. Alternativ kann die Adresse des Moduls auch in den E/A-Bereich gelegt werden.
V02-00.08.00	04.03.2014	Verhalten Im Profibusprotokoll erschien der Wert '-1.175495e-038'. Ursache Es wird vermutet, das ein Sprung an die verkehrte Stelle gemacht wurde. Behebung Sprungmarke gesetzt, sowie Sprunganweisung angepasst im Teil 't074:'.
V02-00.07.00	25.10.2014	Verhalten 1 Die Daten von get detial info waren nicht stimmig. 2 Das Referenzieren wird nicht immer ausgeführt. Ursache 1 Variable vertauscht. 2 Wenn die aktuellen Werte während des Referenzierens abgerufen werden, kann dies auf der Antriebsseite zum Problem führen. Behebung 1 Variable von der Rückgabe vom Modul wurde entsprechend eingesetzt. 2 Die entsprechenden Abfragen wurden unterbunden: – 0xA0 - SET TARGET VEL [Geschwindigkeit] – 0xA1 - SET TARGET ACC [Beschleunigung] – 0xA2 - SET TARGET JERK [Ruck] – 0xA3 - SET TARGET CUR [Strom] – 0xA4 - SET TARGET TIME [Zeit] – 0x96 - GET_DETAILED_ERROR_INFO Folgende Kommandos werden nicht verwendet, die auch zu Problem führen könnten: – 0xA6 - SET TARGET POS [Posi1on] – 0xA7 - SET TARGET POS REL [rel.Positon]

Version	Datum	Bemerkung
V02-00.06.00	26.09.2013	<p>Verhalten Sporadisch werden die Regelungswerte zur Positionierung nicht genommen oder entsprechend angesteuert.</p> <p>Ursache Wenn die Vorgabewerte sich während der Übertragung an den fu ändern, wird der aktuelle Wert zwischen gespeichert, der aber nicht der richtige ist.</p> <p>Behebung Alle regelungsrelevanten Daten werden beim Senden des Auftrage zwischen gespeichert (cmd.: a0..4), um diesen Wert dann als Vergleichswert weiter zu verwenden.</p>
V02-00.05.00	24.09.2013	<p>Verhalten Der Antrieb fährt manchmal nur mit einer langsamen Geschwindigkeit auf die Position, die von der vorherigen Geschwindigkeit gesetzt war.</p> <p>Ursache Der FB20 unterscheidet Tip-v bzw. Regelungs-soll-v, doch das Modul arbeitet nur mit einem Geschwindigkeitswert. Daher kann es vorkommen, das bei der Regelungsfahrt kein soll-v übertragen wird. Daher fuhr das Modul mit der letzten soll-v.</p> <p>Behebung Wenn die Geschwindigkeitsfahrt ausgeführt wird (cmd.:b5) wird der soll-v Zwischenspeicher zurückgesetzt.</p>
V02-00.04.00	04.02.2013	<p>Verhalten Das Statussignal blockiert kommt nicht.</p> <p>Ursache Bei dem FB20 werden die Werte für die Regelung (z. B. Strom, Position) regelmäßig neu gesendet. Wenn der Antrieb in der Regelung ist, wird dadurch das Statusbit zurückgesetzt und neu bewertet, das dann meistens nicht mehr neu angesteuert wird.</p> <p>Behebung Wenn der Antrieb nun in der Regelung ist, wird nur ein Regelungswert neu gesendet, wenn dieser sich verändert hat.</p>
V02-00.03.00	14.12.2012	<p>Verhalten Wenn gleichzeitig die Regelungswerte, die Regelungsart und im gleichen SPS-Zyklus das Start Signal gekommen ist, wurde die Funktion nicht ausgeführt. Erst durch einen erneuten Startimpuls.</p>

Version	Datum	Bemerkung
		Behebung Beim Verändern der Regelungsart (z. B. cmd.#b08 --> cmd.#b0) wurde das Signal '#sd.fkt_mod_delta' auf 1 gesetzt, wodurch der Startimpuls unterdrückt wurde. Jetzt wird das Signal nur angesteuert, wenn keine Start und keine Regelungsfunktion aktiv ist.
V02-00.02.00	27.11.2012	Verhalten Wenn das Modul/Antrieb bereits referenziert ist, konnte keine erneute Referenzierung durchgeführt werden. Behebung Wenn das interne Referenzierungsbit vorhanden ist, kann kein erneutes Referenzieren angesteuert werden. Daher wird nun das interne Referenzierungsbit mit der positiven Flanke zum Starten der Referenzierung zurückgesetzt.
V02-00.01.00	21.11.2012	Verhalten Antrieb wird über cmd xb8 gestartet, fährt auf Position aber meldet nicht, dass die Funktion mit io abgeschlossen wurde und bleibt in der Regelung des FB's. Behebung Das Steuersignal von fu 'fpr position io' wird nicht zurückgesetzt, wenn die gleiche Position erneut wieder angefahren werden soll. Es öffnet zwar die Bremse aber das Signal 'Position io' bleibt erhalten. Daher musste für das Signal zwischen gespeichert werden, welches über Flanken gesetzt/rückgesetzt wird. Hier gab es Probleme. Die Ansteuerung zum Setzen des Signals für Funktion io wurde verbessert. Pointer Adressierung wurde erweitert, so dass auch eine db-Adresse angegeben werden kann.