



Inhaltsverzeichnis

	Seite
1. Allgemeine Beschreibung	2
2. Installation des S7-Programmes / Bausteine	3
2.1 Einbinden der Projektdatei	3
2.2 Hardwarekonfiguration	3
2.3 GSD / GSE Dateien	5
2.4 Bausteinverschachtelung	7
3. Beschreibung der Funktionsbausteine	8
3.1 DRIVECOM-Profil	8
3.1.1 DRIVECOM Steuerwort	8
3.1.2 DRIVECOM Statuswort	9
3.2 Funktion FC30	10
3.2.1 Netzwerk 1, Prozessdaten \leftarrow Antr. Regler	11
3.2.2 Netzwerk 2, Antrieb ein, Netzschütz ein	11
3.2.3 Netzwerk 3, 4 und 5, RFR, HLG (Hochlaufgeber)und QSP \Rightarrow Antr. Regler... ..	11
3.2.4 Netzwerk 6, DRIVECOM \Rightarrow Antr. Regler	11
3.2.5 Netzwerk 7, Prozessdatenworte \Rightarrow Antr. Regler	12
3.2.6 Netzwerk 8, PZD \Rightarrow Antr. Regler	12
3.2.7 Netzwerk 9, DP Parameter S7 \Rightarrow Antr. Regler	13
4. Funktion FC 127	14
5. Datenbaustein DB31	15
5.2 Beispiel DB31	16
5.1 Zusammenwirken der Funktion FC30 mit dem Datenbaustein DB31	17

1. Allgemeine Beschreibung

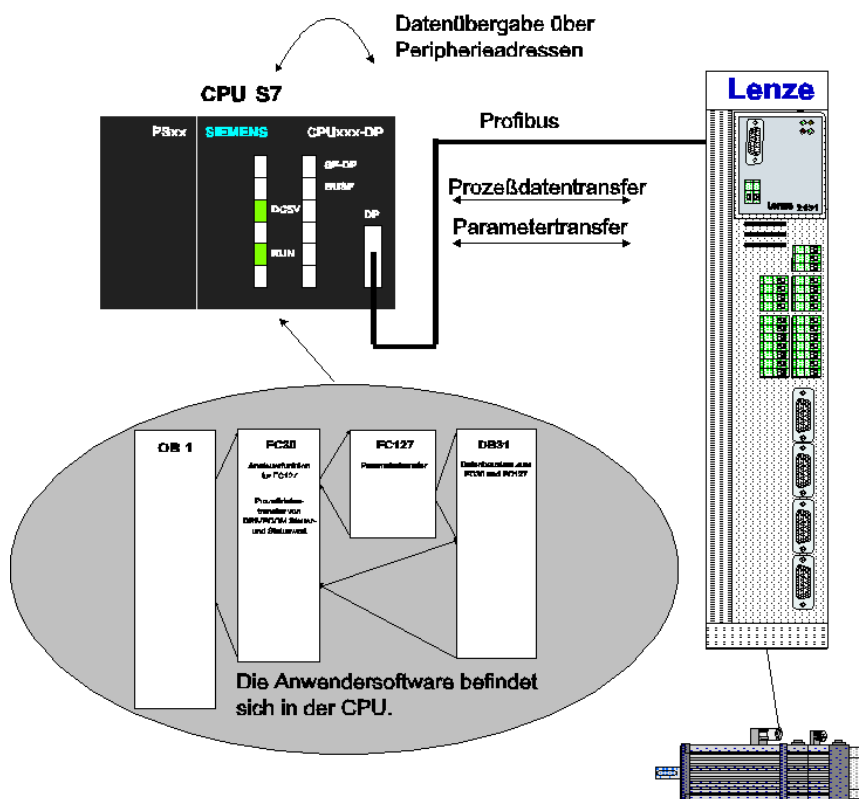
Die Bausteine für S7 sind in Verbindung mit Profibus allgemeingültige Anwendungsbeispiele von Lenze. Die Bausteine sollen bei der Programmierung und Projektierung von Profibus DP Anwendungen mit unseren Produkten unterstützen und Lösungsansätze aufzeigen. Dies ist ein kostenfreies Produkt unseres Unternehmens. Ein Anspruch auf Gewährleistung und Update besteht für den Anwender nicht. Ebenso schließen wir die Haftung für Schäden, die infolge Einsatz dieser Bausteine auftreten, ausdrücklich aus.

Zweck:

Die Profibusschnittstelle besitzt einen Parameterkanal und einen Prozessdatenkanal. Die vorliegenden Funktionsbausteine unterstützen das Parametrieren und das Senden/Empfangen von Prozessdaten.

Parameterkanal: Der Parameterkanal ermöglicht das Lesen und Schreiben von Parameterdaten. Diese Daten müssen nicht zeitkritisch beim Kommunikationspartner vorliegen. Es handelt sich dabei z.B. um Betriebsparameter (z.B. C11 – n_{max}), Diagnoseinformationen (z.B. C161 – aktueller Fehler) oder Motordaten (z.B. C15 – U/f Nennfrequenz).

Prozessdatenkanal: Prozeßdaten müssen in kürzester Zeit zyklisch ausgetauscht werden. Dabei werden Daten wie z.B. das Steuerwort, Statuswort, Soll- und Istwerte übertragen. Die Größe des Prozessdatenkanales ist über die Hardwarekonfiguration festlegbar.



Hardware-Voraussetzungen:

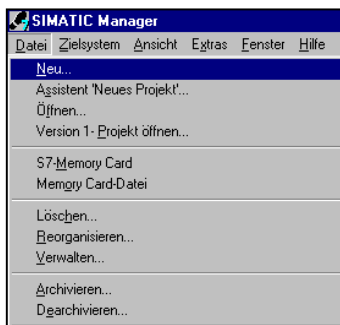
- Antriebsregler mit Profibusschnittstelle (z.B. 2131, 2133, Profibus-Funktionsmodul)
- S7-CPU 3xx/4xx mit einer Profibus-Schnittstelle

2. Installation des S7-Programmes / Bausteine

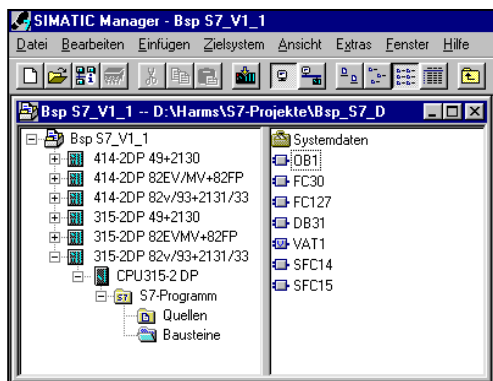
2.1 Einbinden der Projektdatei

1. Wählen Sie im S7-Manager ⇒ **Datei** ⇒ **Dearchivieren**.
2. Wählen Sie die mitgelieferte Datei **D_S7V1_2.zip** aus.
3. Wählen Sie das Zielverzeichnis aus (z.B. Unterverzeichnis "Step7s7proj").

Ein Projekt besteht aus mehreren Komponenten und kann aus einer oder mehreren Stationen (S7-300/400) bestehen. Zu jeder Station gehört eine CPU (z.B. 315-2 DP) und ein oder mehrere S7 Programme mit jeweils einem Bausteinordner, Quellenordner und einer Symboltabelle.



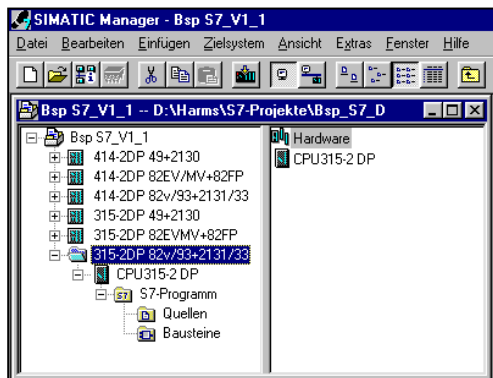
Unter dem Menüpunkt ⇒ **Öffnen** im „SIMATIC Manager“ kann nun das Projekt ausgewählt werden. Es kann auch ein neues Projekt unter dem Menüpunkt ⇒ **Neu** angelegt werden.



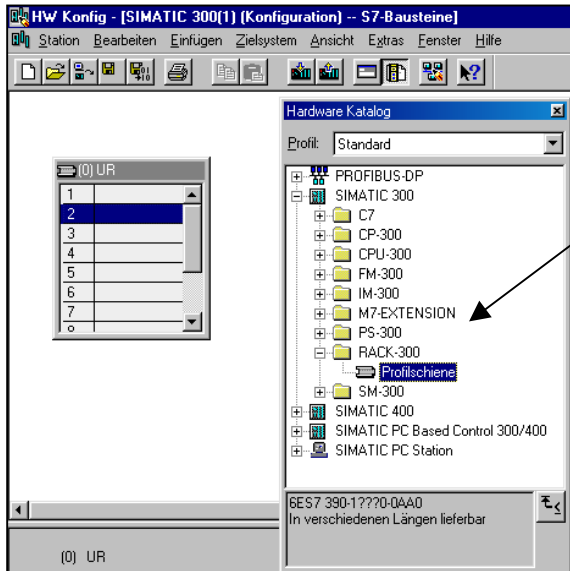
In diesem Projekt befinden sich Teilprojekte, bestehend aus der CPU 315-2 DP und der CPU 414-2 DP und den verschiedenen Profibus Modulen (2130, 2131, usw.). Alle Projekte arbeiten mit den gleichen Bausteinen, die sich durch unterschiedliche Prozessdatenlängen unterscheiden. Das hier im weiteren beschriebene Beispiel bezieht sich nur auf einen Profibus Slave. Wenn mit mehr Teilnehmern gearbeitet wird, sind die Bausteine jeweils einmal pro Teilnehmer verwendbar. Da nicht mit gleichen Bausteinnummern gearbeitet werden darf, sind die Bausteine für jeden Teilnehmer umzubenennen.

2.2 Hardwarekonfiguration

Im folgenden wird das Erstellen einer neuen Hardwarekonfiguration anhand eines Beispiels erläutert. In dieser Beschreibung wurde mit der Step 7 Version 5.1 + Hotfix 2 Ausgabestand K 5.1.0.2 gearbeitet.

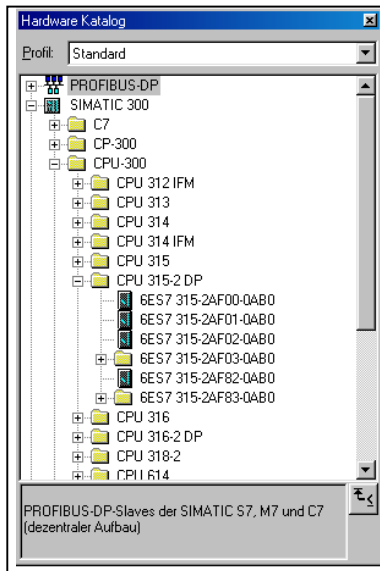


Nachdem Sie ein neues Projekt erzeugt und eine Station (S7-300/400) über den Menüpunkt ⇒ **Einfügen** ⇒ **Station** hinzugefügt haben, gelangen Sie durch einen Doppelklick im rechten Fenster auf **Hardware** in die Hardwarekonfiguration.



Wenn der Hardwarekatalog noch nicht eingeblendet ist, ist dieser unter dem Menüpunkt ⇒ **Ansicht** ⇒ **Katalog** auszuwählen.

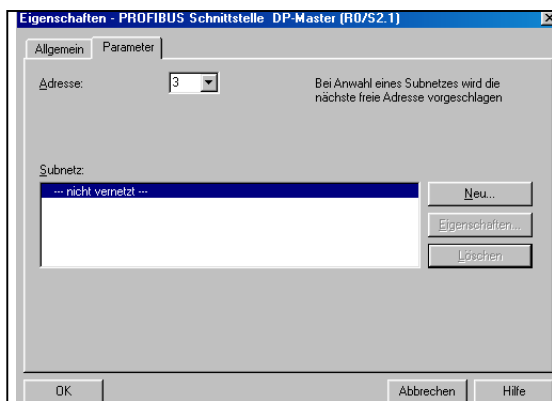
Als erstes eine Profilschiene (Baugruppenträger) unter **Rack** im Katalog auswählen und per Drag and Drop ins freie linke Fenster ziehen.



Aus dem Katalog kann das verwendete Netzteil und die CPU per Drag and Drop ausgewählt werden.

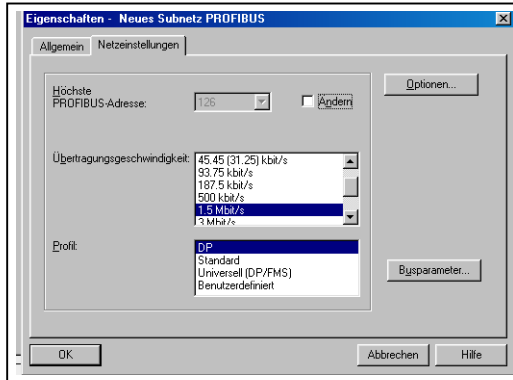
Hinweis: ⇒ Im Katalog stehen bei der CPU 315-2 DP mehrere Typen zur Auswahl. Vergleichen Sie immer die Ausgabestandsnummer ihrer CPU mit der im Katalog, damit Hardwarekonflikte vermieden werden.

Der Ausgabestand der Baugruppe steht an deren Seitenwänden.



Bei Verwendung einer CPU mit DP Master, öffnet sich automatisch das Dialogfeld „Eigenschaften – PROFIBUS Schnittstelle DP-Master“.

Hier wird die DP-Master Adresse eingestellt und über den Button ⇒ **Neu** kann ein Profibus Netz erzeugt werden.



In dem nun erscheinenden Dialogfeld kann in der Registerkarte Netzinstellungen die Baudrate eingestellt werden.

Danach alle Fenster mit **OK** bestätigen. Ihre Hardwarekonfiguration besteht bis jetzt aus einer Stromversorgung, einer DP-Master CPU und einem Profibus Netzwerk.

2.3 GSD / GSE Dateien

Wenn Sie zum ersten Mal mit Lenze Antrieben in Verbindung mit Profibus arbeiten bzw. neue GSD Dateien einfügen wollen, sind diese erst in den Hardwarekatalog zu importieren.

⇒ Hinweis: Die jeweils aktuellen Lenze GSD / GSE Dateien stehen kostenfrei auf der LENZE-Homepage im Internet zur Verfügung.

www.lenze.de

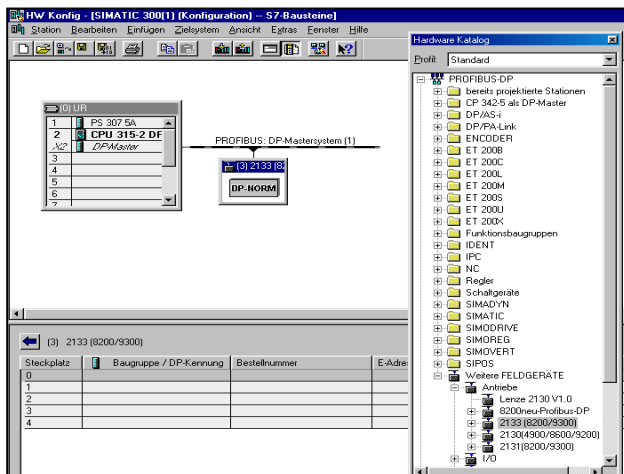
(⇒ Service ⇒ Download ⇒ Profibus)



Installieren der GSD/GSE-Dateien:

Es darf keine Konfiguration im Hardwarekonfigurator geladen sein. Unter dem Menüpunkt ⇒ **Extras** ⇒ **Neue GSD installieren** auswählen. Im folgenden Dialogfeld die neuen GSD Dateien auswählen und mit OK bestätigen.

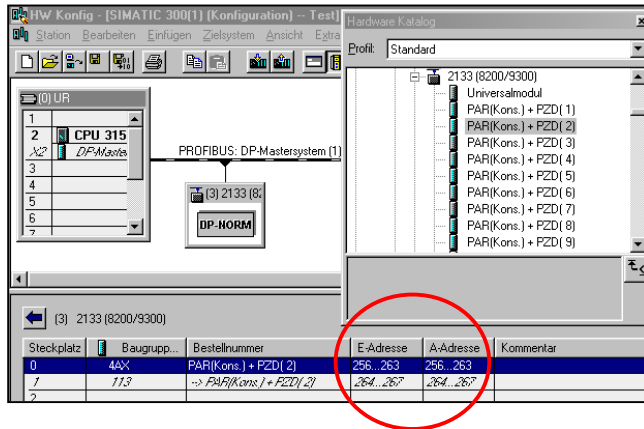
Sie können aber auch alle neuen GSD Dateien in den Ordner ⇒ **Siemens** ⇒ **Step7** ⇒ **S7data** ⇒ **gsd** kopieren und dann im Hardwarekonfigurator den Menüpunkt ⇒ **Extras** ⇒ **Katalog aktualisieren** anklicken.



Generieren des DP-Slaves:

Die Lenze DP-Slave Module befinden sich im Katalog unter ⇒ **PROFIBUS DP** ⇒ **weitere Feldgeräte** ⇒ **Antriebe**.

Um einen Slave einzubinden, ist der ganze Ordner (2133 (8200/9300)) per Drag and Drop auf den Profibus zu ziehen. Im anschließenden Dialogfeld wird die Profibusadresse eingestellt. In diesem Beispiel hat das Modul 2133 die Adresse 3 erhalten.



Nach einem Klick auf das Pluszeichen des jeweiligen Antriebsmoduls können Sie aus den unterstützten Prozessdatenformaten auswählen. Dazu ist der DP-Slave zu markieren und per Drag and Drop in die Tabelle unterhalb der projektierten Hardware zu ziehen.

In der dazugehörigen Tabelle des Slaves wurden 2 Spalten ausgefüllt. **Die erste Spalte gibt den Parameterkanal und die zweite den Prozessdatenkanal an.** Wichtig sind hier die Spalten E/A-Adresse (roter Kreis).

⇒ Hinweis: Die Kommunikation kann nach verschiedenen Prozeßdatenformaten erfolgen.

Auszug aus dem Hardware Katalog:

```

Hardware Katalog
~ PROFIBUS DP
  :
  ~ weitere Feldgeräte
    ~ Antriebe
      Lenze 2130 V1.0
        * 8200neu-Profibus DP
          ~ 2133 (8200/9300)
            :
            Par(Kons.) + PZD(1W) AR
            :
            PPO1
            :
            Par(Kons.) + PZD(1W)
            :
            * 2130 (4900/8600/9200) (GSD)
            * 2131 (8200/9300)
    
```

Profil	Name	Bemerkung
AR	Lenze Geräte- steuerung	direkter Zugriff auf AIF-CTRL
PPO	Profidrive	Profidrive Zustandsmaschine
ohne Kenn- zeichen	Drivecom	Drivecom Zustandsmaschine

Die Belegung des Steuer- und Statuswortes entnehmen Sie bitte der Betriebsanleitung des jeweiligen Moduls.

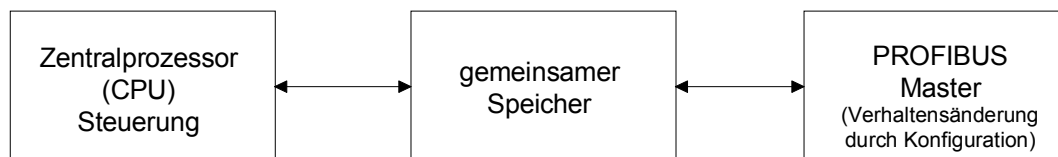
Die Bezeichnung (Kons.) bedeutet, daß die Daten (Parameter und/oder Prozeßdaten) konsistent übertragen werden.

⇒ Hinweis: Wenn Parameter mit der Funktion FC127 übertragen werden, ist mit Datenkonsistenz zu arbeiten.

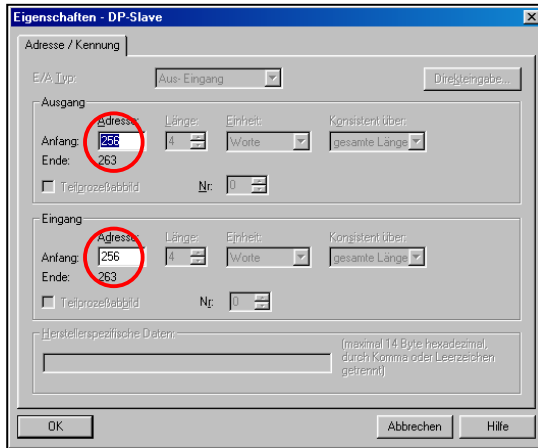
Was ist Konsistenz?

Es wird von „Konsistenz“ gesprochen, wenn es einen Bereich von mehr als einem Wort bzw. einem Byte zusammenhängende Daten gibt. Die eingeschaltete "Konsistenz" sorgt dann für einen fehlerfreien Datenaustausch zwischen Zentralprozessor (CPU) der Steuerung und dem PROFIBUS Master über den gemeinsamen Speicher (Dual Port Memory). Das bedeutet, wird die Konsistenz nicht aktiviert, könnten z.B. beim Leseauftrag Daten in die CPU eingehen, die nicht zusammengehören. Somit könnte eine fehlerhafte Information entstehen.

Beispiel: ein Parameter soll gelesen werden



Wartet, bis Daten vollständig gelesen. Danach wird vollständiger Datensatz in den gemeinsamen Speicher geschrieben. Schreibt nur, wenn CPU nicht liest.

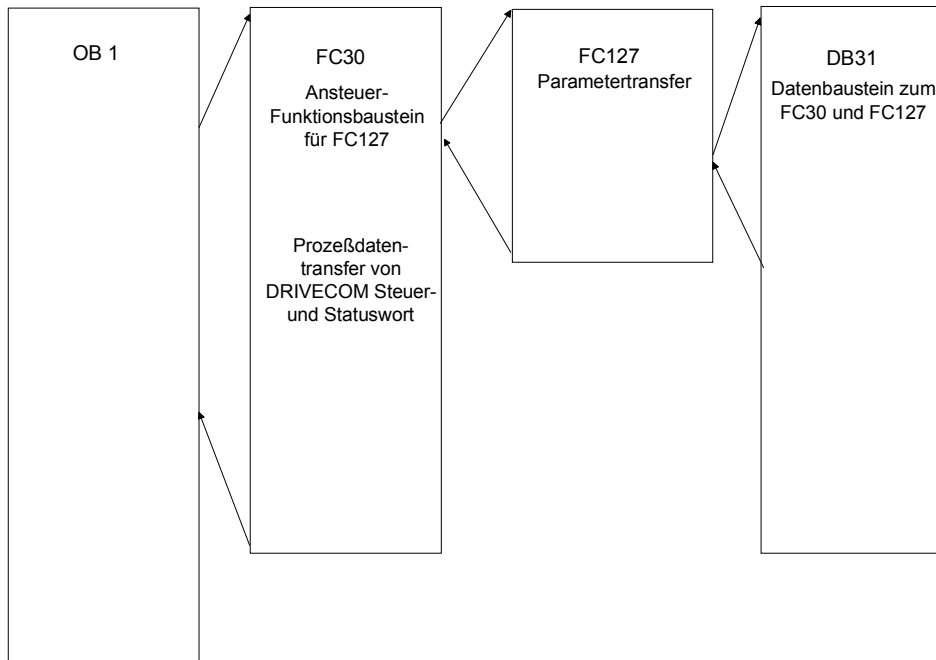


Die Peripherie-Anfangsadressen werden im späteren SPS-Programm benötigt. Die E/A-Adressen können auch verändert werden.

Durch einen Doppelklick auf die jeweilige Spalte öffnet sich ein Dialogfeld, in dem der Adressbereich verändert werden kann. Der Adressbereich für eine Spalte (Ein- und Ausgangsadresse) sollte immer gleich sein (rote Kreise). Dies ist übersichtlicher und erleichtert die spätere Arbeit im Anwenderprogramm.

2.4 Bausteinverschachtelung

Der Organisationsbaustein OB1 ruft den Funktionsbaustein FC30 zyklisch auf.



Die Funktion FC30 ist ein Beispiel und zeigt, welche Voraussetzungen bezüglich des Parameter- und Prozessdatentransfer zu schaffen sind.

Außerdem beschreibt der Baustein über dem Prozessdatenkanal das DRIVECOM-Steuerwort, bzw. das DRIVECOM-Statuswort wird über diesen Baustein ausgelesen. Er arbeitet somit nach der DRIVECOM Zustandsmaschine und bewirkt auf diese Art und Weise Reglerfreigabe über den PROFIBUS, vorausgesetzt, daß die DRIVECOM Zustandsmaschine aktiviert wurde. Des weiteren zeigt der Baustein FC30, wie die Funktion FC127 angewendet wird und somit PROFIBUS Parameter gelesen bzw. geschrieben werden können.

Dabei werden die entsprechenden Codestellen, die gelesen oder beschrieben werden sollen im Datenbaustein DB31 eingetragen.

Für den Parametertransfer sind die Systembausteine SFC 14 und SFC 15 zu laden. Diese werden zur konsistenten Datenübertragung benötigt.

3. Beschreibung der Funktionsbausteine

Im folgenden sollen nun die einzelnen Bausteine erläutert werden. Die Bausteine sind in Anweisungsliste geschrieben.

3.1 DRIVECOM-Profil

Der Baustein FC30 unterstützt das DRIVECOM-Profil. Er setzt voraus, daß das erste Wort des Prozeßdatenkanales das DRIVECOM Steuerwort bzw. Statuswort ist. Nach diesem Profil ist eine bestimmte Reihenfolge beim Freigeben des Reglers einzuhalten. Der FC30 arbeitet diese automatisch ab. Die Funktionalität des DRIVECOM Steuerwortes entnehmen Sie bitte dem nächsten Kapitel.

3.1.1 DRIVECOM Steuerwort

In Klammern stehen die Netzwerknummern (NW) des FC30, in denen die Bits bearbeitet werden.

Bit	0 := Einschalten	<- wird vom Funktionsbaustein angesprochen (NW 6) (sollte nicht vom Programmierer angesprochen werden)
	1 := Spannung-Sperren	<- wird vom Funktionsbaustein angesprochen (NW 6) (sollte nicht vom Programmierer angesprochen werden)
	2 := Schnellhalt	<- wird vom Funktionsbaustein angesprochen (NW 6) (sollte nicht vom Programmierer angesprochen werden)
	3 := Betrieb-Freigegeben	=> Reglersperre := 0; Reglerfreigabe := 1, (NW 3)
	4 := HLG-Sperren	=> QSP := 0; QSP nicht aktiv := 1, (NW 5)
	5 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133) WICHTIG! Werksabgleich bei den meisten Antriebsregler: HLG-Stoppen := 0; HLG_Stoppen nicht aktiv := 1 (NW 4)
	6 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133) WICHTIG! Werksabgleich bei den meisten Antriebsregler: HLG-Null := 0; HLG-Null nicht aktiv := 1 (NW 4)
	7 := Reset-Störung	Bitwechsel von 0 nach 1
	8 := DRIVECOM reserviert	
	9 := DRIVECOM reserviert	
	10 := DRIVECOM reserviert	
	11 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)
	12 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)
	13 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)
	14 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)
	15 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)

3.1.2 DRIVECOM Statuswort

Der augenblickliche Gerätezustand ist im DRIVECOM Statuswort vorhanden.

Bit	0 := Einschaltbereit	-> Information für den Funktionsbaustein
	1 := Eingeschaltet	-> Information für den Funktionsbaustein
	2 := Betrieb-Freigegeben	-> Information für den Funktionsbaustein
	3 := Störung	<= keine Störung (TRIP) := 0; Störung (TRIP) := 1
	4 := Spannung-Gesperrt	-> Information für den Funktionsbaustein
	5 := Schnellhalt	DRIVECOM Schnellhalt := 0; kein Schnellhalt := 1
	6 := Einschaltsperr	-> Information für den Funktionsbaustein
	7 := Warnung	keine Warnung := 0; Warnung := 1
	8 := Meldung	keine Meldung := 0; Meldung := 1
	9 := Remote	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)
	10 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)
	11 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)
	12 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)
	13 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)
	14 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)
	15 := FREI DRIVECOM	siehe Profibus Betriebsanleitung (z.B. 2131, 2133)

Die freien Bits des Steuer- bzw. Statuswortes sind frei belegbar. Informieren Sie sich bitte in der jeweiligen Profibus Betriebsanleitung, da die Zuordnung der Bits für die verschiedenen Regler unterschiedlich sein kann.

Welche Technische Betriebsanleitung beschreibt was?

Betriebsanleitung Feldbusmodul	Profibusmodul für die Antriebsregler:
2130	4800, 4900, 8600 und 9200
2131, 2133	8200 und 9300
„Feldbus-Funktionsmodule für Frequenzumrichter“	8200vector und 8200motec

3.2 Funktion FC30

Der FC30 wird entweder im OB1 oder in einem anderen Baustein aufgerufen. Die Funktion wird an dieser Stelle Schritt für Schritt erläutert. Hierbei handelt es sich um den FC30 mit dem Funktionsmodul 2131.

Hinweis: Wenn mit konsistenter Datenübertragung gearbeitet werden soll, sind die Systembausteine SFC 14 und SFC 15 in die SPS zu laden.

Der Baustein arbeitet mit zwei temporären Datenstrukturen in der Variablen-tabelle.

data_receive (Empfang von Daten)
 data_send (Senden von Daten)

In diesem Beispiel ist der Baustein FC30 für eine Kommunikation mit 3 Prozeßdatenworten (PZD) eingerichtet. In der Variablen-tabelle ergibt sich je Struktur eine Größe von 6 Bytes bzw. 3 Worten.

Hinweis: Ob die Datenübertragung Byte – oder wortweise geschieht, entnehmen Sie bitte der Beschreibung Ihres Profibus Masters.

Die Variablen-tabelle muß auf die Anzahl der Prozeßdatenworte angepaßt werden.

Beispiel:

Adresse	Deklaration	Name	Typ	Kommentar
+2.0	temp	Prozessdatenwort_1	INT	Prozessdatenwort 1 <- Antr.-Regler
+4.0	temp	Prozessdatenwort_2	INT	Prozessdatenwort 2 <- Antr.-Regler
+6.0	temp	Prozessdatenwort_3	INT	Prozessdatenwort 3 <- Antr.-Regler
+8.0	temp	Prozessdatenwort_4	INT	Prozessdatenwort 4 <- Antr.-Regler
=8.0			END_STRUCT	

Die Tabelle zeigt die Struktur data_receive mit 4 PZD. Der gleiche Schritt muß explizit für die Struktur data_send gemacht werden.

Der FC30 besteht aus insgesamt 9 Netzwerken und unterstützt das DRIVECOM-Profil. Im folgenden wird jedes Netzwerk erläutert.

3.2.1 Netzwerk 1, Prozessdaten \Leftarrow Antr. Regler

Netzwerk 1

```
// 1.#####
// konsistente Übertragung über 1 Prozessdatenwort
// Prozessdatenwort 0 (Statuswort)
// L PEW 264 // 1.Peripherieadresse der Prozessdatenübertragung
// T LW 0 // temp. Lokaldatenvariable
// Prozessdatenwort 1 (Istwert)
// L PEW 266
// T #data_receive.Prozessdatenwort_1 // temp. Lokaldatenvariable

// 2.#####
// konsistente Übertragung über 2 Prozessdatenworte
// L PED 264 // 1.Peripherieadresse der Prozessdatenübertragung
// T LD 0 // temp. Lokaldatenvariable

// 3.#####
// konsistente Übertragung über mehr als 2 Prozessdatenworte
CALL "DPRD_DAT"
LADDR :=W#16#108 // 1.Peripherieadresse der Prozessdatenübertragung
RET_VAL:=#data_receive_ret_val
RECORD :=#data_receive // Prozessdaten <- Antr.-Regler
```

Im Netzwerk 1 sind drei konsistente Übertragungen der PZD zum Antriebsregler vorbereitet.

1. konsistente Übertragung über 1 PZD
2. konsistente Übertragung über 2 PZD
3. konsistente Übertragung über mehr als 2 PZD

Wählen Sie die Übertragung, die Sie bei der Hardwarekonfiguration ausgewählt haben, und löschen oder kommentieren Sie die andere aus.

\Rightarrow Achtung: Die Peripherieadressen müssen mit denen aus der Hardwarekonfiguration übereinstimmen und werden hexadezimal eingetragen.

3.2.2 Netzwerk 2, Antrieb ein, Netzschütz ein

Dieses Netzwerk kann zur eigenen Verknüpfung mit dem Anwenderprogramm genutzt werden.

3.2.3 Netzwerk 3, 4 und 5, RFR, HLG (Hochlaufgeber) und QSP \Rightarrow Antr. Regler

Netzwerk 3

```
SET //ersetzt Verknüpfung des Anwenderprogramms
= #data_send.Steuerwort.RFR_Antr_Regler
```

Netzwerk 4

```
/"HGL-Stoppen/HGL-freigegeben"
SET // ersetzt Verknüpfung des Anwenderprogramms
= #data_send.Steuerwort.HGL_nicht_Stoppen

/"HGL-Null/HGL-freigegeben"
= #data_send.Steuerwort.HGL_nicht_Null
```

In diesen Netzwerken wird die Reglerfreigabe RFR, QSP (QuickStop) und HLG (Hochlaufgeber) aktiviert, bzw. nicht aktiviert. Die Befehle SET sind durch das Anwenderprogramm ersetzbar.

Netzwerk 5

```
SET // ersetzt Verknüpfung des Anwenderprogramms
= #data_send.Steuerwort.NOT_QSP_Antr_Regler
```

3.2.4 Netzwerk 6, DRIVECOM \Rightarrow Antr. Regler

Netzwerk 6

```
///## Gerätezustand (Drivecom) -> Antr.-Regler
// => "Eingeschaltet"
SET
S #data_send.Steuerwort.DRIVECOM_Einschalten
S #data_send.Steuerwort.DRIVECOM_Spg_sperren
S #data_send.Steuerwort.DRIVECOM_Schnellhalt
// => "Einschaltbereit"
UN #data_receive.Statuswort.DRIVECOM_Einschaltbereit
UN #data_receive.Statuswort.Stoerung_TRIP
R #data_send.Steuerwort.DRIVECOM_Einschalten
```

Das Netzwerk darf nicht verändert werden. Nach erfolgter Freigabe im Netzwerk 3, 4 und 5 und Vorgabe eines Geschwindigkeitssollwertes wird der Antriebsregler vom DRIVECOM Zustand „Einschaltbereit“ in den Zustand „Eingeschaltet“ wechseln. Der Motor dreht dann mit der vorgegebenen Drehzahl, vorausgesetzt der Antriebsregler wurde über die Steuerklemmen (Kl. 28) freigeschaltet.

3.2.5 Netzwerk 7, Prozessdatenworte ⇒ Antr. Regler

Netzwerk 7

```
// Prozessdatenwort 1 (Sollwert 1)
L 0 // ersetzt Anwenderdatum
T #data_send.Prozessdatenwort_1
```

In diesem Netzwerk werden die Prozessdateninhalte zum Regler definiert.
⇒ Hinweis: Die Variablen-tabelle ist anzupassen, wenn Sie mit mehr als 3 Prozeßdatenworten arbeiten möchten.

3.2.6 Netzwerk 8, PZD ⇒ Antr. Regler

Netzwerk 8

```
// 1. #####
// konsistente Übertragung über 1 Prozessdatenwort
// Prozessdatenwort 0 (Steuerwort)
L LW 4 // temp. Lokaldatenvariable
T "PAW 264" // 1.Peripherieadresse der Prozessdatenübertragung
// Prozessdatenwort 1 (Sollwert)
L #data_send.Prozessdatenwort_1 // temp. Lokaldatenvariable
T "PAW266"

// 2. #####
// konsistente Übertragung über 2 Prozessdatenworte
// I LD 4 // temp. Lokaldatenvariable
// t PAD 264 // 1.Peripherieadresse der Prozessdatenübertragung

// 3. #####
// konsistente Übertragung über mehr als 2 Prozessdatenworte
CALL "DPWR_DAT"
LADDR :=W#16#108 // 1.Peripherieadresse der
Prozessdatenübertragung
RECORD :=#data_send // Prozessdaten <- Antr.-Regler
RET_VAL:=#data_send_ret_val // Rückgabewert (Fehlercode) des SFC's
```

An dieser Stelle werden die Prozessdateninhalte (entsprechend NW 7) zum Antriebsregler geschrieben.

Analog zum Netzwerk 1, gibt es auch hier die drei Auswahlmöglichkeiten zum Übertragen der Daten. Die nicht benötigten sind entweder zu löschen oder auszukommentieren.

Die Peripherieadressen sind ebenso an die Hardwarekonfiguration anzupassen.

Zusammenfassung Prozeßdaten:

Soll mit 3 PZD (konsistent über jeweils ein Wort) gearbeitet werden, sind lediglich die Netzwerke 1 und 8 anzupassen. Die restlichen Netzwerke sind auszufüllen (NW 2,3,4,5 und 7). Wenn Sie mit mehr oder weniger PZD arbeiten, ist zusätzlich:

- die Variablen-tabelle anzupassen.
- Netzwerk 7 anzupassen.

3.2.7 Netzwerk 9, DP Parameter S7 ⇒ Antr. Regler

```

### Aufträge aktivieren =>
/# 01) C 011 -> Antr.-Regler
//- Sollwert -> Auftragsliste
L L#50 // ersetzt Vorgabe vom Anwenderprogramm
L L#10000 // 4 Kommastellen berücksichtigen
*D // => 50 Hz
T "DB31".telegram_001.data
//- Auftrag aktivieren
SET // ersetzt Vorgabe vom Anwenderprogramm
= DB31.DBX 8.6

/# 02) C 050 <- Antr.-Regler
//- ? Auftrag fertig ohne Fehler
O DB31.DBX 16.6 // Auftrag aktiviert
O DB31.DBX 16.7 // Auftrag mit Fehler
SPB nOK2
//- Istwert <- Auftragsliste
L "DB31".telegram_002.data
L L#10000 // 4 Kommastellen berücksichtigen
/D // => in min-1
//T xx // ersetzt Transferziel vom Anwenderprogramm
nOK2: NOP 0
//- Auftrag aktivieren
SET // ersetzt Vorgabe vom Anwenderprogramm
= DB31.DBX 16.6

/# 03) C 012 -> Antr.-Regler
//- Sollwert -> Auftragsliste
L L#123 // ersetzt Vorgabe vom Anwenderprogramm
L L#1000 // 4 Kommastellen berücksichtigen
*D // => 12.3sec
T "DB31".telegram_003.data
//- Auftrag aktivieren
SET // ersetzt Vorgabe vom Anwenderprogramm
= DB31.DBX 24.6

/# 04) C 013 -> Antr.-Regler
//- Sollwert -> Auftragsliste
L L#123 // ersetzt Vorgabe vom Anwenderprogramm
L L#100 // 4 Kommastellen berücksichtigen
*D // => 1.23sec
T "DB31".telegram_004.data
//- Auftrag aktivieren
SET // ersetzt Vorgabe vom Anwenderprogramm
= DB31.DBX 32.6

### Aufträge abwickeln
CALL FC 127
peripherieadr_1st_byte:=256 // Peripherieanfngsadresse
DB_transferlist := "DB31" // DB mit der Auftragsliste
begin_DB_transferlist := 0 // Beginn der Auftragsliste im DB
enable_transfer := TRUE // Freigabe der Parameterübertragung
timeout_timer := T35 // Überwachung Auftragslaufzeit

```

Im Netzwerk 9 befindet sich eine Auftragsliste, in der vier Parameteraufträge angestoßen werden für die Codestellen:

C011 (fdmax),
 C051 (fdist),
 C012 (Hochlaufzeit) und
 C013 (Ablaufzeit).

Ein Parameterauftrag ist 8 Byte, bzw. 4 Worte groß. Die einzelnen Aufträge werden in den Datenbaustein 31 geschrieben. Am Ende der Auftragsliste wird der FC127 aufgerufen, der die Aufträge der Reihe nach abarbeitet.

Jeder Auftrag ist einzeln zu aktivieren. Dies geschieht über das Lenze Servicebyte. Näheres dazu bei der Beschreibung des DB31 (Kapitel 5, Seite 15).

4. Funktion FC 127

Diese Funktion arbeitet die einzelnen Parameternaufträge ab. Sie arbeitet nur mit einer konsistenten Parameterkonfiguration, weil Parameter grundsätzlich konsistent übertragen werden sollten.

Achtung ⇒ In der Hardwarekonfiguration muß über den gesamten Bereich der Peripherieadressen für den Parameterkanal die Konsistenz eingeschaltet sein.

CALL FC127

peripherieadr_1st_byte	:=256	Peripherieanfangsadresse in Dezimal
DB_transferlist	:=„DB31“	DB mit der Auftragsliste
begin_DB_transferlist	:=0	Beginn der Auftragsliste im DB
enable_transfer	:=TRUE	Freigabe der Parameterübertragung, wird nach fehlerfreier Rückmeldung wieder auf FALSE gesetzt
timeout_timer	:=T35	Überwachung Auftragslaufzeit

begin_DB_transferlist: Hier wird festgelegt, ab welcher Auftragsnummer im DB31 die Aufträge bearbeitet werden sollen. Vorzugsweise sollte immer bei Null angefangen werden.

enable_transfer: Mit diesem Bit wird der Parameterübertrag angestoßen.

timeout_timer: Der Timer überwacht die Auftragslaufzeit des gerade bearbeiteten Parameternauftrages. Wenn der Auftrag nicht binnen 5s fehlerfrei abgearbeitet worden ist, wird das Bit 7 im Servicebyte auf Eins gesetzt. Danach wird der nächste Auftrag bearbeitet.

5. Datenbaustein DB31

Der DB31 besteht aus 3 Parameternaufträgen und ist beliebig erweiterbar. Das Format eines Auftrages ist immer identisch. Die ersten 4 Worte des Bausteines dürfen nicht verändert werden.

Die 8 Bytes eines Auftrages haben folgende Bedeutung:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Servicebyte	Subindex	Index High Byte	Index Low Byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

1. Byte: Servicebyte => über dieses Byte wird die Datenlänge des zu übertragenden Parameter angegeben und ob ein Lese- oder Schreibauftrag ausgelöst werden soll.
- | | | | | | | | | | | | | | | | |
|-------|--|---|--|---|--|---|--|---|--|---|--|---|-------------------------------|---|--|
| Bit 7 | | 6 | | 5 | | 4 | | 3 | | 2 | | 1 | | 0 | |
| | | | | | | | | | | | | | 0 1 => Auftrag lesen | | |
| | | | | | | | | | | | | | 1 0 => Auftrag schreiben | | |
| | | | | | | | | | | | | | ↪ wird zur Zeit nicht genutzt | | |
| | | | | | | | | | | | | | ↪ reserviert | | |
| | | | | | | | | | | | | | 0 0 => 1 Byte Datenlänge | | |
| | | | | | | | | | | | | | 0 1 => 2 Byte Datenlänge | | |
| | | | | | | | | | | | | | 1 0 => 3 Byte Datenlänge | | |
| | | | | | | | | | | | | | 1 1 => 4 Byte Datenlänge | | |
- ↪ **Handshakebit:** Um einen Lese- bzw. Schreibauftrag zu aktivieren, ist vom Anwender dieses Bit zu setzen. Der Baustein setzt nach erfolgter Übertragung (mit/ohne Fehler) dieses Bit zurück.
- ↪ **Fehlerbit:** Im Falle eines Auftragfehlers ist dieses Bit gesetzt. Wird danach ein erfolgreicher Auftrag abgesetzt, wird das Bit nach der Bearbeitung zurückgesetzt.
- Die Datenlänge des Parameters ist anhand der Attributtabelle des Systemhandbuchs (Kapitel „Codetabelle“) ersichtlich.**

2. Byte: Subindex der Codenummer, die beschrieben bzw. gelesen werden soll.
- 3-4. Byte: Index bzw. Codenummer des Antriebsreglers, der gelesen bzw. beschrieben werden soll, Berechnung: 24575 - Code-Nr.
- 5-8. Byte: Daten, die beim Auftrag „Lesen“ zum Antriebsregler gesendet bzw. beim Auftrag „Schreiben“ vom Antriebsregler zur Steuerung gesendet wird.

Daten:

- Die Datenlänge und das Codestellenformat entnehmen Sie bitte der Attributtabelle der Codetabelle des Systemhandbuchs des verwendeten Gerätes. Das gängige Datenformat ist das Festkommaformat mit 4 Nachkommastellen.

Bei diesem Format sind die Parameter mit dem Wert 10000 zu multiplizieren.

Im Datenbaustein DB31 werden folgende Informationen jeweils für einen Parameternauftrag übertragen:

- Lese- oder Schreibauftrag
- Datenlänge
- Index (bzw. Codenummer des Parameters)
- Subindex von der Codenummer des Parameters
- eigentliche Daten

Hinweis bei Auftritt von Fehlern

- Das Fehlerbit (Bit 7) im Servicebyte sollte ausgewertet werden. Wird ein Fehler bei einem Parameterauftrag gemeldet, so steht in den Datenbytes (Byte 5-8) ein Fehlercode. Die Beschreibung zu dem Dateninhalt der möglichen Fehlermeldung entnehmen Sie bitte der Technischen Beschreibung für das verwendete Profibusmodul.
- Bei Auftritt eines Fehlers steht in der Adresse 4 „error“ (Doppelwort) vom Datenbaustein 31 (W4 und W6) der zuletzt aufgetretene Fehlercode. Dieser Adressbereich ist als Speicher ausgelegt, so daß deren Inhalt erst beim Auftreten eines neuen Fehlers mit diesem überschrieben wird.

5.2 Beispiel DB31

Aufgabenstellung: Der Wert der Codestelle C012 soll auf 12,3s verändert werden.

Lösung:

```

telegram_001STRUCT          ##### C 012 -> Antr.-Regler
1.0  servicebyte           BYTE    B#16#32  Servicebyte -> Lenze
2.0  subindex              BYTE    B#16#0    Subindex (Subcode)
4.0  index                 INT     24563     Index = 24575 - Code-Nr.
8.0  data                  DINT    L#123000  Auftragsdatum ( Format beachten: Kommastellen)
=8.0  END_STRUCT
    
```

Erläuterung:

In der ersten Spalte sehen Sie die Speicherbereiche, die sich aus den verwendeten Datentypen ergeben.

```

BYTE           := eine Speicherstelle
WORT/INT       := zwei Speicherstellen
DINT/DWORD     := vier Speicherstellen
    
```

Im Servicebyte ist ein Schreibauftrag mit einer Datenlänge von vier Byte deklariert. Die Codestelle, die beschrieben werden soll hat keinen Subindex (=0).

$$24575 - 12 = 24563$$

Die Hochlaufzeit soll auf den Wert 12,3 s eingestellt werden. Im Bereich „data“ steht der neue Wert der Codestelle.

$$12,3 \times 10000 = 123000 \text{ (4 Kommastellen berücksichtigen)}$$

5.1 Zusammenwirken der Funktion FC30 mit dem Datenbaustein DB31

Der Datenbaustein DB31 wird mit den durchzuführenden Aufträgen aufgefüllt. Hierzu stehen 2 Möglichkeiten zur Auswahl.

- 1.) Es wird mit der Auftragsliste im FC30 gearbeitet und die Daten werden von dort in den DB31 geschrieben.
- 2.) Im DB31 werden alle erforderlichen Daten als Anfangswerte angelegt.

Zum Anstoßen eines Auftrages ist das Handshakebit (Bit 6) im Servicebyte zu setzen. Dies geschieht im FC30 (NW 9), der den Funktionsbaustein FC127 ansteuert.

Die schwarz markierten Programmzeilen setzen das Handshakebit.

```
//# 03) C 012 -> Antr.-Regler
//- Sollwert -> Auftragsliste
L L#123          // ersetzt Vorgabe vom Anwenderprogramm
L L#1000         // 4 Kommastellen berücksichtigen
*D              // => 12.3sec
T "DB31".telegram_003.data
// - Auftrag aktivieren
SET           // ersetzt Vorgabe vom Anwenderprogramm
= DB31.DBX 24.6
```

Die Funktion FC127 wechselt automatisch den Zustand des Handshakebit nach erfolgter Übertragung vorausgesetzt, daß die Antwort richtig eingegangen ist. Wenn die Parameter zyklisch eingelesen werden sollen, so ist permanent das Handshakebit in dem Ansteuer-Funktionsbaustein (z.B. FC30) zu setzen (SET).

Die Funktion FC127 durchläuft den Datenbaustein DB31 und vergleicht, ob ein Auftrag aktiviert wurde. Der FC127 fängt wieder mit dem erneuten Durchlauf des Datenbausteins an, wenn als letzter Auftrag im Service-Byte die Information „30 hex“ steht.