

Technische Beschreibung der Profibus-Kommunikation

zwischen Simatic S7 Steuerungen und Emotron-Frequenzumrichtern

FDU/VFX 2.0

Übersicht der Varianten für die Profibus-Kommunikation:

1. Steuern des Gerätes und Auslesen von Drehzahl, Drehmoment, Strom, Leistung und Kühlkörpertemperatur
(Ohne DP-V1 Funktionalität, nur über den Prozessdatenkanal.)
(Firmware des Gerätes muss mindestens Version 4.06 sein.)

Simatic-Projekt: L2DP-M07.ZIP

L2DP-M08.ZIP (zusätzliches Auslesen der Digitaleingänge)

L2DP-M09.ZIP (zusätzliche Auswahl des Antriebsmodus bei VFX und Schalten der Relais des Frequenzumrichters)

2. Steuern des Gerätes und Auslesen von fünf Werten, die aus einer Auswahl zusammengestellt werden können. (Mit DP-V1 Funktionalität.)

Simatic-Projekt: L2DPV1-K.ZIP

L2DPV1-K01.ZIP (zusätzliche Auswahl des Antriebsmodus bei VFX)

3. Steuern des Gerätes und freier Zugriff (lesend und schreibend) auf alle Daten. (Mit DP-V1 Funktionalität.)

Simatic-Projekt: L2DPV1-E.ZIP

1. Version zur Steuerung und zum festen Lesen von Daten aus dem Frequenzumrichter

In dieser Version können Drehzahl, Drehmoment, Strom, Leistung und Kühlkörpertemperatur aus dem Frequenzumrichter in die S7 gelesen werden.

(In der Version L2DP-M08.ZIP auch zusätzlich die Digitaleingänge des Gerätes.)

1.1. Simatic S7 Beispielprojekt

Emotron stellt ein archiviertes S7-Beispielprojekt (L2dp_M07.zip) zur Verfügung, in dem alle notwendigen Bausteine für die Profibus-Kommunikation mit den Frequenzumrichtern enthalten sind.

Folgende Bausteine sind im Beispielprojekt enthalten:

- OB1 Beispiel für den Aufruf des Funktionsbausteins
- FC177 Funktion für die Kommunikation mit den Emotron-Frequenzumrichtern
- FC10 dient dem Aufruf des FC177 (und der Skalierung des Sollwertes)
- VAT2 Variablen-tabelle für Testzwecke (das Projekt ist lauffähig).

1.2. Konfiguration im S7-Projekt

Als Erstes ist in der Hardware-Konfiguration die gsd-Datei Version 2.0 einzubinden.

Steckplatz	DP-Kennung	Bestellnummer / Bezeichnung	E-Adresse	A-Adresse	Komm...
1	144	Input 1 byte	8		
2	144	Input 1 byte	9		
3	144	Input 1 byte	10		
4	144	Input 1 byte	11		
5	209	Input 2 words	12..15		
6	160	Output 1 byte		8	
7	160	Output 1 byte		9	
8	160	Output 1 byte		10	
9	160	Output 1 byte		11	
10	225	Output 2 words		12..15	
11					
12					
13					
14					
15					
16					
17					
18					

Nach dem Laden und Aktualisieren des Hardware-Katalogs lässt sich der Frequenzumrichter folgendermaßen im Verzeichnisbaum finden:

PROFIBUS-DP – Weitere Feldgeräte – Allgemein – **Anybus-CC PROFIBUS DP V1**

Dieses Gerät ist am Bus zu installieren, dann sind von diesem Gerät viermal „Input 1 Byte“, einmal „Input 2 words“, viermal „Output 1 Byte“ und einmal „Output 2 words“ einzufügen. (siehe Bild)

Die Adressen für die Eingangs- und Ausgangsperipherie sind einzutragen.

Es ist wichtig, wie im Bild zu sehen, dass die **gleiche Basisadresse** für Ein- und Ausgang verwendet wird (im Beispiel auf dem Bild die Adresse „8“).

Diese Adresse wird später auch für die Konfiguration der Kommunikation benötigt.

Die Adresse des Frequenzumrichters am Profibus (im Beispiel „10“) ist am Gerät selbst im Parameter **2631** einzutragen.

Bei dieser Variante ist der Parameter **2632** (Datengröße) auf „8“ zu stellen.

Achtung ! – Wenn die Profibus-Adresse (Parameter 2631) oder die Datengröße (Parameter 2632) im Gerät geändert wird, so muss die Versorgungsspannung ausgeschaltet werden, das Gerät muss völlig inaktiv sein, und dann kann die Versorgungsspannung wieder eingeschaltet werden, erst dadurch ist der geänderte Wert wirksam !

1.3. Arbeitsweise der Bausteine im Beispielprojekt

OB1/FC10/FC177:

Der Aufruf des Funktionsbausteines FC177 für die Kommunikation mit dem Frequenzumrichter erfolgt im FC10, der wiederum im OB1 aufgerufen wird.

Das Projekt ist lauffähig und kann für Tests der Kommunikation benutzt werden.

Der Baustein „EMO_DP_READ5“ (FC177) steuert die gesamte Kommunikation mit dem Frequenzumrichter am Profibus.

Die Eingangsvariable „**I_O_Addr**“ ist die Adresse aus der Hardwarekonfiguration (im Beispiel ist es die „8“). Sie wird im Format „LONG“ eingetragen, also in der Form „L#8“.

Mit der Variablen „**VFX_Start**“ kann der Frequenzumrichter-Antrieb gestartet werden. Gleichzeitig muss auch eine der beiden Variablen für die Drehrichtung „**VFX_Left**“ oder „**VFX_Right**“ gesetzt sein (die Drehrichtung kann auch permanent gesetzt bleiben).

Geht der Frequenzumrichter in einen Fehlerzustand, so kann er mit der Variablen „**VFX_Reset**“ quittiert werden, sofern der Fehler quittierbar ist.

Mit der Variablen „**VFX_Refer**“ kann die Drehzahl bzw. die Frequenz vorgegeben werden. Der Datenbereich dieses Wertes geht von 0 bis 27648. Dabei entspricht 27648 dem Maximalwert, der im Gerät selbst im Parameter 343 eingestellt ist.

In der Version L2DP-M09.ZIP gibt es die zusätzliche Eingangsvariable „**VFX_Drehz_mode**“ zur Auswahl des Antriebsmodus bei den Geräten der „VFX“-Baureihe. Wird diese Variable auf „1“ gesetzt, so wird im Gerät der Antriebsmodus „Drehzahl“ aktiviert (feldorientierte Regelung). Ist die Variable gleich „0“, so arbeitet das Gerät im V/Hz-Modus (die Baureihe „FDU“ kennt nur den V/Hz-Modus).

In der Version L2DP-M09.ZIP gibt es die zusätzliche Eingangsvariable „**relay_set**“ zum Schalten der Relais 2 und 3 des Frequenzumrichters. Die Variable wirkt auf folgende Weise (die Werte sind dezimal):

„relay_set“ = 0	-	Keine Aktion
„relay_set“ = 20	-	Relais 2 ausschalten
„relay_set“ = 21	-	Relais 2 einschalten
„relay_set“ = 30	-	Relais 3 ausschalten
„relay_set“ = 31	-	Relais 3 einschalten

Die Variable sollte einmalig gesetzt werden, nach erfolgter Aktion wird die Variable auf Null zurückgesetzt. Die Funktion der Relais im Frequenzumrichter wird deaktiviert („AUS“).

Die Variable „**statcount**“ ist ein Zähler zur Kontrolle der internen Funktion des Bausteins.

„**Drehzahl**“, „**Moment**“ (in % des Motornennmoments), „**Strom**“, „**Leistung**“ und „**Temp**“ (Kühlkörpertemperatur) sind die Werte aus dem Frequenzumrichter. Bei der Version aus dem Projekt ab L2DP-M08.ZIP werden zusätzlich die 8 Digitaleingänge des Frequenzumrichters gelesen („**Inputs**“)

Die Ausgangsvariablen „**VFX_oper**“, „**VFX_L**“, „**VFX_R**“ zeigen an, ob der Antrieb gestartet wurde, und in welcher Richtung er läuft. Dabei kann es sein, dass die Laufrichtung auch noch im Stillstand angezeigt wird, deshalb darf nur die Variable „**VFX_oper**“ benutzt werden, um den eingeschalteten Zustand des Antriebs zu überwachen.

Die Variable „**VFX_Rel_1**“ gibt den Zustand des Relais 1 des Frequenzumrichters wieder. Dieses Relais ist werksseitig auf die Funktion „Fehler“ eingestellt, damit wird dieses Ausgangsbit im Fehlerfall aktiv.

Die Einstellung des Relais 1 kann im Parameter 551 des Frequenzumrichters geprüft oder geändert werden.

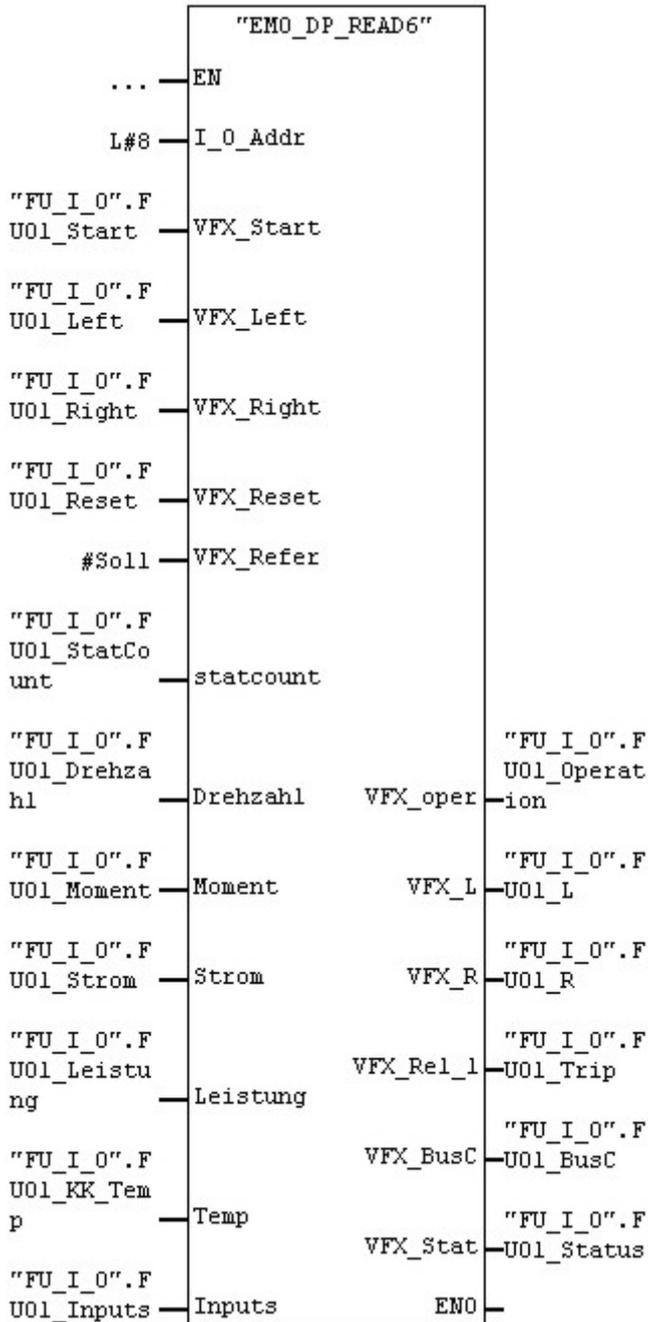
Die Variable „**VFX_BusC**“ zeigt an, ob am Frequenzumrichter die Steuerung über den Profibus aktiviert ist oder nicht. Bspw. gibt es die Möglichkeit, über einen elektrischen Eingang des Gerätes von Bussteuerung auf Klemmensteuerung umzuschalten (Vor-Ort-Steuerung), deshalb ist es sehr sinnvoll, diese Variable auszuwerten.

Das Byte „**VFX_Stat**“ liefert in codierter Form Informationen über Fehler oder Warnungen des Frequenzumrichters.

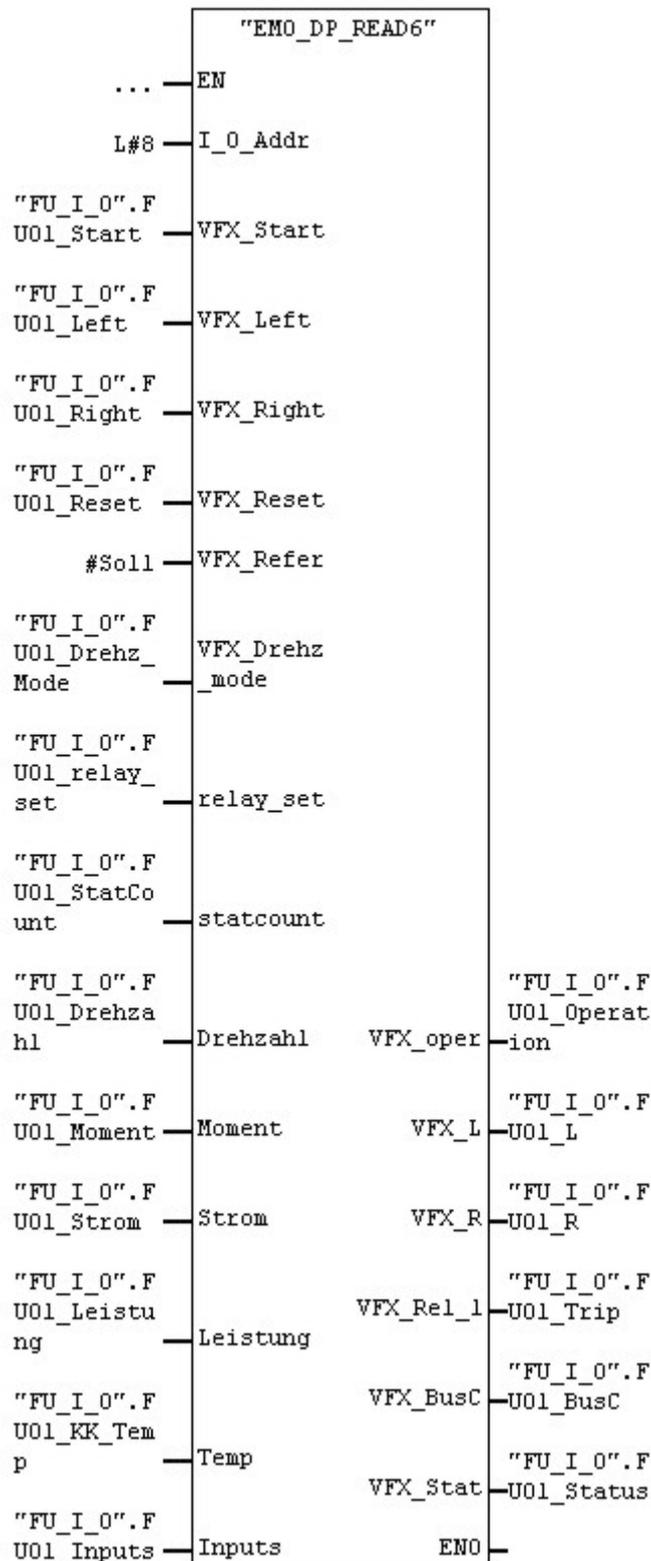
Fehlercode für „**VFX_Stat**“ bzw. Menü 722:

0=kein Fehler	16=Übertemperatur (vom Powerboard)
1=I ² t Motorschutz	17=Überstrom F
2=PTC Motorschutz	18=Überspg. Bremsrampe
3=Motoranschluss	19=Überspg. konst. Bremsen
4=Rotor blockiert	20=Überspg. im Betrieb
5=Fehler ext. Baugruppe	21=Überdrehzahl
6=Lastwächter Max.	22=Unterspannung
7=Lastwächter Min.	23=Fehler Leistungsteil
8=Kommunikation	24=Desat Fehler (Ausg.-Kurzschluss)
9=PT100 Fehler	25=DC-Link Fehler (bei parall. Modulen)
10=Kranhubwerkskontrolle	26=Interner Fehler
11=Pumpenfunktion	27=Überspannungsgrenze
12=ext. Motortemp.	28=Warnung Überspannung
13=Error 13	29=Error 29
14=Error 14	30=Error 30
15=Optionsbaugruppe	31=Error 31

Baustein aus dem Projekt L2DP-M08.ZIP
 (mit der zusätzlichen Variablen „Inputs“, die die 8 Digitaleingänge des Frequenzumrichters ließt):



Baustein aus dem Projekt L2DP-M09.ZIP
 (mit der zusätzlichen Variablen „VFX_Drehz_mode“ zur Auswahl des Antriebsmodus bei den Geräten der „VFX“-Baureihe und der Variablen „relay_set“ zum Schalten der Relais):



VAT2:

Mit der Variablen-tabelle VAT2 („Test-FU“) kann das Beispielprogramm getestet werden. Es sind alle beschriebenen Funktionen der Steuerung und des Datenaustauschs damit möglich.

1.4. Betrieb mehrerer Frequenzumrichter am Bus

Das Beispielprojekt ist für den Betrieb eines Gerätes am Bus ausgelegt. Der Funktionsbaustein FC177 kann für jedes weitere Gerät ebenfalls aufgerufen werden. Die insgesamt mögliche Zahl von Geräten am Bus ist in erster Linie durch die Größe des Speichers und die Leistungsfähigkeit der S7-CPU begrenzt, das ist bei der Projektierung zu beachten.

2. Version zur Steuerung und zum auswählbaren Lesen von Daten aus dem Frequenzumrichter

In dieser Version können 5 Parameter zum Lesen aus dem Frequenzumrichter in die S7 nach Bedarf ausgewählt werden. Es wird die DP-V1 Funktionalität genutzt.

2.1. Simatic S7 Beispielprojekt

Emotron stellt ein archiviertes S7-Beispielprojekt (L2dpv1_K.zip) zur Verfügung, in dem alle notwendigen Bausteine für die Profibus-Kommunikation mit den Frequenzumrichtern enthalten sind.

Folgende Bausteine sind im Beispielprojekt enthalten:

OB1	Beispiel für den Aufruf des Funktionsbausteins
FC177	Funktion für die Kommunikation mit den Emotron-Frequenzumrichtern
SFB52	Systemfunktion für das Lesen (benutzt in FC177)
DB250	Instanz-DB zu SFB52
DB252	Kopie von DB250 (für einen zweiten Frequenzumrichter)
VAT1	Variablentabelle für Testzwecke (das Projekt ist lauffähig).

2.2. Konfiguration im S7-Projekt für einen Emotron-Frequenzumrichter

Als Erstes ist in der Hardware-Konfiguration die gsd-Datei Version 2.0 einzubinden.

Nach dem Laden und Aktualisieren des Hardware-Katalogs lässt sich der Frequenzumrichter folgendermaßen im Verzeichnisbaum finden:

PROFIBUS-DP – Weitere Feldgeräte – Allgemein – **Anybus-CC PROFIBUS DP V1**

Dieses Gerät ist am Bus zu installieren, dann sind von diesem Gerät viermal „Input 1 Byte“ und viermal „Output 1 Byte“ einzufügen. (siehe Bild unten)

Die Adressen für die Eingangs- und Ausgangsperipherie sind einzutragen.

Es ist wichtig, wie im Bild zu sehen, dass die **gleiche Basisadresse** für Ein- und Ausgang verwendet wird (im Beispiel auf dem Bild die Adresse „8“).

Diese Adresse wird später auch für die Konfiguration der Kommunikation benötigt.

Die Adresse des Frequenzumrichters am Profibus (im Beispiel „10“ bzw. „12“) ist am Gerät selbst im Parameter **2631** einzutragen.

Bei dieser Variante ist der Parameter **2632** (Datengröße) auf „4“ zu stellen.

Achtung ! – Wenn die Profibus-Adresse (Parameter 2631) oder die Datengröße (Parameter 2632) im Gerät geändert wird, so muss die Versorgungsspannung ausgeschaltet werden, das Gerät muss völlig inaktiv sein, und dann kann die Versorgungsspannung wieder eingeschaltet werden, erst dadurch wird der geänderte Wert wirksam !

Steckplatz	DP-Kennung	Bestellnummer / Bezeichnung	E-Ad...	A...	K...
1	144	Input 1 byte	8		
2	144	Input 1 byte	9		
3	144	Input 1 byte	10		
4	144	Input 1 byte	11		
5	160	Output 1 byte		8	
6	160	Output 1 byte		9	
7	160	Output 1 byte		10	
8	160	Output 1 byte		11	
9					
10					

2.3. Arbeitsweise der Bausteine im Beispielprojekt

Alle Variablen des Typs „Merker“ (Bit, Byte, Word) sind nur als Beispiel verwendet und sollten natürlich durch konkrete Variablen des Projekts ersetzt werden.

OB1/FC177:

Der Aufruf der Funktionsbausteine für die Kommunikation mit dem Frequenzumrichter ist der Einfachheit halber gleich im Baustein OB1 dargestellt (in der Praxis wird der Aufruf sicherlich nicht direkt im OB1 erfolgen).

Das Projekt ist lauffähig und kann für Tests der Kommunikation benutzt werden.

Der Baustein „EMO_DPV1_READ5“ (FC177) steuert die gesamte Kommunikation mit dem Frequenzumrichter am Profibus.

Die Eingangsvariable „**I_O_Addr**“ ist die Adresse aus der Hardwarekonfiguration (im Beispiel ist es die „8“). Sie wird im Format „LONG“ eingetragen, also in der Form „L#8“.

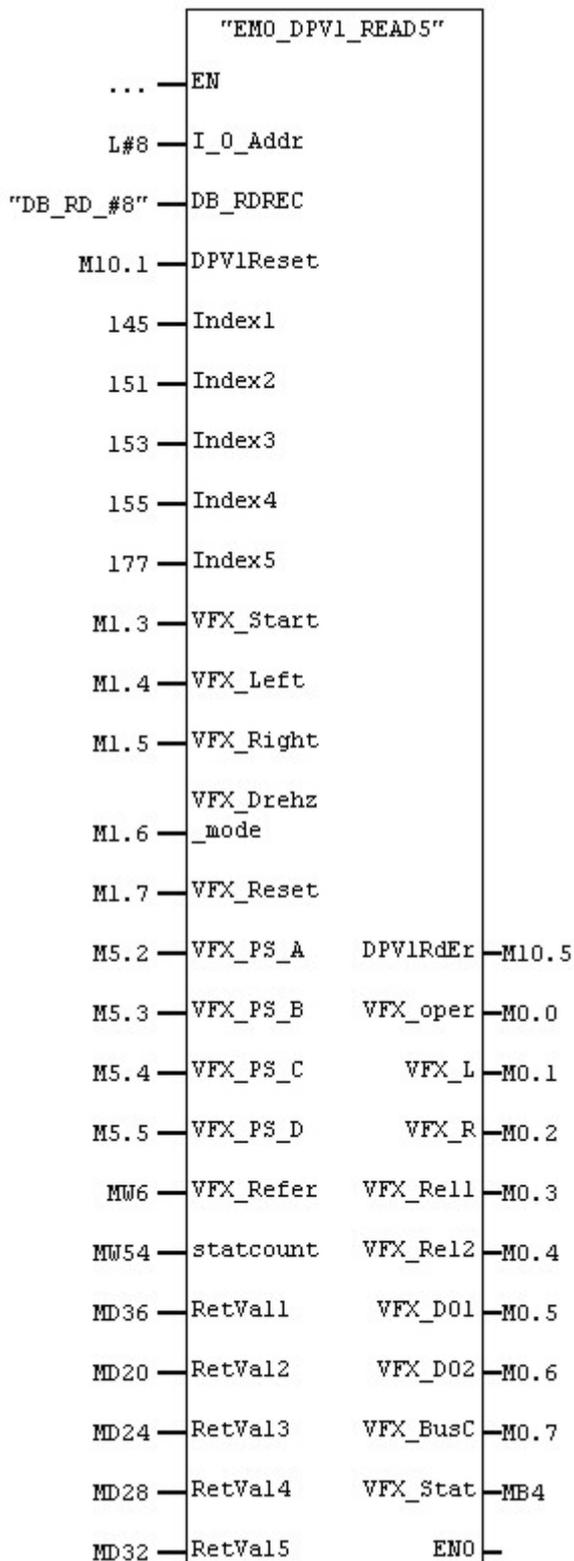
Für die DPV1-Kommunikation über den Profibus wird im FC177 die Systemfunktion SFB52 (RDREC) benutzt, die einen Instanz-Datenbaustein benötigt. Die Nummer dieses Instanz-DBs kann in der Variablen „**DB_RDREC**“ angegeben werden.

Das Format ist in diesem Fall die komplette Bezeichnung des Datenbausteins „DB250“ bzw. die entspr. symbolische Bezeichnung.

Sollte es zu einem Lesefehler kommen, so wird die Ausgangsvariable „**DPV1RdEr**“ gesetzt. In diesem Fall liegt entweder ein Fehler in der Konfiguration vor oder es sind falsche Index-Nummern verwendet worden.

Nach einem Lesefehler erfolgt kein weiteres Lesen.

Mit dem Setzen der Variablen „**DPV1Reset**“ können Lesefehler quittiert werden. Zuvor sollte jedoch die Ursache geklärt sein (Korrektur der Index-Werte).



In der Version L2DPV1-K01.ZIP gibt es die zusätzliche Eingangsvariable „**VFX_Drehz_mode**“ zur Auswahl des Antriebsmodus bei den Geräten der „VFX“-Baureihe. Wird diese Variable auf „1“ gesetzt, so wird im Gerät der Antriebsmodus „Drehzahl“ aktiviert (feldorientierte Regelung). Ist die Variable gleich „0“, so arbeitet das Gerät im V/Hz-Modus (die Baureihe „FDU“ kennt nur den V/Hz-Modus).

Mit den Eingangsvariablen „**Index1**“ ... „**Index5**“ werden die 5 Werte ausgewählt, die aus dem Frequenzumrichter gelesen werden. Sie können direkt als Zahl eingetragen werden, oder auch als Variable.

Achtung ! – Es dürfen nur gültige Index-Werte verwendet werden (siehe Tabelle unten).

Mit der Variablen „**VFX_Start**“ kann der Frequenzumrichter-Antrieb gestartet werden. Gleichzeitig muss auch eine der beiden Variablen für die Drehrichtung „**VFX_Left**“ oder „**VFX_Right**“ gesetzt sein.

Geht der Frequenzumrichter in einen Fehlerzustand, so kann er mit der Variablen „**VFX_Reset**“ quittiert werden, sofern der Fehler quittierbar ist.

Für die Auswahl eines der vier Parametersätze im Frequenzumrichter stehen vier Variable bereit – „**VFX_Par_Set_A**“ bis „**VFX_Par_Set_D**“.

Mit der Variablen „**VFX_Refer**“ kann die Drehzahl bzw. die Frequenz vorgegeben werden. Der Datenbereich dieses Wertes geht von 0 bis 1023. Dabei entspricht 1023 dem Maximalwert, der im Gerät selbst im Parameter 343 eingestellt ist.

Die Variable „**statcount**“ ist ein Zähler zur Kontrolle der internen Funktion des Bausteins.

Mit den 5 Variablen „**RetVal1**“ ... „**RetVal5**“ werden die 5 Werte aus dem Frequenzumrichter ausgegeben, die mit den Index-Werten ausgewählt worden sind. Einige Werte sind vorzeichenbehaftet (Zweierkomplement-Darstellung). Falls sie als Absolutwerte benötigt werden, können sie mit den Funktionen FC198 „Abs_Wert16“ bzw. FC199 „Abs_Wert32“ umgewandelt werden.

Die Ausgangsvariablen „**VFX_oper**“, „**VFX_L**“, „**VFX_R**“ zeigen an, ob der Antrieb gestartet wurde, und in welcher Richtung er läuft. Dabei kann es sein, dass die Laufrichtung auch noch im Stillstand angezeigt wird, deshalb darf nur die Variable „**VFX_oper**“ benutzt werden, um den eingeschalteten Zustand des Antriebs zu überwachen.

Weiterhin sind die Zustände der Relais und Digitalausgänge des Frequenzumrichters verfügbar: „**VFX_Rel1**“, „**VFX_Rel2**“, „**VFX_DO1**“, „**VFX_DO2**“. Ihre Funktion wird am Gerät selbst programmiert.

Die Variable „**VFX_BusC**“ zeigt an, ob am Frequenzumrichter die Steuerung über den Profibus aktiviert ist oder nicht. Bspw. gibt es die Möglichkeit, über einen elektrischen Eingang des Gerätes von Bussteuerung auf Klemmensteuerung umzuschalten (Vor-Ort-Steuerung), deshalb ist es sehr sinnvoll, diese Variable auszuwerten.

Das Byte „**VFX_Stat**“ liefert in codierter Form Informationen über Fehler oder Warnungen des Frequenzumrichters.

Eine einfache binäre Fehlermeldung ist aber auch über ein Relais-Signal oder einen Digitalausgang als Bussignal verfügbar, wenn sie am Gerät entsprechend programmiert sind.

Fehlercode für „**VFX_Stat**“ bzw. Menü 722 / Index 160:

0=kein Fehler	16=Übertemperatur (vom Powerboard)
1=I ² t Motorschutz	17=Überstrom F
2=PTC Motorschutz	18=Überspg. Bremsrampe
3=Motoranschluss	19=Überspg. konst. Bremsen
4=Rotor blickiert	20=Überspg. im Betrieb
5=Fehler ext. Baugruppe	21=Überdrehzahl
6=Lastwächter Max.	22=Unterspannung
7=Lastwächter Min.	23=Fehler Leistungsteil
8=Kommunikation	24=Desat Fehler
9=PT100 Fehler	25=DC-Link Fehler
10=ext. Abweichkontrolle	26=Interner Fehler
11=Pumpenfunktion	27=Überspannungsgrenze
12=ext. Motortemp.	28=Warnung Überspannung
13=Error 13	29=Error 29
14=Error 14	30=Error 30
15=Optionsbaugruppe	31=Error 31

VAT1:

Mit der Variablen-tabelle VAT1 kann das Beispielprogramm getestet werden. Es sind alle grundsätzlichen Funktionen der Steuerung und des Datenaustauschs damit möglich.

Folgende Parameter des Frequenzumrichters können mit dem Funktionsbaustein FC177 durch die Auswahl der entspr. **Index**-Werte gelesen werden:

Menü-Nr.	Bedeutung	Index	Beispiel
711	Prozesswert (entspr. Menü 321)	145	1490 = 1490 U/min
712	Drehzahl	146	1490 = 1490 U/min
713	Drehmoment Nm	147	36 = 3,6 Nm
713	Drehmoment %	148	24 = 24% Mot.Nennm.
714	Wellenleistung	149	4650 = 4,650 kW
715	Elektrische Leistung	150	4820 = 4,820 kW
716	Strom	151	107 = 10,7 A
717	Ausgangsspannung	152	3920 = 392,0 V
718	Frequenz	153	498 = 49,8 Hz
719	DC-Spannung	154	5275 = 527,5 V
71A	Kühlkörpertemperatur	155	377 = 37,7°C
71B	ext. PT100 1 (wenn vorh.)	156	377 = 37,7°C
71B	ext. PT100 2 (wenn vorh.)	157	377 = 37,7°C
71B	ext. PT100 3 (wenn vorh.)	158	377 = 37,7°C
721	interner Status des Freq.Umr.	159	Emotron-Code
722	Fehler / Warnungen	160	s. oben
723	Digitaleingänge	161	Byte0 DI 87654321
724	Digitalausgänge	162	Byte1 Rel 321 Byte0 DO 21
725	Analogeingang 1	163	25 = 25%
725	Analogeingang 2	164	25 = 25%
726	Analogeingang 3	165	25 = 25%
726	Analogeingang 4	166	25 = 25%
727	Analogausgang 1	167	25 = 25%
727	Analogausgang 2	168	25 = 25%
728	Status I/O Board 1 (wenn vorh.)	169	Emotron-Code
729	Status I/O Board 2 (wenn vorh.)	170	Emotron-Code
72A	Status I/O Board 3 (wenn vorh.)	171	Emotron-Code
731	Run Zeit Stunden	172	114 = 114 Stunden
731	Run Zeit Minuten	173	45 = 45 Minuten
731	Run Zeit Sekunden	174	18 = 18 Sekunden
732	Netz Zeit Stunden	175	285 = 285 Stunden
732	Netz Zeit Minuten	176	31 = 31 Minuten
732	Netz Zeit Sekunden	177	20 = 20 Sekunden
733	Energie	178	32850 = 32,85 kW
620	Logic Y	179	Emotron-Code
630	Logic Z	180	Emotron-Code
921	Freq.Umr. Typ	181	Emotron-Code
922	Software	182	Emotron-Code

2.4. Betrieb mehrerer Frequenzumrichter am Bus

Das Beispielprojekt ist für den Betrieb von zwei Geräten am Bus ausgelegt. Der Funktionsbaustein FC177 kann für jedes Gerät aufgerufen werden. Lediglich der Instanz-Datenbaustein (DB250) muss jeweils in einer separaten Kopie für jedes Gerät benutzt werden (z.B. DB252). Die insgesamt mögliche Zahl von Geräten am Bus ist in erster Linie durch die Größe des Speichers und die Leistungsfähigkeit der S7-CPU begrenzt, das ist bei der Projektierung zu beachten.

3. Version zum Steuern sowie zum Lesen und Schreiben aller Daten

In dieser Version können alle Parameter des Frequenzumrichters zum Lesen in die S7 oder zum Schreiben in den Umrichter nach Bedarf ausgewählt werden. Die DP-V1 Funktionalität wird zum Lesen und Schreiben der Parameter genutzt.

3.1. Simatic S7 Beispielprojekt

Emotron stellt ein archiviertes S7-Beispielprojekt zur Verfügung (L2dpv1_E.zip), in dem alle notwendigen Bausteine für die Profibus-Kommunikation mit den Frequenzumrichtern enthalten sind.

Folgende Bausteine sind im Beispielprojekt enthalten:

OB1	Beispiel für den Aufruf der Funktionsbausteine
FC137	Funktion für die Kommunikation mit den Emotron-Frequenzumrichtern
SFB52	Systemfunktion für das Lesen (benutzt in FC137)
DB250	Instanz-DB zu SFB52
DB252	Kopie von DB250
SFB53	Systemfunktion für das Schreiben (benutzt in FC137)
DB251	Instanz-DB zu SFB53
DB253	Kopie von DB253
DB152	Emotron-Datenbaustein für die Kommunikation und die Parameterdaten.
DB153	Kopie von DB152, für die Kommunikation mit einem 2. Frequenzumrichter.
VAT1	Variablentabelle für Testzwecke

3.2. Konfiguration im S7-Projekt für einen Emotron-Frequenzumrichter

Die Konfiguration in dieser Version ist identisch mit der „Nur Lesen“ - Version (s. 2.2).

3.3. Arbeitsweise der Bausteine im Beispielprojekt

Alle Variablen des Typs „Merker“ (Bit, Byte, Word) sind nur als Beispiel verwendet und sollten natürlich durch konkrete Variablen des Projekts ersetzt werden.

OB1/FC137:

Der Aufruf der Funktionsbausteine für die Kommunikation mit dem Frequenzumrichter ist der Einfachheit halber gleich im Baustein OB1 dargestellt. Das Projekt ist lauffähig und kann für Tests der Kommunikation benutzt werden.

Der Baustein „EMO_DP1_COMM“ (FC137) steuert die gesamte Kommunikation mit dem Frequenzumrichter am Profibus.

Die Eingangsvariable „**I_O_Address**“ ist die Adresse aus der Hardwarekonfiguration (im Beispiel ist es die „8“).

Sie wird im Format „LONG“ eingetragen, also in der Form „L#8“.

Mit „**DB_Number**“ wird die Nummer des zugehörigen Datenbausteins angegeben, in dem alle Parameter für diesen Antrieb abgelegt werden (z.B. DB152). Sie ist im „Integer“-Format einzugeben, also einfach als Zahl „152“.

Für die DPV1-Kommunikation über den Profibus werden im FC137 die beiden Systemfunktionen SFB52 (RDREC) und SFB53 (WRREC) benutzt, die beide einen Instanz-Datenbaustein benötigen. Die Nummer des jeweiligen Instanz-DBs kann in den Variablen „**DB_RDREC**“ und „**DB_WRREC**“ angegeben werden. Das Format ist in diesem Fall die komplette Bezeichnung des Datenbausteins „DB250“ bzw. „DB251“ oder die entspr. symbolischen Bezeichnungen.

Durch das Setzen der Variablen „**Emo_DPV1_READ**“ wird das periodische Lesen derjenigen Parameter aktiviert, die dafür im DB152 markiert sind. (Erklärung folgt beim DB152)

Die Variable kann permanent gesetzt bleiben, solange wie die Parameterdaten periodisch gelesen werden sollen.

Mit dem Setzen der Variablen „**Emo_DPV1_rd_All**“ wird das einmalige Lesen aller Parameter aktiviert, die im DB152 enthalten sind. Dieser Merker sollte nur einmal gesetzt werden.

Gleichzeitig muss auch „Emo_DPV1_READ“ gesetzt sein.

```

CALL "EMO_DPVI_COMM#1"                                FC137
  I_O_Address      :=L#8
  DB_Number        :=152
  DB_RDREC         :="DB_RD_#1"                       DB250
  DB_WRREC         :="DB_WR_#1"                       DB251
  Emo_DPVI_READ    :="Emo_DPVI_READ"                 M10.0
  Emo_DPVI_rd_All  :="Emo_DPVI_rd_All"               M10.2
  Emo_DPVI_WRITE   :="Emo_DPVI_WRITE"                M10.3
  Emo_DPVI_wr_All  :="Emo_DPVI_wr_All"               M10.4
  Emo_DPVI_Reset   :="Emo_DPVI_Reset"                M10.1
  VFX_Start        :="Emo_START"                     M1.3
  VFX_Left         :="Emo_LEFT_DIR"                  M1.4
  VFX_Right        :="Emo_RIGHT_DIR"                 M1.5
  VFX_Reset        :="Emo_RESET"                     M1.6
  VFX_Par_Set_A    :="Emo_PARAM_A"                   M5.2
  VFX_Par_Set_B    :="Emo_PARAM_B"                   M5.3
  VFX_Par_Set_C    :="Emo_PARAM_C"                   M5.4
  VFX_Par_Set_D    :="Emo_PARAM_D"                   M5.5
  VFX_Reference    :="Emo_SOLLWERT"                  MW6
  DPVI_busy        :="Emo_busy"                      M10.7
  DPVI_read_error  :="Emo_DPVI_rd_error"             M10.5
  DPVI_write_error:= "Emo_DPVI_wr_error"             M10.6
  VFX_operation    :="Emo_RUN_STATUS"                M0.0
  VFX_L            :="Emo_RUNS_LEFT"                 M0.1
  VFX_R            :="Emo_RUNS_RIGHT"                 M0.2
  VFX_Relais_1     :="Emo_RELAY_1"                   M0.3
  VFX_Relais_2     :="Emo_RELAY_2"                   M0.4
  VFX_DigOut_1     :="Emo_DIGOUT_1"                  M0.5
  VFX_DigOut_2     :="Emo_DIGOUT_2"                  M0.6
  VFX_BusControl   :="Emo_BUSSTEUERUNG"              M0.7
  VFX_Status       :="Emo_ERROR_BYTE"                MB4
  rpm              :="Emo_rpm"                       MW18
  Nm               :="Emo_Nm"                         MD20
  kW               :="Emo_kW"                         MD24
  Amp              :="Emo_Amp"                       MD28
  Temp             :="Emo_Temp"                      MD32

```

Sollte es zu einem Lesefehler kommen, so wird die Ausgangsvariable „**DPV1_read_error**“ gesetzt. In diesem Fall kann im DB152 der Fehler analysiert werden (siehe DB152).

Nach einem Lesefehler erfolgt kein weiteres Lesen.

Mit dem Setzen der Variablen „**Emo_DPVI_Reset**“ können Schreib- oder Lesefehler quittiert werden. Zuvor sollte jedoch die Ursache geklärt sein (z.B. Schreibversuch von Parametern bei laufendem Antrieb, die nur im Ruhezustand geschrieben werden dürfen.)

Diese Variable sollte nur einmalig gesetzt werden.

Durch das Setzen der Variablen „**Emo_DPVI_WRITE**“ wird das Schreiben derjenigen Parameter aktiviert, die dafür im DB152 markiert sind. (Erklärung folgt beim DB152) Diese Variable sollte immer dann einmalig gesetzt werden, wenn geänderte Daten geschrieben werden sollen.

Mit dem Setzen der Variablen „**Emo_DPVI_wr_All**“ wird das einmalige Schreiben aller Parameter aktiviert, die im DB152 enthalten sind. Diese Variable sollte nur einmal gesetzt werden.

Gleichzeitig mit „Emo_DPV1_wr_All“ muss auch „Emo_DPV1_WRITE“ gesetzt werden.
ACHTUNG: Die Daten werden im Frequenzumrichter sofort wirksam. Es muss sichergestellt sein, dass es sich um autorisierte Daten handelt, die z.B. vorher durch komplettes Lesen aus dem Gerät gewonnen wurden.

Sollte es zu einem Schreibfehler kommen, so wird die Ausgangsvariable „DPV1_write_error“ gesetzt.

In diesem Fall kann im DB152 der Fehler analysiert werden (siehe DB152).

Nach einem Schreibfehler erfolgt kein weiteres Schreiben.

Die Ausgangsvariable „DPV1_busy“ zeigt an, dass der Kommunikationsbaustein noch lesend oder schreibend aktiv ist.

Mit der Variablen „VFX_Start“ kann der Frequenzumrichter-Antrieb gestartet werden. Gleichzeitig muss auch eine der beiden Variablen für die Drehrichtung „VFX_Left“ oder „VFX_Right“ gesetzt sein.

Geht der Frequenzumrichter in einen Fehlerzustand, so kann er mit der Variablen „VFX_Reset“ quittiert werden, sofern der Fehler quittierbar ist.

Für die Auswahl eines der vier Parametersätze im Frequenzumrichter stehen vier Variable bereit – „VFX_Par_Set_A“ bis „VFX_Par_Set_D“.

Mit der Variablen „VFX_Reference“ kann die Drehzahl bzw. die Frequenz vorgegeben werden. Der Datenbereich dieses Wertes geht von 0 bis 1023. Dabei entspricht 1023 dem Maximalwert, der im Gerät selbst im Parameter 343 eingestellt ist.

Die Ausgangsvariablen „VFX_operation“, „VFX_L“, „VFX_R“ zeigen an, ob der Antrieb gestartet wurde, und in welcher Richtung er läuft.

Weiterhin sind die Zustände der Relais und Digitalausgänge des Frequenzumrichters verfügbar: „VFX_Relais_1“, „VFX_Relais_2“, „VFX_DigOut_1“, „VFX_DigOut_2“. Ihre Funktion wird am Gerät selbst programmiert.

Die Variable „VFX_BusControl“ zeigt an, ob am Frequenzumrichter die Steuerung über den Profibus aktiviert ist oder nicht. Bspw. gibt es die Möglichkeit, über einen elektrischen Eingang des Gerätes von Bussteuerung auf Klemmensteuerung umzuschalten.

Das Byte „VFX_Status“ liefert in codierter Form Informationen über Fehler oder Warnungen. Eine einfache binäre Fehlermeldung ist aber auch über ein Relais-Signal oder einen Digitalausgang als Bussignal verfügbar, wenn sie am Gerät entsprechend programmiert sind.

Die Variablen „rpm“, „Nm“, „kW“, „Amp“ und „Temp“ liefern auf einfache Weise die aktuellen Werte von Drehzahl, Drehmoment (in % Motornennmoment), Wellenleistung, Strom und Kühlkörpertemperatur. Sie sind, wie sehr viele andere Daten auch, ebenfalls im DB152 verfügbar. Diese Werte werden hier ohne Vorzeichen dargestellt. Im DB152 sind „rpm“, „Nm“, „kW“ auch mit Vorzeichen (Zweierkomplement-Darstellung) verfügbar.

DB152:

+120.0	Activ_71A	WORD	W#16#1	71A Heatsink Temp.*****
+122.0	Slot_ID_71A	DWORD	DW#16#0	
+126.0	Index_71A	INT	155	
+128.0	Data_71A	DWORD	DW#16#0	126h= 29,4°C
+132.0	Status_71A	DWORD	DW#16#0	
+136.0	Length_71A	INT	4	
+138.0	Fill_71A	WORD	W#16#0	
+140.0	Activ_723	WORD	W#16#1	723 Dig. Inputs *****
+142.0	Slot_ID_723	DWORD	DW#16#0	
+146.0	Index_723	INT	161	
+148.0	Data_723	DWORD	DW#16#0	4 = 00000100 (DigIn: 8 7 6 5 4 3 2 1)
+152.0	Status_723	DWORD	DW#16#0	
+156.0	Length_723	INT	2	
+158.0	Fill_723	WORD	W#16#0	
+160.0	Activ_725	WORD	W#16#1	725 Anal. Input 1 *****
+162.0	Slot_ID_725	DWORD	DW#16#0	
+166.0	Index_725	INT	163	
+168.0	Data_725	DWORD	DW#16#0	16h= 22%
+172.0	Status_725	DWORD	DW#16#0	
+176.0	Length_725	INT	4	
+178.0	Fill_725	WORD	W#16#0	
+180.0	Activ_725A	WORD	W#16#1	725A Anal. Input 2 *****
+182.0	Slot_ID_725A	DWORD	DW#16#0	
+186.0	Index_725A	INT	164	
+188.0	Data_725A	DWORD	DW#16#0	14h= 20%
+192.0	Status_725A	DWORD	DW#16#0	
+196.0	Length_725A	INT	4	
+198.0	Fill_725A	WORD	W#16#0	
+200.0	Activ_726	WORD	W#16#1	726 Anal. Input 3 *****
+202.0	Slot_ID_726	DWORD	DW#16#0	
+206.0	Index_726	INT	165	
+208.0	Data_726	DWORD	DW#16#0	14h= 20%
+212.0	Status_726	DWORD	DW#16#0	
+216.0	Length_726	INT	4	
+218.0	Fill_726	WORD	W#16#0	
+220.0	Activ_351	WORD	W#16#6	351 Max. Torque *****
+222.0	Slot_ID_351	DWORD	DW#16#8000	
+226.0	Index_351	INT	45	
+228.0	Data_351	DWORD	DW#16#6E	78h= 120%
+232.0	Status_351	DWORD	DW#16#0	
+236.0	Length_351	INT	4	
+238.0	Fill_351	WORD	W#16#0	
+240.0	Activ_5531	WORD	W#16#2	5531 ON / OFF Relais 3 *****
+242.0	Slot_ID_5531	DWORD	DW#16#8000	
+246.0	Index_5531	INT	179	
+248.0	Data_5531	DWORD	DW#16#0	1 = ON, 0 = OFF
+252.0	Status_5531	DWORD	DW#16#0	
+256.0	Length_5531	INT	2	
+258.0	Fill_5531	WORD	W#16#0	
+260.0	Activ_211	WORD	W#16#8	211 Language *****
+262.0	Slot_ID_211	DWORD	DW#16#8	

Dieser Datenbaustein ist bei Bedarf konfigurierbar und enthält alle Parameter und Werte, die zum und vom Frequenzumrichter übertragen werden.

Bei der Byte-Adresse 40 beginnen die eigentlichen Parameter-Daten.

(Davor liegen interne Systemdaten.)

Für jeden Parameter, der aus dem Frequenzumrichter gelesen, oder dorthin geschrieben werden soll, ist ein **Bereich von 20 Byte** vorhanden, der sich folgendermaßen zusammensetzt:

Activ_xxx – steuert bitweise die Lese- oder Schreibaktivitäten.

Die einzelnen Bits dieses Wortes können separat gesetzt werden:

Bit 0: Dieser Parameter wird periodisch gelesen („Emo_DPV1_READ“).
(dez. „1“)

Bit 1: Dieser Parameter wird mit „Emo_DPV1_WRITE“ geschrieben.
(dez. „2“)

Bit 2: An dieser Stelle im DB wird das periodische Lesen beendet.
(dez. „4“) (Auch wenn in nachfolgenden Werten noch Bit 0 markiert ist.)
(Der Parameter selbst wird nicht mehr gelesen.)
(Durch das Beenden und den erneuten Start wird der Zyklus schneller.)

Bit 3: An dieser Stelle im DB wird das Schreiben mit „Emo_DPV1_WRITE“
(dez. „8“) beendet.
(Auch wenn in nachfolgenden Werten noch Bit 1 markiert ist.)
(Der Parameter selbst wird nicht mehr geschrieben.)

Alle Parameter, bei denen in „Activ_xxx“ das Bit 0 gesetzt ist (dezimal „1“), werden also periodisch gelesen (siehe Bsp. oben - Parameter 71A, 723, 725, 725A, 726).

Im Beispiel wird das periodische Lesen in Zeile 220 beendet, weil mit der „6“ dort das Bit 2 gesetzt ist. Alle dahinter folgenden Parameter werden nur noch mit der Eingangsvariablen „Emo_DPV1_rd_All“ gelesen.

Es ist in Zeile 220 auch das Bit 1 gesetzt, d.h. der Parameter 351 (max. Drehmoment) kann periodisch zum Frequenzrichter geschrieben werden. Außerdem wird auch der Zustand von Relais 3 geschrieben (Parameter 5531, ab Zeile 240), das Relais kann also über den Bus geschaltet werden.

Zeile 260 enthält eine „8“, es ist Bit 3 gesetzt, damit wird das periodische Schreiben beendet.

Alle dahinter folgenden Parameter werden nur mit der Eingangsvariablen „Emo_DPV1_wr_All“ geschrieben.

Slot_ID_xxx – Zugriffskennung für den Parameter (vorprogrammiert).
(Bei Bedarf bitte in das Gerätehandbuch sehen.)

Index_xxx – Zugriffsindex für den Parameter (vorprogrammiert).
(Bei Bedarf bitte in das Gerätehandbuch sehen.)

Data_xxx – hier steht der gelesene oder zu schreibende Wert.
Darauf kann aus dem Anwenderprogramm zugegriffen werden.

Status_xxx – sollte es zu einem Schreib- oder Lesefehler kommen, dann steht hier ein Fehlercode. Bestimmte Parameter können nicht bei laufendem Antrieb geschrieben werden.
Der Status ist im Normalfall gleich Null.
Einige Fehlercodes (Byte 2 und 3):
80 B0 - ungültiger Index
80 B1 – “write length” error
80 B2 – ungültiger Slot
80 B6 – Zugriff gesperrt (evtl. nicht im RUN-Mode möglich)
80 B7 – außerhalb des Zahlenbereichs (beim Schreiben)
Andere Codes zeigen grundsätzliche Fehler des Datenbausteins.

Length_xxx – Länge der Daten (2 oder 4 Byte).

Fill_xxx – Reserve

Anpassen des DB152:

Der Datenbaustein DB152 (bzw. Kopien von ihm) kann nach Bedarf angepasst werden. Dabei ist einiges zu beachten:

- Der Adressbereich von 0 bis 36 ist Systembereich und darf nicht verändert werden.
- Die ersten fünf Parameter (Adressen 40, 60, 80, 100 und 120) werden direkt vom FC137 gelesen und sollten deshalb ebenfalls nicht verändert werden.
(Werden andere Parameter gelesen, so stimmen die Variablen-Namen am Baustein-Ausgang nicht mehr.)
- Der Bereich ab Adresse 140 kann (im Block von 20 Bytes) verändert werden.
- Wird ein Parameter entfernt oder hinzugefügt, so sind immer die kompletten 20 Byte zu löschen oder einzufügen:

+116.0	Length_716	INT	4	
+118.0	Fill_716	WORD	W#16#0	
+120.0	Activ_71A	WORD	W#16#1	71A Heatsink Temp.*****
+122.0	Slot_ID_71A	DWORD	DW#16#0	
+126.0	Index_71A	INT	155	
+128.0	Data_71A	DWORD	DW#16#0	126h= 29,4°C
+132.0	Status_71A	DWORD	DW#16#0	
+136.0	Length_71A	INT	4	
+138.0	Fill_71A	WORD	W#16#0	
+140.0	Activ_723	WORD	W#16#1	723 Dig. Inputs *****
+142.0	Slot_ID_723	DWORD	DW#16#0	
+146.0	Index_723	INT	161	
+148.0	Data_723	DWORD	DW#16#0	4 = 00000100 (DigIn: 8 7 6 5 4 3 2 1)
+152.0	Status_723	DWORD	DW#16#0	
+156.0	Length_723	INT	2	
+158.0	Fill_723	WORD	W#16#0	
+160.0	Activ_726	WORD	W#16#1	726 Anal. Input 3 *****
+162.0	Slot_ID_726	DWORD	DW#16#0	
+166.0	Index_726	INT	165	
+168.0	Data_726	DWORD	DW#16#0	14h= 20%
+172.0	Status_726	DWORD	DW#16#0	
+176.0	Length_726	INT	4	
+178.0	Fill_726	WORD	W#16#0	
+180.0	Activ_726A	WORD	W#16#1	726A Anal. Input 4 *****
+182.0	Slot_ID_726A	DWORD	DW#16#0	
+186.0	Index_726A	INT	166	

Die Gesamtlänge des Bausteins wird im FC137 automatisch erkannt.

- Die Parameter sollten so sortiert sein, dass zuerst alle kommen, die zum periodischen Lesen vorgesehen sind (in „Activ_xxx“ ist das Bit 0 gesetzt, dezimal „1“).
- Nach den zu Lesenden sollten die zu schreibenden Parameter kommen, falls geschrieben werden soll.
(Im Bsp.: „351 Max. Drehmoment“ und „5531 Ein/Aus Relais 3“)

+198.0	Fill_726A	WORD	W#16#0	
+200.0	Activ_351	WORD	W#16#6	351 Max. Torque *****
+202.0	Slot_ID_351	DWORD	DW#16#8000	
+206.0	Index_351	INT	45	
+208.0	Data_351	DWORD	DW#16#6E	78h= 120%
+212.0	Status_351	DWORD	DW#16#0	
+216.0	Length_351	INT	4	
+218.0	Fill_351	WORD	W#16#0	
+220.0	Activ_5531	WORD	W#16#2	5531 ON / OFF Relais 3 *****
+222.0	Slot_ID_5531	DWORD	DW#16#8000	
+226.0	Index_5531	INT	179	
+228.0	Data_5531	DWORD	DW#16#0	1 = ON, 0 = OFF
+232.0	Status_5531	DWORD	DW#16#0	
+236.0	Length_5531	INT	2	
+238.0	Fill_5531	WORD	W#16#0	
+240.0	Activ_211	WORD	W#16#8	211 Language *****
+242.0	Slot_ID_211	DWORD	DW#16#3	
+246.0	Index_211	INT	170	
+248.0	Data_211	DWORD	DW#16#0	0= english 3= deutsch (s. Manual)
+252.0	Status_211	DWORD	DW#16#0	

- Wenn der Baustein insgesamt sehr groß ist, so kann mit dem ersten zu schreibenden Parameter das periodische Lesen beendet werden, indem das Bit 2 von „Activ_xxx“ gesetzt wird. (Bei „Activ_351“ ist Bit 1 für „Schreiben“ gesetzt, sowie Bit 2 für „Lesen beenden“, das ergibt dezimal 2 + 4 = 6.)
- Der zu schreibende Parameterbereich sollte mit dem Setzen von Bit 3 in „Activ_xxx“ beendet werden. (Bei „Activ_211“ ist Bit 3 gesetzt, dezimal „8“.)
- Der Zugriff auf die Daten aus dem Anwender-Programm heraus kann folgendermaßen programmiert werden:

```
L DB152.DBD248 // lädt „Data_211“ – eingestellte Sprache
```

oder in symbolischer Schreibweise:

```
L „DB_DPV1_#1“.“Data_211“
```

(Die Variable „Data_xxx“ ist immer der eigentliche Parameter-Wert.)

- Alle Parameter tragen die Menü-Nummer, mit der sie im Gerät selbst erreichbar sind, und unter der auch die Beschreibung im Handbuch zu finden ist.

VAT1:

Mit der Variablen-tabelle VAT1 kann das Beispielpogramm getestet werden. Es sind alle grundsätzlichen Funktionen der Steuerung und des Datenaustauschs damit möglich.

3.4. Betrieb mehrerer Frequenzumrichter am Bus

Das Beispielprojekt ist für den Betrieb von zwei Geräten am Bus ausgelegt. Für jeden Frequenzumrichter sind ein Datenbaustein und die beiden Instanz-DBs für die Systemfunktionen notwendig. Sie können einfach durch Kopieren der Bausteine DB152 und DB250/DB251 erzeugt werden. Die insgesamt mögliche Zahl von Geräten am Bus ist in erster Linie durch die Größe des Speichers und die Leistungsfähigkeit der S7-CPU begrenzt, das ist bei der Projektierung zu beachten.