Technical description of the Profibus communication

between Simatic S7 PLC's and Emotron Variable Speed Drives

FDU/VFX 2.0

Overview of the different versions of the Profibus communication:

1. Control of the device and read out of speed, torque, current, mechanical power and heatsink temperature.
   (Without using DPV1 functionality)
   (Requires device firmware version 4.06 or higher)

   Simatic project: L2DP-M07.ZIP

   L2DP-M08.ZIP (additional reading of Digital Inputs)

   L2DP-M09.ZIP (additional selection of Drive Mode for VFX and control of relay contacts of the drives)

2. Control of the device and read out of five values which can be chosen out of a selection (With DPV1 functionality)

   Simatic project: L2DPV1-K.ZIP

   L2DPV1-K01.ZIP (additional selection of Drive Mode for VFX)

3. Control of the device and random access to all parameters of the device (read and write) (With DPV1 functionality)

   Simatic project: L2DPV1-E.ZIP

1. Version for control and reading of speed, torque, current, mechanical power and heatsink temperature to the PLC

With this version the device can be controlled and speed, torque, current, mechanical power and heatsink temperature can be read to the S7 PLC.
(With version L2DP-M08.ZIP also the Digital Inputs are available)

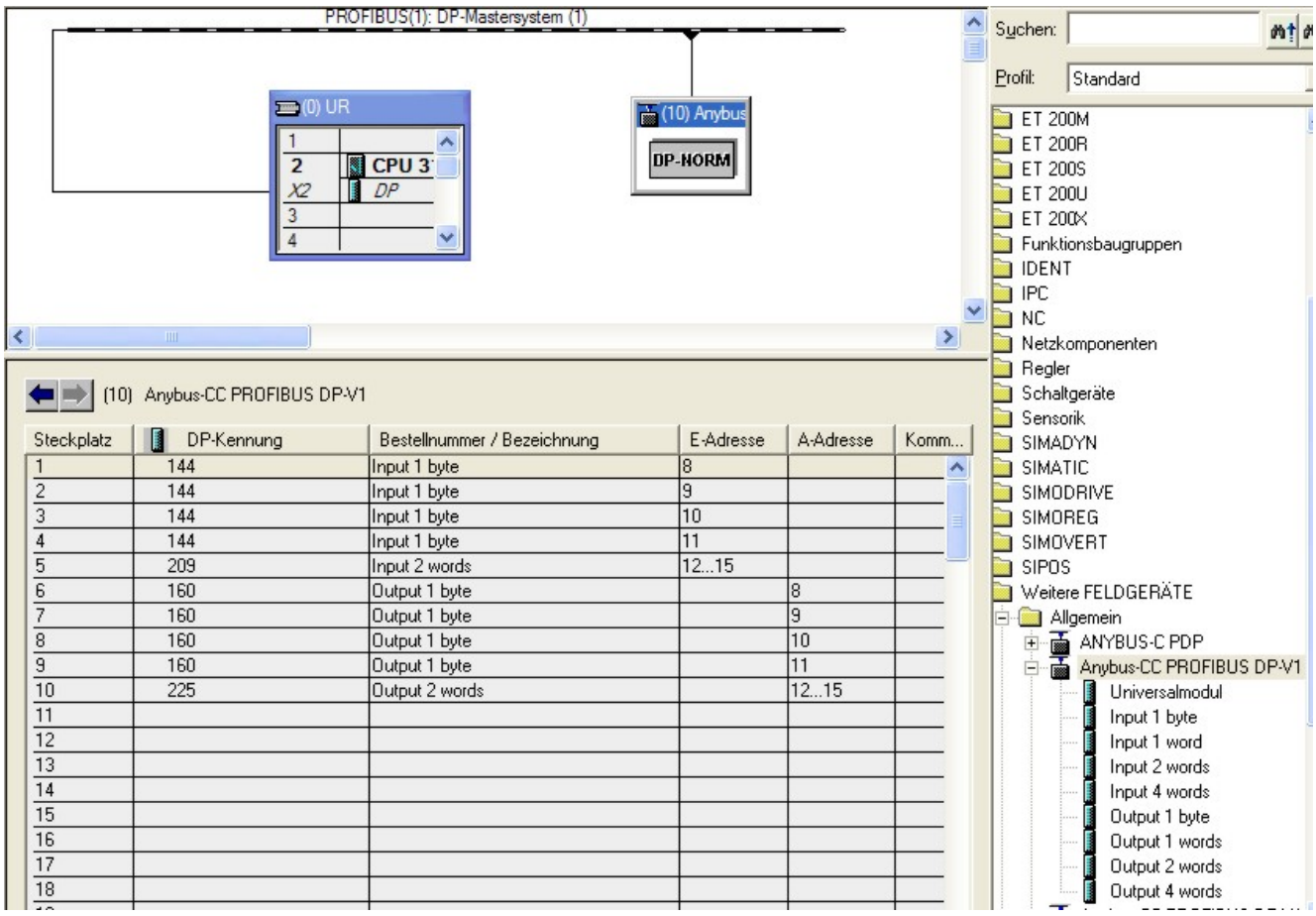
1.1. Simatic S7 example project

Emotron makes an archived example project (L2DPV1-M07.zip) available, which contains all necessary elements for the Profibus communication with the Emotron Variable Speed Drives.

The following blocks are part of the example project:

OB1	example for the function call
FC177	Function for the communication with the Emotron devices
FC10	calls the function FC177 and scales the reference value
VAT2	Table of variables for test purposes

1.2. Configuration of the S7 project with an Emotron Variable Speed Drive



At first you have to link the “gsd” file in the Hardware Configuration.

After loading the gsd file and actualizing the view, you can find the device as following:

PROFIBUS-DP – further field devices – General – **Anybus-CC PROFIBUS DP V1**

This device has to be installed at the field bus. Then you have to insert 4 times “Input 1 byte“, one time “Input 2 words”, 4 times “Output 1 byte“ and one time “Output 2 words”. (see picture above)

You have to fill in the address of the input and the output.
It is important to use **the same base address** for the inputs and outputs, as to be seen in the picture (address “8” in the example).

You will need this address later also for the configuration of the communication function block.

The fieldbus address of the device, which you chose (“10“ in the picture), can be set at the device itself in the menu **2631 “Address”**.

In this version the size of data (menu **2632**) has to be set to “8”.

CAUTION: If the Profibus address or data size in the VSD has been changed, this will only be effective after switching OFF and ON the power supply again!

1.3. Description of the function and data blocks

OB1/FC10/FC177:

To present the function call of communication with the Variable Speed Drive the function block FC177 is called in the function FC10, which itself is called for in the main block OB1. The Emotron project is runable and can be used for tests.

The function block „EMO_DP_READ5“ (FC177) controls the whole communication with the Variable Speed Drive which is connected to the Profibus network.

The input variable of the function FC177 **„I_O_Addr“** is the address, that was defined in the hardware configuration (in the example the “8”). The format is “LONG”, i.e. it has to be written as “L#8”.

With the flag **„VFX_Start“** you can set the device (the drive) into operation.
At the same time you have to set one of the flags **„VFX_Left“** or **„VFX_Right“** which defines the direction of rotation. (this flag for rotation direction can be set continuously)

If the fieldbus slave device itself switches to an error state you can reset it with the flag **„VFX_Reset“**.

With the input variable **„VFX_Refer“** the reference value can be set in a range from 0 to 27648. The value 27648 represents the maximum speed, which is set at the device itself in menu 343.

In the version L2DP-M09.ZIP also the input variable **„VFX_Drehz_mode“** (Drive mode) is available for use in combination with the VFX-series. When setting this flag to “1”, the drive mode “Speed” (direct torque control) is activated. When this flag is set to “0”, the drive is working in V/Hz mode (the FDU-series has only V/Hz mode available).

In the version L2DP-M09.ZIP the additional input variable **„Relay_set“** for controlling the Relay 2 and 3 of the drive is available. This function works as following (values are decimals):

„relay_set“= 0	-	No action
„relay_set“= 20	-	Relay 2 is de-energized
„relay_set“= 21	-	Relay 2 is energized
„relay_set“= 30	-	Relay 3 is de-energized
„relay_set“= 31	-	Relay 3 is energized

The variable **„statcount“** is only for observing of the internal work of the function block.

„Drehzahl“ (speed), **„Moment“** (torque in % of the nominal motor value) **„Strom“** (current), **„Leistung“** (mechanical power) and **„Temp“** (heatsink temperature) are the values which are read from the VSD.
With the version L2DP-M08.ZIP, additionally the 8 Digital Inputs are read (**„ Inputs“**).

The output flags of the function block **„VFX_oper“**, **„VFX_L“**, **„VFX_R“** show the actual operation status and the direction of rotation of the drive. Because the direction can be shown even when at standstill, only the flag **„VFX_oper“** can be used to determine the opteration status.

The variable **„VFX_Rel_1“** shows the status of Relay 1 of the drive. This Relay is default (factory setting) set on the function „Trip“, which will make the output bit at a fault situation active.
The function selection of Relay 1 can be checked or changed in parameter 551 of the drive.

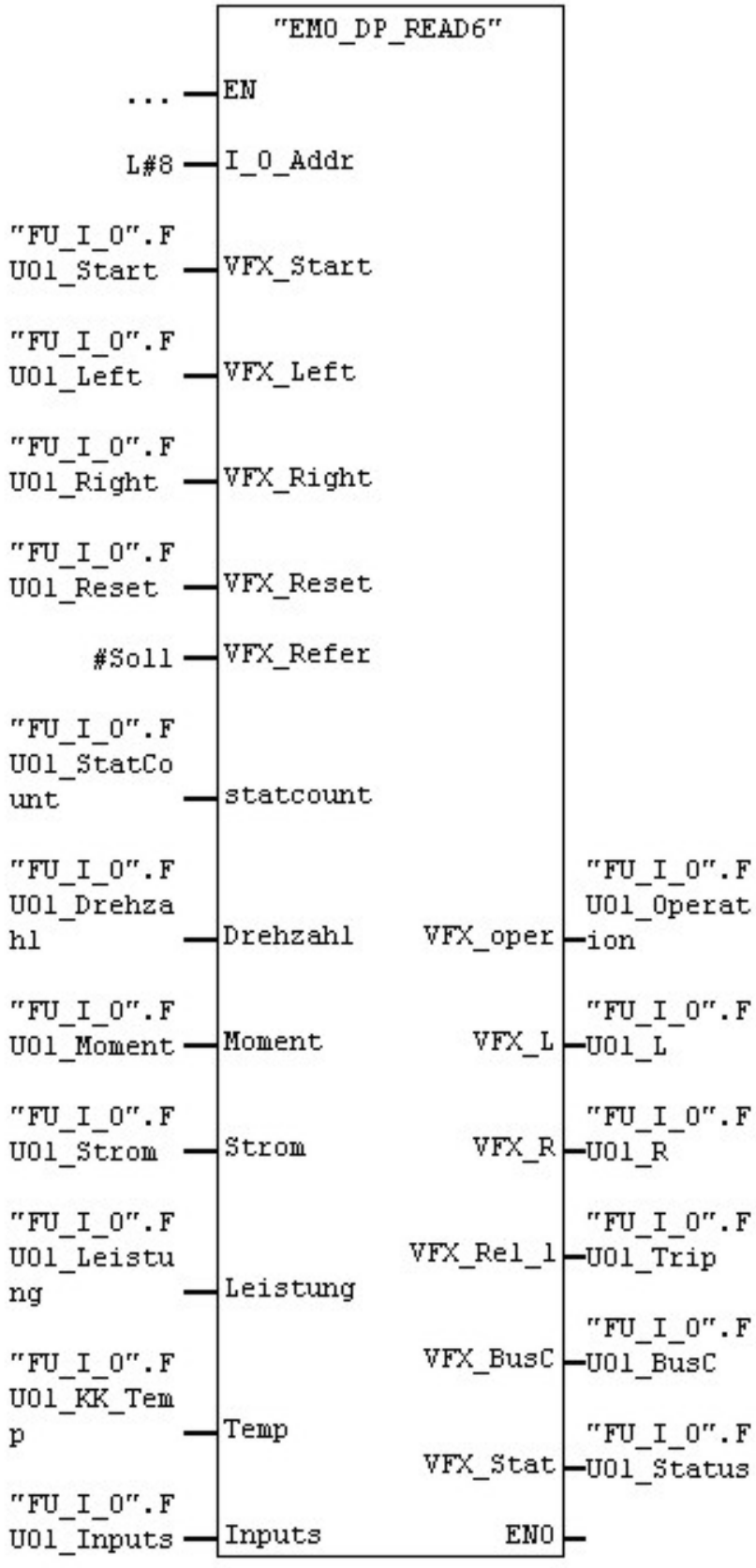
The output flag **„VFX_BusC“** shows if the device is controlled by the fieldbus communication or not. Because it is possible to switch between control via ‘communication’ and ‘remote’ via a Digital Input, it makes sense to monitor this flag to determine the control source.

The output variable **„VFX_Stat“** delivers a code of warnings or faults occurring in the Variable Speed Drive.

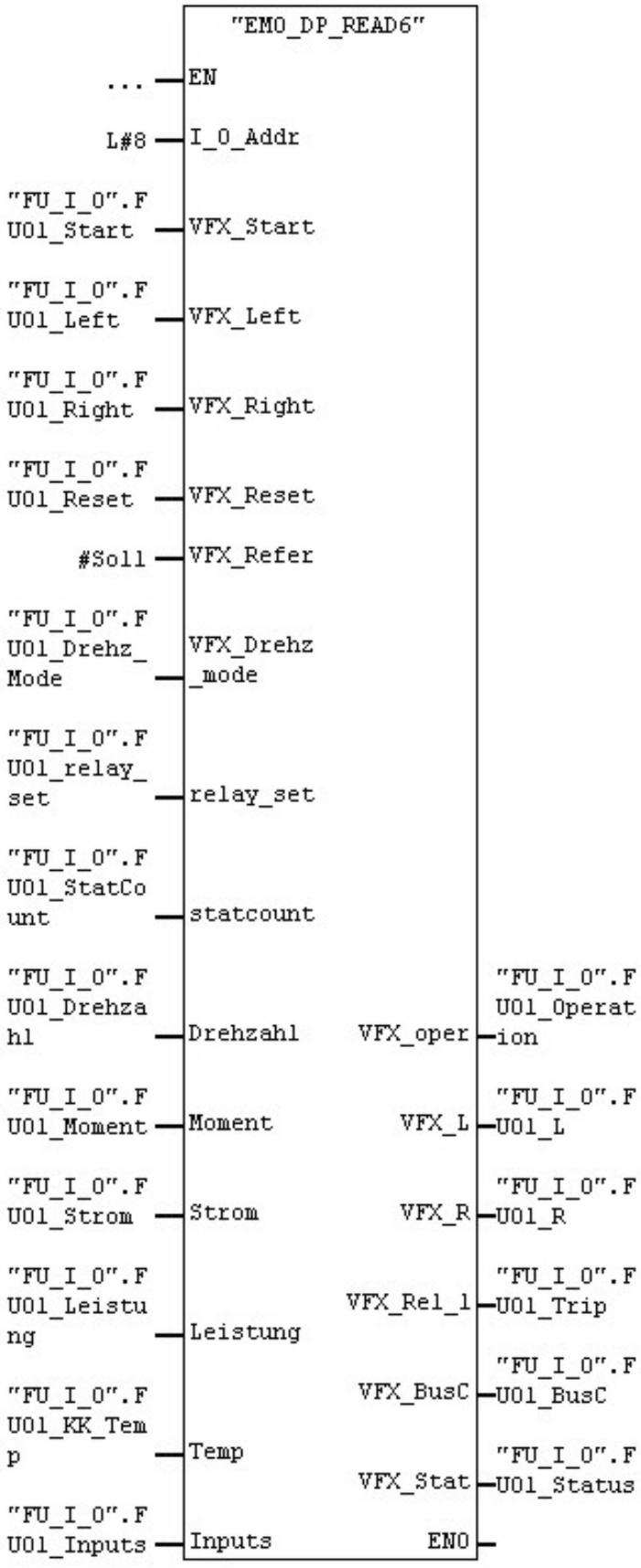
Existing error codes in **„VFX_Stat“** are:

0=No Error	16=Over temp from power board
1=I2t	17=Over curr F
2=PTC	18=Over volt D
3=Motor lost	19=Over volt G
4=Locked rotor	20=Over volt M
5=Ext trip	21=Over speed
6=Mon MaxAlarm	22=Under voltage
7=Mon MinAlarm	23=Power fault
8=Comm error	24=Desat Error
9=PT100 Trip	25=DC-Link Error
10=Deviation	26=Int error
11=Pump	27=Ovolt m cut
12=ext. Motortemp.	28=Over volt. Warning
13=Error 13	29=Error 29
14=Error 14	30=Error 30
15=Option	31=Error 31

Function and data block as used in the project L2DP-M08.ZIP
(with the additional variables „Inputs“, which reads the 8 digital inputs of the drive):



Function block as used in the project L2DP-M09.ZIP
(with the additional variable „VFX_Drehz_mode“ for selecting the drive mode with the VFX-series and the variable „relay_set“ for switching the relay’s):



VAT2:

With the table of variables VAT2 („Test-FU“) the example program can be tested. All described control functions and data exchange are possible to use.

1.4. Two or more devices connected to the field bus

The example project contains the function for the communication with one variable speed drive connected to the field bus. For every further Emotron device at the fieldbus the function FC177 can be called.
The maximal number of fieldbus devices depends on the performance and the memory size of the used Simatic S7 CPU.

2. Version for control and reading of selectable data to the PLC

With this version it is possible to read the actual values of 5 selectable parameters from the Variable Speed Drive to the Simatic S7 PLC. The DPV1 functionality is used to read the parameters.

2.1. Simatic S7 example project

Emotron makes an archived example project (L2dpv1_K.zip) available, which contains all necessary elements for the Profibus communication with the Emotron Variable Speed Drives.

The following blocks are part of the example project:

OB1 function call example

FC177 Function for the communication with the Emotron devices

SFB52 System function for reading (used in FC137)
DB250 Instance DB for SFB52
DB252 Copy of DB250 (used for a second unit)

VAT1 Table of variables for test purposes

2.2. Configuration of the S7 project with an Emotron Variable Speed Drive

At first you have to link the “gsd” file in the Hardware Configuration.

After loading the gsd file and actualizing the view, you can find the device as following:

PROFIBUS-DP – further field devices – General – **Anybus-CC PROFIBUS DP V1**

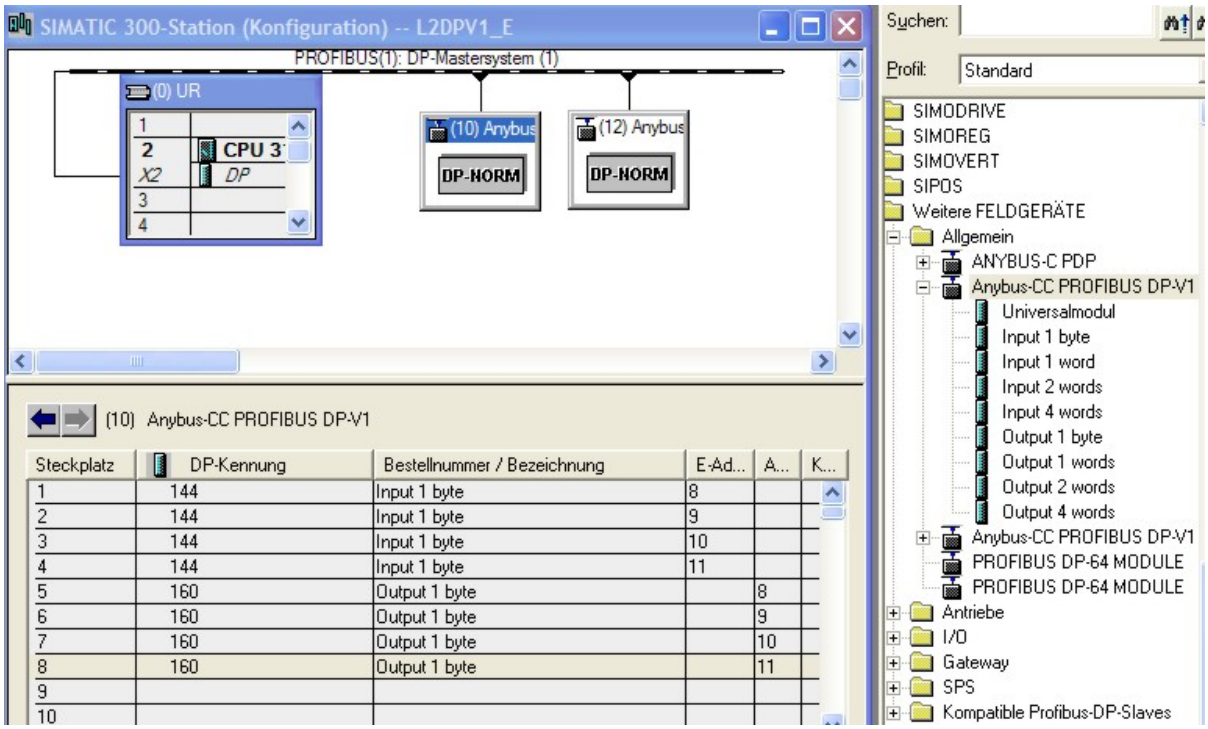
This device has to be installed at the field bus. Then you have to insert 4 times „Input 1 byte“ and 4 times „Output 1 byte“. (see picture below)

You have to fill in the address of the input and the output.
It is important to use **the same base address** for the inputs and outputs, as to be seen in the picture (address “8” at the example).
You will need this address later on also for the configuration of the communication function block.

The fieldbus address of the device, which you chose („10“ in the picture), can be set at the device itself in the menu **2631**.

In this version the data size (menu **2632**) has to be set to “4”.

CAUTION: If the Profibus address or the size of data setting in the VSD has been changed, this will only become effective after switching OFF and back ON again of the power supply!



2.3. Description of the function and data blocks

All variables of the type „Marker“ (Bit, Byte, Word) are only an example and should be replaced by real variables of your project.

OB1/FC177:

To present the function call for communication with the Variable Speed Drive in an easy way, it is written in the main organizational block OB1.
The Emotron project is runable and can be used for testing the communication.

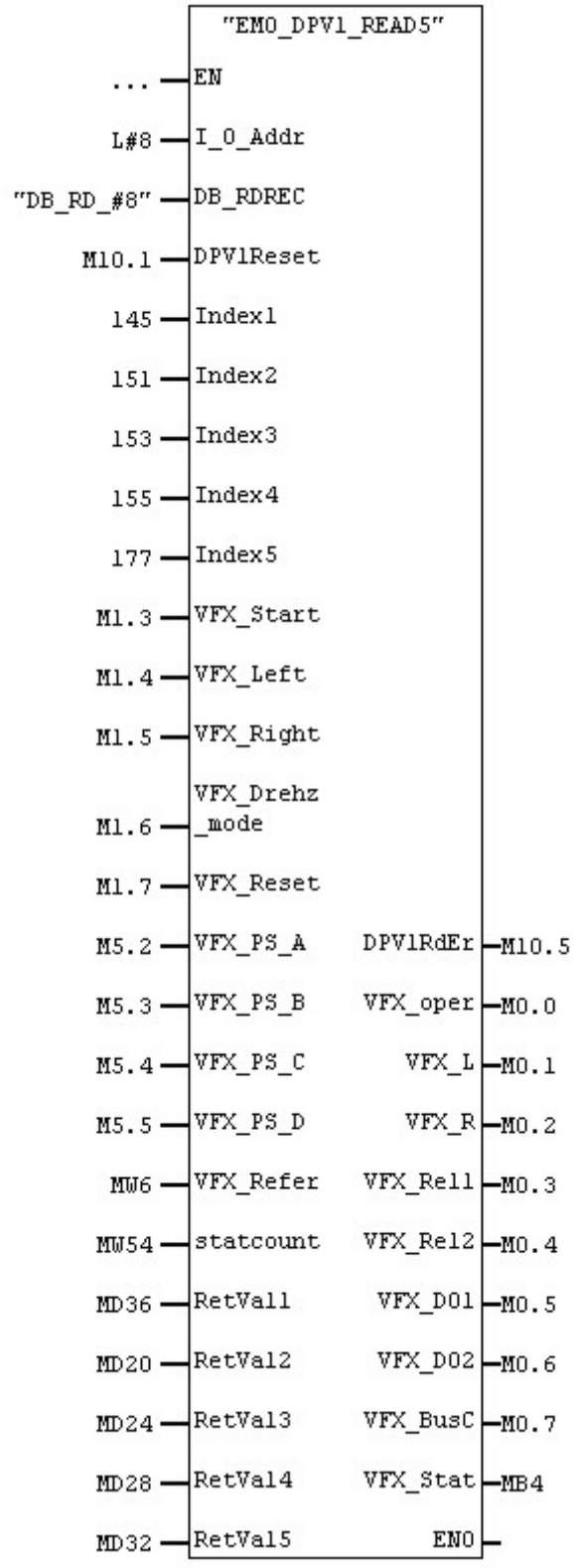
The function block „EMO_DPV1_READ5“ (FC177) controls the complete communication with the Variable Speed Drive at the bus system.

The input variable of the function FC177 **„I_O_Addr“** is the address, that was defined in the hardware configuration (in the example the “8”). The format is “LONG”, i.e. it has to be written as “L#8”.

The function FC177 uses for the DPV1 communication over the bus, the system function SFB52 (RDREC) which needs an instance data block. The number of this DB can be set with the input variable **„DB_RDREC“** (DB250 e.g.).
The format needs the complete name of the data block “DB250”.

If a reading error occurs, the flag **„DPV1RdEr“** will become „True“. In such situation the problem with communication is caused by either wrong configuration, or by the use of faulty “Index” numbers.
The reading of parameters is stopped, until the error is reseted (see flag „DPV1Reset“).

By setting flag **„DPV1Reset“**, reading errors can be reseted.
Before making a reset, the cause should be cleared and solved. Especially the “Index” numbers should be verified.
This flag should be set to “True” only one single time.



With the version L2DPV1-K01.ZIP also the input variable **„VFX_Drehz_mode“** (Drive mode) is available for use in combination with the VFX-series. When setting this flag to “1”, the drive mode “Speed” (direct torque control) is activated. When this flag is set to “0”, the drive is working in V/Hz mode (the FDU-series has only V/Hz mode available).

With the five input variables **“Index1”** … **“Index5”** one can select the returned parameters, read from the VSD; they can be entered direct as a number or as a variable.
Caution: Only valid “Index” values can be used (see the list below).

With the flag **„VFX_Start“** you can set the device (the drive) in to operation.
At the same time you have to set one of the flags **„VFX_Left“** or **„VFX_Right“** which defines the direction of rotation.

If the fieldbus slave device itself switches to an error state you can reset it with the flag **„VFX_Reset“**.

There are four variables **„VFX_Par_Set_A“** …. **„VFX_Par_Set_D“** to select one of the four parameter sets of the Variable Speed Drive.

With the input variable **„VFX_Refer“** the reference value can be set in a range from 0 to 1023. The value 1023 represents the maximum speed, which is set at the device itself in menu 343.

The variable “**statcount**” is a counter which is only for observing the internal work of the function block.

With the five variables **“RetVal1”** … **“RetVal5”** the actual value of the, by the index number selected parameters, are read from the VSD.
Some values do have a polarity sign (two complement description). When they are needed as an absolute value, they can be converted with use of the functions FC198 „Abs_Wert16“ or FC199 „Abs_Wert32“.

The output flags of the function block **„VFX_oper“**, **„VFX_L“**, **„VFX_R“** show the actual operation status and the direction of rotation of the drive. Because the direction can be shown even when at standstill, only the flag **„VFX_oper“** can be used to determine the operation status.

With this project, also the actual status of the relays and the digital outputs of the device are available, see variables: **„VFX_Rel1“**, **„VFX_Rel2“**, **„VFX_DO1“**, **„VFX_DO2“.** The functions have to be set at the device itself.

The output flag **„VFX_BusC“** shows if the device is controlled by field bus communication or not. Because it is possible to switch between control via ‘communication’ and ‘remote’ via a Digital Input, it makes sense to monitor this flag to determine the control source.

The output variable **„VFX_Stat“** gives a code of warnings or faults occurring in the Variable Speed Drive.
A simple binary error message is via a Relay or Digital Output signal as bus signal available, when they are programmed correctly in the device itself.

Existing error codes for **“VFX_Stat”** respectively menu 722/Index 160:

0=No Error	16=Over temp from power board
1=I2t	17=Over curr F
2=PTC	18=Over volt D
3=Motor lost	19=Over volt G
4=Locked rotor	20=Over volt M
5=Ext trip	21=Over speed
6=Mon MaxAlarm	22=Under voltage
7=Mon MinAlarm	23=Power fault
8=Comm error	24=Desat Error
9=PT100 Trip	25=DC-Link Error
10=Deviation	26=Int error
11=Pump	27=Ovolt m cut
12=ext. Motortemp.	28=Over volt. Warning
13=Error 13	29=Error 29
14=Error 14	30=Error 30
15=Option	31=Error 31

VAT1:

With the table of variables VAT1 („Test-FU“) the example program can be tested. All described control functions and data exchange are possible to use.

Following parameter can be read from the VSD with the function block FC177 and selecting the related **Index**-number:

VSD Menu	Description	**Index**	Example
711	Process value (see Menu 321)	145	1490 = 1490 rpm
712	Speed	146	1490 = 1490 rpm
713	Torque Nm	147	36 = 3,6 Nm
713	Torque %	148	24 = 24% Mot.Torq.
714	Shaft Power	149	4650 = 4,650 kW
715	Electrical Power	150	4820 = 4,820 kW
716	Current	151	107 = 10,7 A
717	Output voltage	152	3920 = 392,0 V
718	Frequency	153	498 = 49,8 Hz
719	DC Voltage	154	5275 = 527,5 V
71A	Heatsink Temperature	155	377 = 37,7°C
71B	ext. PT100 1 (if existing)	156	377 = 37,7°C
71B	ext. PT100 2 (if existing	157	377 = 37,7°C
71B	ext. PT100 3 (if existing)	158	377 = 37,7°C
721	VSD Status	159	Emotron-Code
722	Faults / Warnings	160	s. below „VFX_Stat“
723	Digital Inputs	161	Byte0 DI 87654321
724	Digital Outputs	162	Byte1 Rel 321 Byte0 DO 21
725	Analogue Input 1	163	25 = 25%
725	Analogue Input 2	164	25 = 25%
726	Analogue Input 3	165	25 = 25%
726	Analogue Input 4	166	25 = 25%
727	Analogue Output 1	167	25 = 25%
727	Analogue Output 2	168	25 = 25%
728	Status I/O Board 1 (if existing)	169	Emotron-Code
729	Status I/O Board 2 (if existing)	170	Emotron-Code
72A	Status I/O Board 3 (if existing)	171	Emotron-Code
731	Run time hours	172	114 = 114 hours
731	Run time minutes	173	45 = 45 minutes
731	Run time seconds	174	18 = 18 seconds
732	Mains time hours	175	285 = 285 hours
732	Mains time minutes	176	31 = 31 minutes
732	Mains time seconds	177	20 = 20 seconds
733	Energy	178	32850 = 32,85 kW
620	Logic Y	179	Emotron-Code
630	Logic Z	180	Emotron-Code
921	VSD. Type	181	Emotron-Code
922	Software	182	Emotron-Code

2.4. Two or more devices connected to the field bus

The example project contains the function and data blocks for the communication with two variable speed drives connected to the field bus. The function block FC177 can be called for all devices, only the instance data block (DB250) should in such a case be used in separate copies for each individual devise (i.e. DB252).
The maximal number of fieldbus devices depends on the performance and the memory size of the used Simatic S7 CPU.

3. Version for control and random reading and writing of data

With this version it is possible to read to the PLC or write to the VSD (Variable Speed Drive) a random selection of parameter data. The DPV1 functionality is used to read and write the parameters.

3.1. Simatic S7 example project

Emotron makes an archived example project (L2dpv1_E.zip) available, which contains all necessary elements for the Profibus communication with the Emotron Variable Speed Drives.

The following blocks are part of the example project:

OB1 example for the function call

FC137 Function for the communication with the Emotron devices

SFB52 System function for reading (used in FC137)
DB250 Instance DB for SFB52
DB252 Copy of DB250

SFB53 System function for writing (used in FC137)
DB251 Instance DB for SFB53
DB253 Copy of DB251

DB152 Emotron data block for communication and parameter data

DB153 Copy of DB152, for the communication with a second device at the bus

VAT1 Table of variables for test purposes

3.2. Configuration of the S7 project with an Emotron Variable Speed Drive

The configuration for this version is identical with the “Read” version. See chapter 2.2 for configuration of the PLC.

Description of the function and data blocks

All variables of the type „Marker“ (Bit, Byte, Word) are only an example and should be replaced by real variables of your project..

OB1/FC137:

To present the function call for communication with the Variable Speed Drive in an easy way, it is written in the main organizational block OB1.
The Emotron project is run-able and can be used for testing the communication.

The function block „EMO_DPV1_COMM“ (FC137) controls the complete communication with the Variable Speed Drive connected to the bus system.

The input variable of the function FC137 **„I_O_Address“** is the address, that was defined in the hardware configuration (in the example the “8”). The format is “LONG”, i.e. it has to be written as “L#8”.

The variable **„DB_Number“** is the number of the data block containing all parameters and values from the fieldbus slave device (DB152 e.g.). The format is “Integer”, so that only the number “152” has to be filled in.

The function FC137 uses for the DPV1communication the system functions SFB52 (RDREC) and SFB53 (WRREC). Both of them need an instance data block, the numbers of these can be set with the input variables **„DB_RDREC“** and **„DB_WRREC“** (DB250 and DB251).
The format needs the complete name of the data block “DB250” or “DB251”.

You can set the flag **“Emo_DPV1_READ”** to „True“, to start the periodical reading of all parameter, which are marked for reading. (For further explanation see block “DB152”)
The flag can remain permanent „True“ as long as the parameter data are supposed to be read. All selected parameters are then read periodically.

If you set the flag **“Emo_DPV1_rd_All”** to „True“, **all** parameters of the Variable Speed Drive, which are included in the data block (DB152) will be read once.
This flag should be set to “True” one single time.
The flag “**Emo_DPV1_READ**” has to be „True“ at the same time.

```
CALL  "EMO_DPV1_COMM#1"                    FC137
 I_O_Address      :=L#8
 DB_Number        :=152
 DB_RDREC         :="DB_RD_#1"             DB250
 DB_WRREC         :="DB_WR_#1"             DB251
 Emo_DPV1_READ    :="Emo_DPV1_READ"        M10.0
 Emo_DPV1_rd_All  :="Emo_DPV1_rd_All"      M10.2
 Emo_DPV1_WRITE   :="Emo_DPV1_WRITE"       M10.3
 Emo_DPV1_wr_All  :="Emo_DPV1_wr_All"      M10.4
 Emo_DPV1_Reset   :="Emo_DPV1_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
 DPV1_busy        :="Emo_busy"             M10.7
 DPV1_read_error  :="Emo_DPV1_rd_error"    M10.5
 DPV1_write_error:="Emo_DPV1_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
```

If a reading error occurs, the flag **„DPV1_read_error“** will become „True“. The reading of parameters stops, until the error is reset (see flag „Emo_DPV1_Reset“).
You can analyze the error in this case with the help of data block DB152 (see “DB152”).

By setting flag **„Emo_DPV1_Reset“** write or reading errors can be reset.
Before making a reset the cause should be clarified. (see block “DB152”)
Some parameters can’t be written during operation for example.
This flag should be set to “True” one single time.

You can set the flag **“Emo_DPV1_ WRITE”** to „True“, to start the writing of all parameters, which are marked for writing. (For further explanation see block “DB152”)
The flag should be set always again, when changed data are supposed to be written.

If you set the flag **"Emo_DPV1_wr_All"** to „True", all parameter which are included in the data block DB152 will be written one time.
This flag should be set to "True" one single time.
The flag "Emo_DPV1_ WRITE" has to be to set „True" at the same time.
***CAUTION:* You should be sure, that you have correct data in the block DB152, before you start the writing of all parameters. For example you can start reading of all data at first.**

If a writing error occurs, the flag **„DPV1_write_error"** will become „True". The writing of parameters stops, until the error is reset (see flag „**Emo_DPV1_Reset**").
You can analyze the error indicated by „**DPV1_write_error**" with the help of data block DB152 (see block "DB152").

The flag **„DPV1_busy"** shows that the communication block is still running and a read or write action is ongoing.

With the flag **„VFX_Start"** you can set the device (the drive) in to operation.
At the same time you have to set one of the flags **„VFX_Left"** or **„VFX_Right"**, which defines the direction of rotation.

If the fieldbus slave device itself switches to an error state you can reset it with the flag **„VFX_Reset"**.

There are four variables **„VFX_Par_Set_A"** …. **„VFX_Par_Set_D"** available to select one of the four parameter sets of the Variable Speed Drive.

With the input variable **„VFX_Reference"** the reference value can be set in a range from 0 to 1023. The value 1023 represents the maximum speed, which is set at the device itself in menu 343.

The output flags of the function block **„VFX_operation"**, **„VFX_L"**, **„VFX_R"** show the operation status and the direction of rotation of the drive. Because the direction can be shown even when at standstill, only the flag **„VFX_operation"** can be used to determine the operation status.

With this project, also the actual status of the relays and the digital outputs of the device are available, see variables: **„VFX_Relais_1"**, **„VFX_Relais_2"**, **„VFX_DigOut_1"**, **„VFX_DigOut_2".** The functions have to be set at the device itself.

The output flag **„VFX_BusControl"** shows if the device is controlled by the fieldbus communication or not. Because it is possible to switch between control via 'communication' and 'remote' via a Digital Input, it makes sense to monitor this flag to determine the control source.

The output variable **„VFX_Status"** delivers a code with warnings or faults occurring in the Variable Speed Drive.

The variables **„rpm"**, **„Nm"**, **„kW"**, **„Amp"** and **„Temp"** deliver in an easy way the actual values of speed, torque, shaft power, current and heat sink temperature of the device. They are, as many other variables, available in the data block DB152. The values are in these variables presented without a polarity sign. In DB152 „rpm", „Nm", „kW" are presented with a polarity sign included (two complement description).

DB152:

With this data block the entire communication is parameterized and controlled.
The DB152 in the example project is prepared with many inverter parameters.
Other parameters which do exist in the VSD may be included into the data block.

+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#3	

With the byte address 40 the actual parameter data starts.
For every parameter which is supposed to be read from, or to be written to the device, a **field of 20 byte** is used, and can be configured as following:

Activ_xxx – bit code to define whether this parameter is supposed to be written or to be read.
The individual bits have following meaning if you set it to “True”:

Bit 0: (decimal “1”) This parameter is read periodically („Emo_DPV1_READ“).

Bit 1: (decimal “2”) This parameter is written with „Emo_DPV1_ WRITE”.

Bit 2: (dec. “4”) At this point of the data block the periodically reading stops.
(Even if “read” marked parameter follows.)
(This parameter itself will not be read.)
(By stopping here and start again at the top, makes the cycle faster.)

Bit 3: (dec. “8”) At this point of the data block the writing with „Emo_DPV1_ WRITE” stops.
(Even if write marked parameter follow.)
(This parameter itself will not be written.)

All parameter which have Bit 0 equals “1” at „**Activ**_xxx“ are read periodically.
(At the example above parameter 71A, 723, 725, 725A, 726.)
The periodically reading is interrupted with a set Bit 2 in „**Activ**_xxx“.
See line 220 at the example, the “6” contains the Bit 2. All following parameters will now only be read with the input variable „Emo_DPV1_rd_all“.
All parameter which have Bit 1 set to “True” at „**Activ**_xxx“ are supposed to be written.
In line 220 the Bit 1 is also set, which means that the parameter 351 (Max. Torque) will be written to the drive. Besides this data, also the actual status of Relais 3 will be written (parameter 5531, from line 240), because this relay can be controlled via the bus.
If an „**Activ**_xxx“ has a set Bit 3, the periodically writing will stop.
See line 260 in the example, the “8” is the equivalent of a set Bit 3. All following parameters will now only be written with the input variable „Emo_DPV1_wr_all“.

Slot_ID_xxx – Communication access slot ID (already programmed).
(Have a look at the manual if you want to extend the data block.)

Index_xxx – Communication access Index (already programmed).
(Have a look at the manual if you want to extend the data block.)

Data_xxx – This is the read value or the value supposed to be written.
(Determined for use in the application program.)

Status_xxx – Error code in the case of read or write failure, normally equals zero.
(If “DPV1_read_error” or “DPV1_write_error” is “True”,
you should look here.)
These errors could occur (Byte 3 and 2 of the double word “Status”):
80 B0 - invalid index
80 B1 – write length error
80 B2 – invalid slot
80 B6 – access denied (can’t be written during operation)
80 B7 – out of range (invalid value during writing)
Other codes indicate general problems with the data block.

Length_xxx – Length of data (2 or 4 byte).

Fill_xxx – Spare

Configuration of the data block:

You can add or delete one or more parameters from the data block (DB152).
Some things should be considered in this case:

- The range from address 0 to 36 is used by the communication and should not be changed.

- You always have to copy, insert or delete the complete 20 byte, which belong to one parameter.

- The first five parameters (at address 40, 60, 80, 100, 120) are directly used by the function block FC137 and should not be changed.

- The range that starts with address 140 can be changed.

+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	

- The total length of the data block FC137 is automatically recognized by the program.

- At first all parameter should be listed in the data block, which are marked for periodically reading (Bit 0 in „Activ_xxx“ is “True”).

- The “reading area” has to be closed with Bit 2 in „Activ_xxx“.
  See line 200 „351 Max. Torque” and „5531 ON/OFF Relais 3” in the example.)

- After the “reading” parameter should those follow, which are supposed to be written (Bit 1 in „**Activ**_xxx“ is “True”).

- The “writing area” has to be closed with Bit 3 in „**Activ**_xxx“.
  (See line 240 at the example.)

+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	

In your user program the data can be accessed as follows:

L DB152.DBD248 // load „Data_211“ – Language

or in the symbolic description:

L „DB_DPV1_#1“.“Data_211“
(the variable „Data_xxx” is always the real parameter number)

VAT1:

Table of variables to perform a test with the example project.
The Variable Speed Drive can be started, stopped, controlled by speed and direction. You can set the flags to start the periodical reading or writing, the one time reading or writing of all parameter. The flags which indicate an error are to be seen, you can also reset an error. And you see the data read out of the device and the data supposed to be written.

3.3. Two or more devices at the fieldbus

The example project contains the function and data blocks for the communication with two variable speed drives connected to the field bus. Every device needs one parameter data block and the two small data blocks for the system calls, for the first device these are DB152, DB250 and DB251. These blocks can be copied for every additional device connected to the field bus as shown above.
The maximal number of fieldbus devices depends on the performance and the memory size of the used Simatic S7 CPU.