Installation and start-up instruction

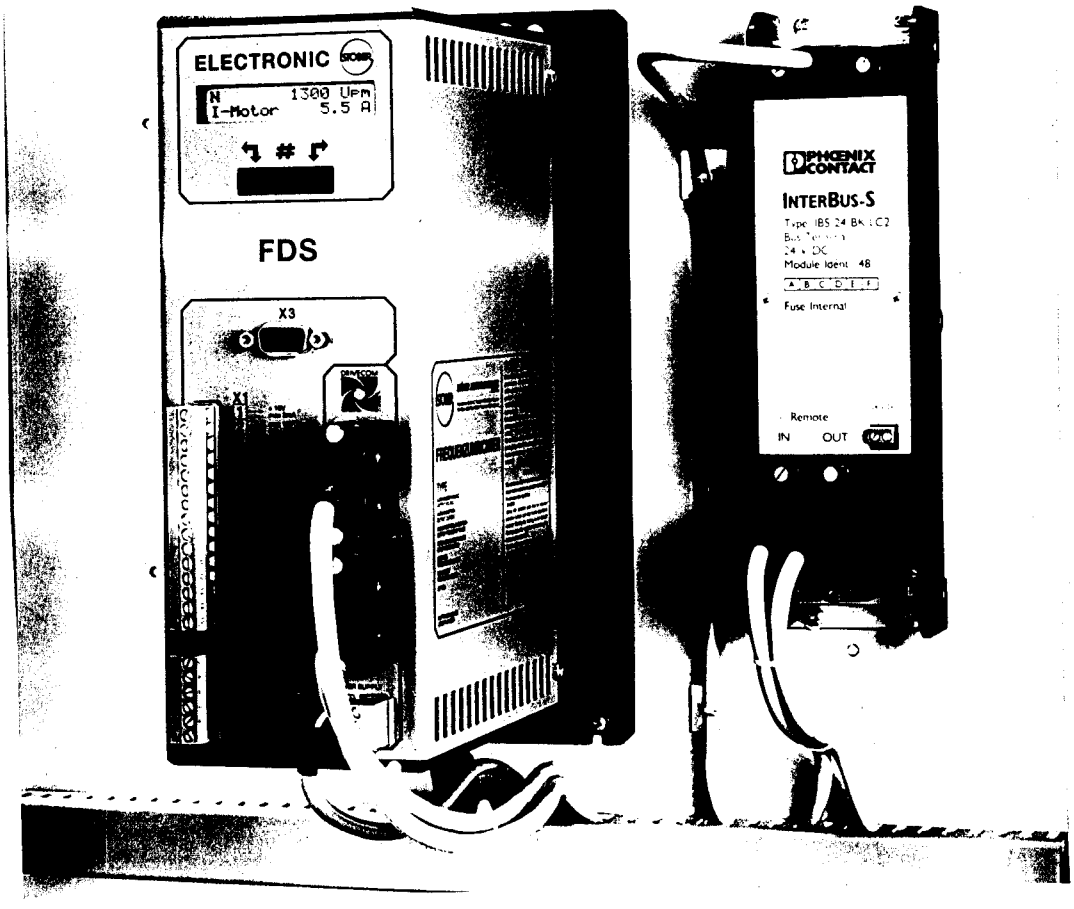
Interbus-S Communication with Localbus-link

Software V 3.1_4

(Modification Rohrwerkanlage)

for Frequency inverter series FDS

Software D 3.1_4



To avoid problems caused by incorrect use, make sure you read the complet manual "Installation an start-up instruction for Frequency inverter series FDS"! Adittionally it is necessary to read the special manuals from Phönix-Contact about the Interbus-S Board and the PCP-Software.

Nr. 440 718

Contents

1 Preface

Within the framework of factory automation, increasingly powerful and more flexible systems are needed in industrial power transmissions. Digitized drive controllers are capable of fulfilling these requirements. However, open and standardized communication capabilities are needed to enable their complete integration into complex production sequences.
The basic concept of open systems is to enable an exchange of information between application functions implemented on hardware from a diversity of manufacturers.
These include defined application functions, a standard user interface for comunications and a standard transmission medium.
The DRIVECOM User Group has taken on the task of standardizing the most important drive hardware functions and of summarizing them in this profile for variable-speed drives.
The Profile Power Transmission with the Profile Number 21 in the Version 1.0 has been declared as binding by the DRIVECOM User Group.

The Interbus-S system, which fulfills the demands of power transmission in terms of real-time response and a standardized user interface, was chosen as the transfer medium. For the user, this profile definition is a meaningful addition for standardized communications and represents a generally valid convention concerning the contents of data and the response of hardware. These function definitions standardize a few essential drive parametes. Consequently, hardware from different manufacturers exhibits the same respose in the communication medium when using these standard parameters.

The frequency inverters of series FDS with Interbus-S Communication enable the control with a part of Parameters of DRIVECOM Profile No. 21 an additionally manufacturer specific parameters.

2 Safety Pecautions

All the safty precautions in the Installation and start up instruction of Frequency inverters series FDS are here valid, too. You shoud read that complet operation instruction manual.
Stöber Antriebstechnik is not liable for any damage done by not following these instructions or other local and state requirements.

Additionally the following Warings are valid:

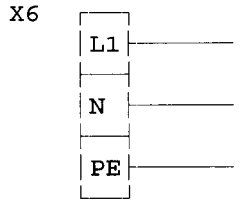
By a break down of 230V-power supply at X6 (only at FDS 1040/B ... 1085/B) the FDS still keeps running with the last transmitted Nominal Speed Value!
A Shutdown of the drive is only possible by remove of the H-Level at binary Input "Enable" or by remove of the power supply.

By a break down ot the Interbus-S it is possible to stop the drive by using several Parameters of the Check Functions (see chapter 6.3).

3 Electrical Installation

It is neccessary to read the hints about the mechanical and electrical Installation in the Installation and start up instruction for Frequency inverter series FDS first!
Additionaly the following hints are important:

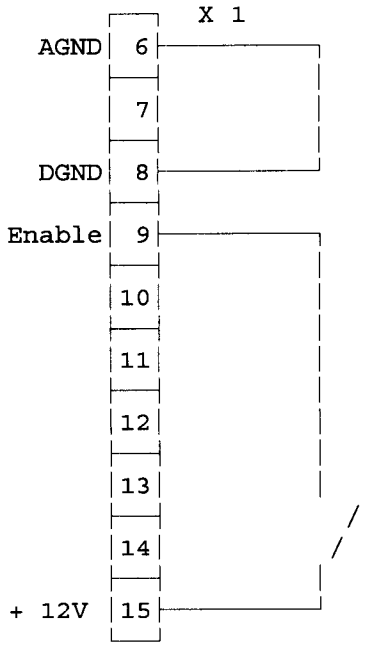
Power supply



By FDS 1040/B ... 1085/B
additionally connect the Voltage
of 230V~ ±10%, 50 Hz
to X6.L1, X6.N and X6.PE.

By FDS 1110/B ... 1200/B that is not neccessary.

Binary Inputs

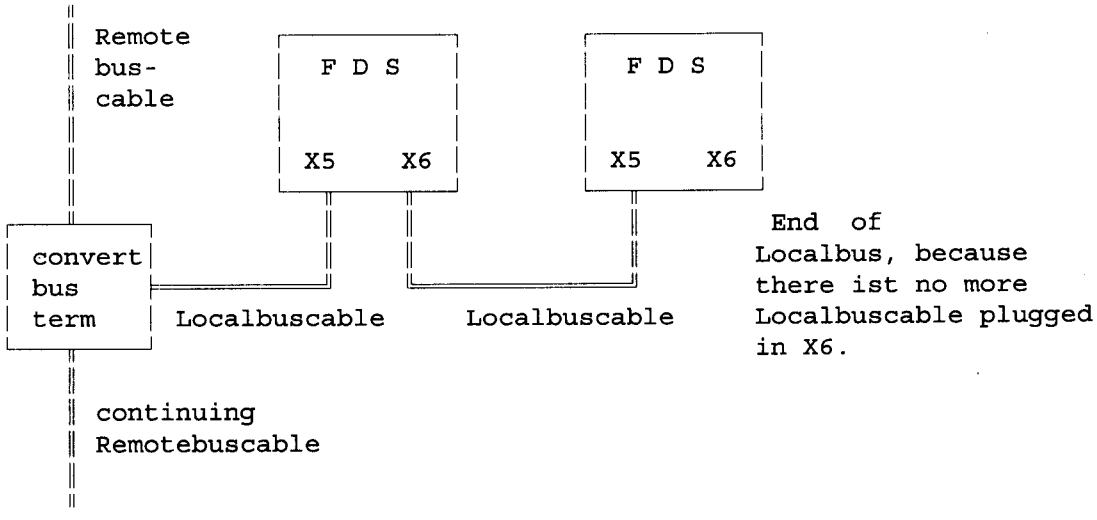


For Controlling the FDS from Interbus-S Communication has binary Input Enable (at X1) must be connect to a H-Level. This Input works as an additional Enable to the Control commands from the Control-Word which is send on Interbus-S

The other Inputs need not be connected for a correct operation with DRIVECOM profile.

Bus Connecting

Example Configuration:



The Interbus-S is devided in a Remotebus and several Localbussegments. At the Interfaces between Remotebus and Localbus is a Converter bus terminal module (BK). Between these Modules the Remotebus is limited at a length of 400m. Up to 64 Converter bus terminal modules are connectable. So it is possible to reach a summary extent of 12,8 km. Every local bus segment has a maximal extent of 10m and a maximal number of 8 modules. The maximal distance of two modules in the local bus is 1,5m

The Frequency Inverters series FDS with Interbus-S Communication are provided with the Local bus Interface. The apperaring local bus cable is to be plugged in to X5 (local in). It is possible to continue the local bus segment by plugging a following local bus cable in to X6 (local out).

Meaning of the LED "BA": Is lightning, if the optional board is connected to power supply and the power supply of the local bus is comming in from the local bus cable but the Interbus-S is not active.
Is blinking, if Interbus-S is active.

4 Commissioning

The Frequency inverters series FDS with Option Interbus-S Communication can be controlled like every "Standard"-FDS. They include the complete functions of the Standard-Device FDS.

Note: The FDS 1040/B, FDS 1070/B and FDS 1085/B with the modification "Rohrwerkanlage" (Software D 3.1_4) do not include the Functions of the serial connection to FDS-TOOL. All other Function are exactly the same as in the Standard-FDS.
The global switch of the control source of the FDS is the parameter *"Operation input"* in the group *"Operation"* . For FDS with Interbus-S Communication the basic adjustment of this parameter is "optional board". With this adjustment the FDS waits for Nominal-Speed-Value and Control Commands from Interbus-S. But it is possible to change the setting ot all Parameters from the Display or from Parabox at any time.

4.1 Commissioning without use Interbus-S

For commisionig a drive it is favorable to start up and test the FDS without controlling from Interbus-S. It is possible to use therefor the *"key control"*. For testing the drive by key control it is neccessary to set the parameter *"Operation input"* to key control. Then use the FDS the signals from the binary inputs Enable, Clockwise and Counter clockwise. Only the set point adjustment comes from the three keys beneath the display.

With this start up procedure you can test the FDS with the connected motor and check the correct settings of the parameters. After this test you shoud switch the parameter *"Operation input"* to "optional board" and save the parameters with the action *"save values"*.

After that you can control the FDS by the Interbus-S and adjust all parameters in the "normal" manner per display and keys. In this case it is not neccessary to read the chapters 6 and 7 (Parametrization of FDS with Interbus-S)

4.2 Commissioning with Interbus-S Communication

After the start up procedure of FDS in chapter 4.1 it is now the moment to connect the FDS to the Interbus-S system.

Before this you have to read the operation instructions of all other used Interbus-S modules and -boards. (Manuals from Phönix Contact like: Reference Manual for Peripherals Communication Protocol and Operating Manual for Controller board)

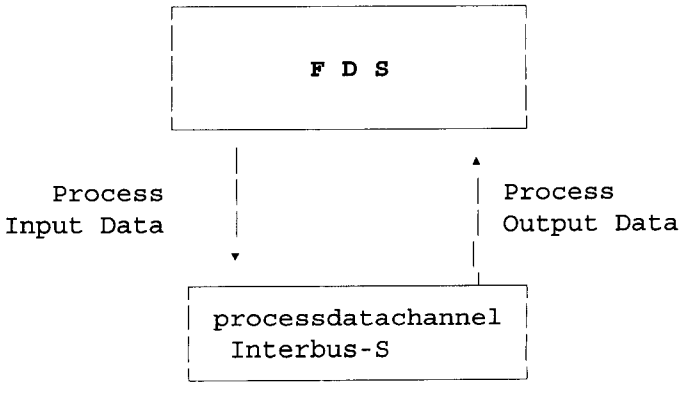
The following technical data describe the FDS as local bus participant.

Identification Code

Every Interbus-S participant has a Identification Code (ID-Code), which ist important for the Interbus-Master. For the FDS with Interbus-S Communication the ID-Code is defined:
- manufactorer specific Code (ID0 - ID7) : 195 (able to communicate with PCP)
- width of Processdata (ID8 - ID10): 3 Words

Mapping of Process data

The direction ot process data is defined by looking from the Interbus-S. i.e.:
Process Output Data are data, which are read by FDS from the process data channel.
Process Input Data are data, which are written by FDS to the process data channel.



Mapping of the Process Data Word's:
Proces Output Data
1. Word : *"Control Word"*
2. Word : *"Nominal Speed Value"*

Process Input Data
1. Word : *"Status Word"*
2. Word : *"Actual Speed Value"*

The Word No. 0 ist reserved for PCP-Communication and may not be written from the Interubs-S user. This Mapping of Process Data can not be changed! (More details in chapter 6.2 Process Data Control)

4.3 Optional menue

During the Commissioning of the FDS at the Interbus-S it would be usable to observe the actions and data send through the Interbus-S.

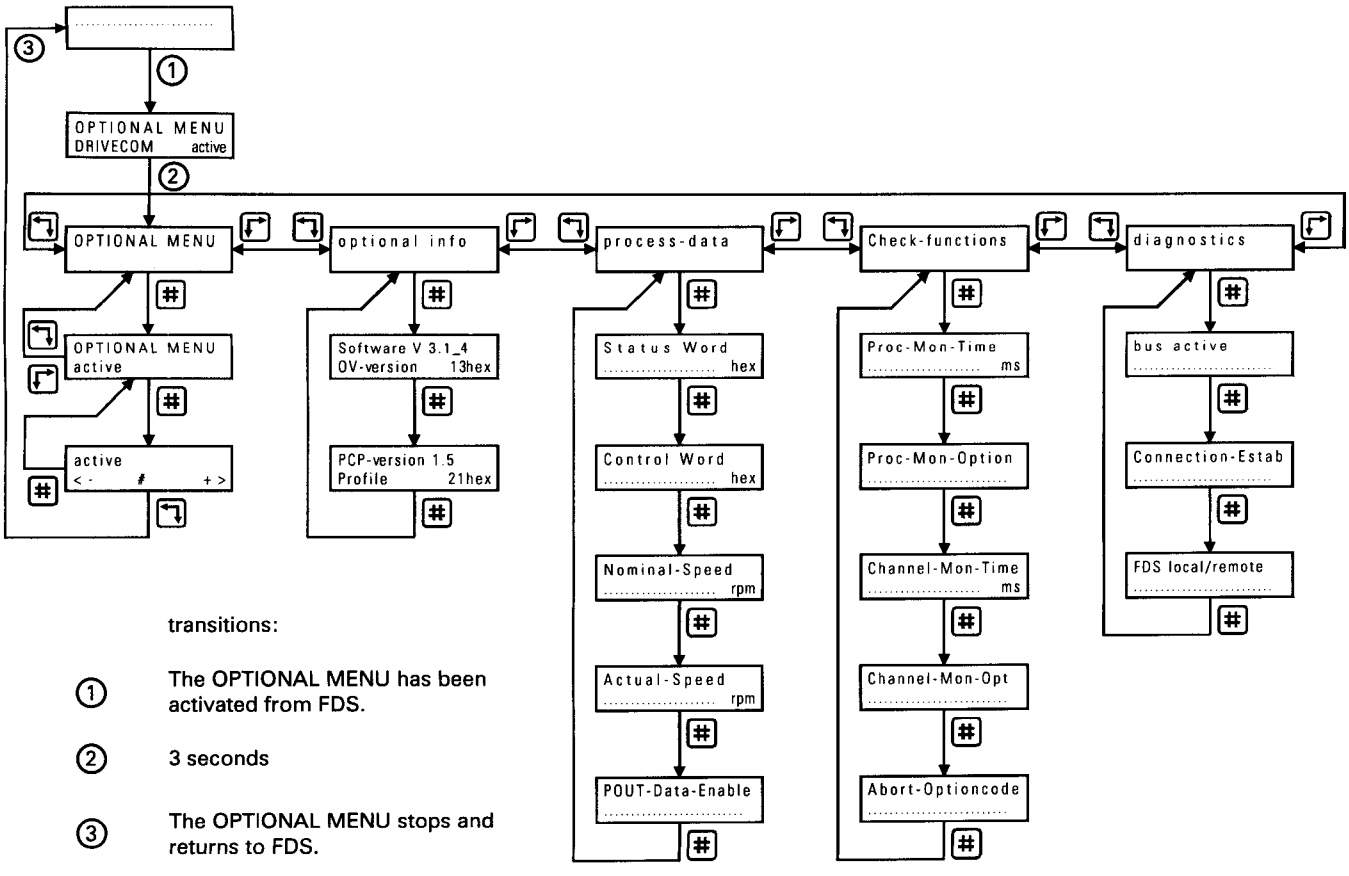
Therefor an Optional Menue is provided using the FDS' keys and LCD. You can see the aktual values of that parameters, which are not supplied by the normal menue of the Standard-FDS.

The Optional menue is controlled by the Optional board inside of the FDS. Therefor the Optional menue has to be started from the Standard-FDS-menue. To do that, you have to switch the parameter "optional menue" in the group "device data" from inactive to active.
The operation of the Optional menue with the keys and the display is exactly the same as in the Standard-FDS-menue. Several parameters can be shown within four groups.

To stop the Optional menue and switch back to the Standard-FDS-menue you have to switch the parameter "Optional menue" from active to inactive.

The following picture shows a general view over the groups and display-menues of the Optional menue. The meaning of the named parameters can be seen in the following chapters.

OPTIONAL MENU DRIVECOM



transitions:

① The OPTIONAL MENU has been activated from FDS.

② 3 seconds

③ The OPTIONAL MENU stops and returns to FDS.

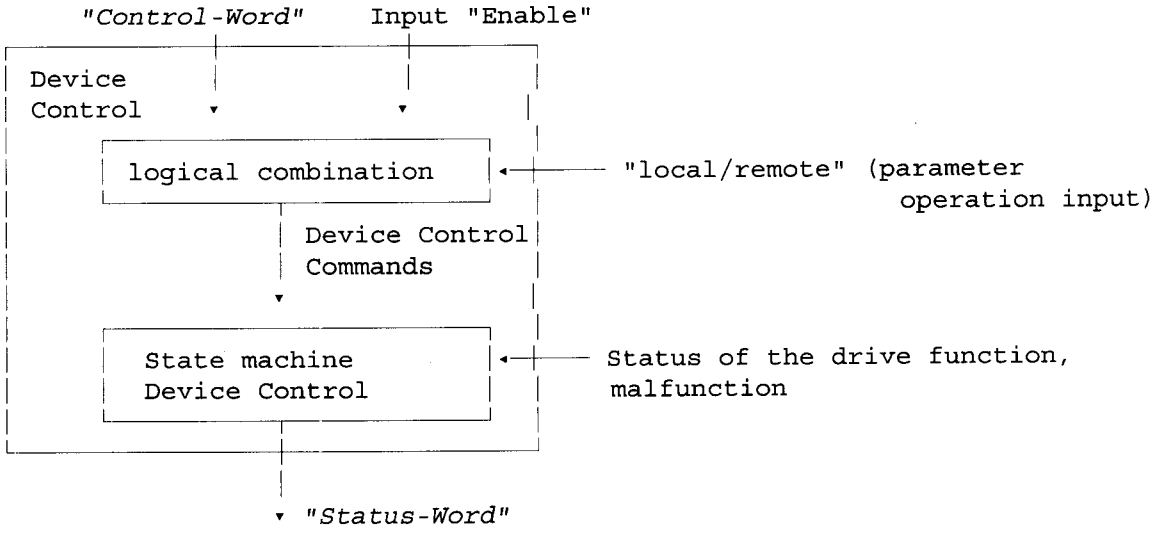
5 Controlling the FDS with Interbus-S

"Controlling the FDS" means: Send a Nominal Speed Value and Control Commands to the FDS for controlling the drive. I.e.: Control Commands like "Switch on", "Shutdown", ...

The function block "Device Control" is used for the manufactoring of the Control Commands, the manufactoring of the Nominal Speed Value is done by the function block "Speed Funcction". These function blocks will be described in details now:
The controlling of the FDS in meaning of the DRIVECOM-Profile is done by the "Control-Word" and the "Nominal Speed Value".

Device Control

The function block Device Control controls all functions of the device (drive function and power section). The control sequence is described by a state machine. Device control is influenced by the "Control-word", the binary input "Enable", the internal signal "local / remote", the drive function status and malfunctions. The device control influences the drive functions. The "Status-Word" is derived from the device status and internal signals and can be read through the bus.



Control-Word

Der Frequency inverter can be controlled by means of the bits in the Control-Word.

Status-Word

The Status-Word parameter provides information about the status and messages of the drive controller.

local / remote

This internal signal indicates whether or not the drive controller parameters can be defined through the bus. By the FDS this signal will be set to local, if the parameter "Operation input" in the group "Operation" is not set to optional board and will be set to remote, if that parameter is set to optional board. If this signal is indicating local, is not possible to change any parameter of FDS through the Interbus-S. Beside this, all Control Commands from the Status-Word will be ignored.

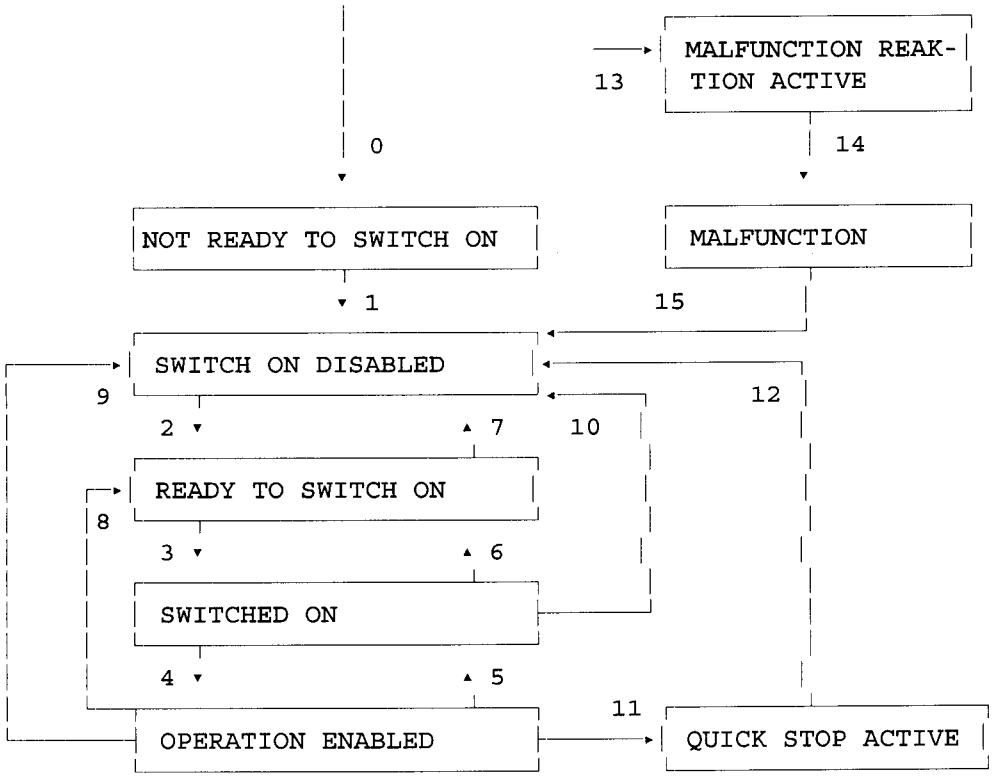
Binary Input "Enable"

The binary Input Enable at the Control Interface (X1) has been connected to a H-Level, therefor the Control Commands coming from the Control-Word have an effect in the FDS and the drive can be run. Oherwise, when no H-Level is at that input, at the same time the drive function will be stoppt and the power section will be switch off. This function can be used to stop the drive directly by the FDS without any operation through the Interbus-S.

5.1 Device Control State Machine

The state machine describes the device statuses and the possible control sequences of the drive. A status represents a specific internal and external response. By means of device control commands and internal events, the status can be changed and a contol sequence thus executed. The current status can be read out by way of the status word.

State diagram



States of the device control:

State	Event / Command	Drive function
NOT READY TO SWITCH ON	- The FDS is receiving its supply voltage - Selftest is running - Initialization is running	disabled
SWITCH ON DISABLED	- Soft / Hardware-Initialization is completed - Switching on is disabled	disabled
READY TO SWITCH ON	- Switching on is enabled	disabled
SWITCHED ON	- Power section is switched on - Motor is under current	disabled
OPERATION ENABLED	- Power section is switched on - Drive is following the nominal speed value	enabled
QUICK STOP ACTIVE	- Quick Stop Function: Drive will be slow down with deceleration ramp	disabled
MALFUNCTION	- Power section is switched off	disabled
MALFUNCTION REACTION ACTIVE	- Power section is switched off	disabled

5.2 Transitions of Device Control

	Transition from state to state	Event / Command	Action
0	Input of the state machine → NOT READY TO SWITCH ON	power supply was switched on	Start Selftest, start Initialization
1	NOT READY TO SWITCH ON → SWITCH ON DISABLED	Selftest completed without errors Initialization completet without errors	no
2	SWITCH ON DISABLED → READY TO SWITCH ON	Command "Shutdown" and Input "Enable" on H-Level	no
3	READY TO SWITCH ON → SWITCHED ON	Command "Switch-on" and Input "Enable" on H-Level	Power section switch on
4	SWITCHED ON → OPERATION ENABLED	Command "Enable-Operation" and Input "Enable" on H-Level	Disable-Drive-Function
5	OPERATION ENABLED → SWITCHED ON	Command "Disable-Operation" and Input "Enable" on H-Level	Drive will be slow down with Deceleration-Ramp
6	SWTICHED ON → READY TO SWTICH ON	Command "Shutdown" and Input "Enable" on H-Level	Power section switch off
7	READY TO SWITCH ON → SWITCH ON DISABLED	Command "Quick stop" or Command "Disable Voltage" or Input "Enable" on L-Level	no
8	OPERATION ENABLED → READY TO SWTICH ON	Command "Shutdown" and Input "Enable" on H-Level	Disable-Drive-Function, Power section switch off
9	OPERATION ENABLED → SWITCH ON DISABLED	Command "Disable Voltage" or Inpupt "Enable" on L-Level	Disable-Drive-Function, Power section switch off
10	SWITCHED ON → SWITCH ON DISABLED	Command "Quick-Stop" or Input "Enable" on L-Level	Power section switch off
11	OPERATION ENABLED → QUICK STOP ACTIVE	Comand "Quick-Stop" and Input "Enable" on H-Level	Drive will be slow down with Decelration-Ramp
12	QUICK STOP ACTIVE → SWITCH ON DISABLED	Quick-Stop has been completed and Input "Enable" on L-Level	Disable-Drive-Function, Power section switch off
13	all states → MALFUNCTION REAKTION ACTIVE	Drive malfunction detected	Power section switch off, save malfunctin in FDS
14	MALFUNCTION REAKTION ACTIVE → MALFUNCTION	Malfunction reaktion concluded	Power section switch off, Disable-Drive-Function
15	MALFUNCTION → SWITCH ON DISABLED	Command "Malfunction-Reset", and Malfunction is no longer present	Malfunction Reset is executed

The status change takes place only if the actions have been executed completely. The sequence of actions corresponds to the sequence in which they are processed when the status is changed. Once the actions have been processed completely, the next status has been attainded and new commands are accepted.

Explanation of above mentioned notions

Disable-Drive-Function, Drive-Function disabled
The FDS shoes following characteristics:
- Nominal Speed Value is not processed,
- The power section can be switched off

Enable-Drive-Function, Drive-Function enabled
The FDS shows following characteristics:
- Nominal Speed Value is processed
- The power section is switched on

Drive malfunctions
Drive malfunctions may occur in all states of the device control. They always cause a changeover to the MALFUNCTION REACTION ACTIVE state. In this state the FDS switches off the power section and saves this malfunction with other data into the internal malfunction memory and, if present at X3 of FDS also into the Parabox. After this action the FDS changes the state to MALFUNCTION. This state can only be left sending the Command Reset-Malfunction. Therefor it is neccessary that the reason of the malfunction is not still existing..

Overview of Control commands and transitions

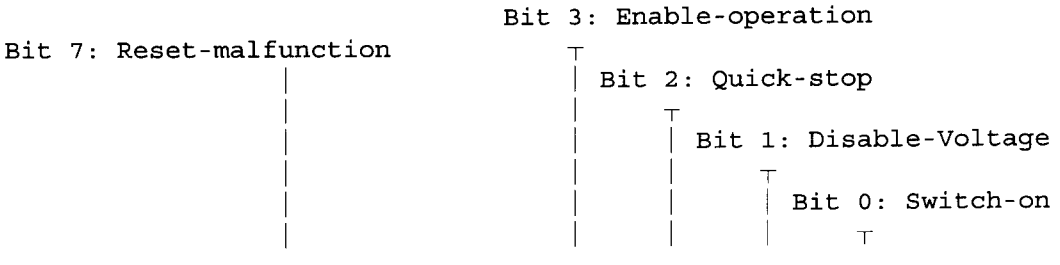
Device control command	triggered state transitions
Input "Enable": L-Level	7, 9, 10, 12
Shutdown	2, 6, 8
Switch on	3
Disable Voltage	7, 9, 10, 12
Quick stop	7, 10, 11
Disable operation	5
Enable operation	4
Reset malfunction	15

5.3 Control-Word

The result of a logical combination of Control-Word and the binary Input Enable produce the device control commands that act on the state machine of the device control. In this way, functions are executed and operation states of the device are defined. The Control-Word is composed of 16 bits that have the following meanings:

Bit-No.	Name
0	Switch-on
1	Disable-Voltage
2	Quick-stop
3	Enable-operation
4	Disable-RFG (RFG = Ramp Function Generator)
5	Stop-RFG
6	RFG-zero
7	Reset-malfunction
8..15	reserved

Commands from Control-Word



Command	7	6	5	4	3	2	1	0	Transitions
Shutdown	X	X	X	X	X	1	1	0	2,6,8
Switch-on	X	X	X	X	X	1	1	1	3
Disable Voltage	X	X	X	X	X	X	0	X	7,9,10,12
Quick-stop	X	X	X	X	X	0	1	X	7,10,11
Disable operation	X	X	X	X	0	1	1	1	5
Enable operation	X	X	X	X	1	1	1	1	4
Reset malfunction	0→1	X	X	X	X	X	X	X	15

Meaning of above used Characters:

0→1: Changing of the Bit from 0 to 1
X: may be 0 or 1

Remaining Bits in Control-Word

Bit 4 Disable-RFG
If the Bit 4 is =0, the drive will slow down with the ramp Speed-Deceleration (Decel1). If the Bit is =1, then the drive will start up with the ramp Speed-Acceleration (Accel1). This Bit has the same effect as connecting H-Level to both binary Inputs "Clockwise" and "Counter Clockwise" at the control interface.

Bit 5 Stop-RFG
If Bit 5 is =0, the ramp function's current output value is frozen. If the Bit is =1, then the drive follows the Nominal Speed Value. The priority of Bit 5 is higher than that of Bit 4.

Bit 6 RFG-zero
If Bit 6 is =0, the drive will slow down with the ramp Speed-Deceleration (Decel1). If the Bit is =1, the drive will start up with the ramp Speed-Acceleration (Accel1). The priority of Bit 6 is higher than that of the Bits 5 and 4.

Bit 8..15 reserved
These bits are reserved for additions.

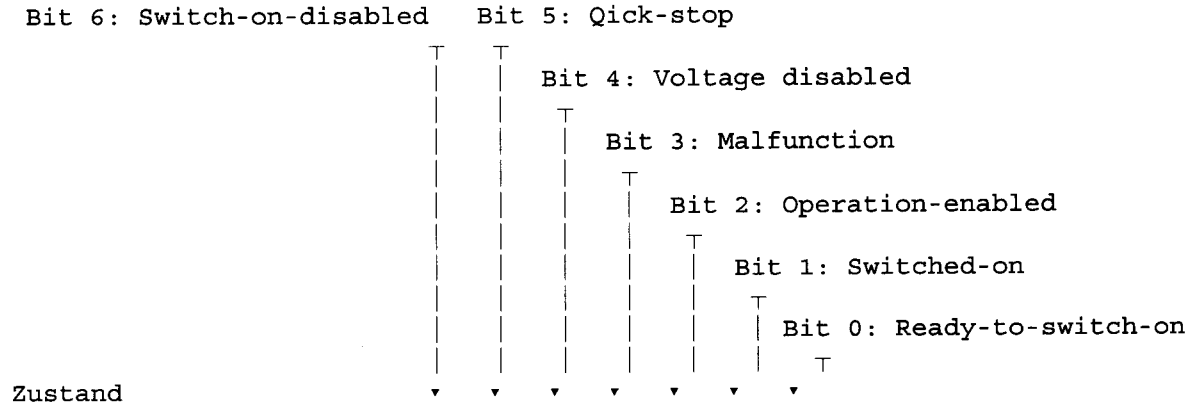
5.4 Status-Word

The Status-Word contains Information about the status and messages of the FDS.

Bit	Name
0	Ready-to-switch-on
1	Switched-on
2	Operation-enabled
3	malfunction
4	Voltage-disabled
5	Quick-stop
6	Switch-on-disabled
7	Warning
8	Message
9	Remote
10	Face-value-reached
11	Limit-value
12..13	reserved
14	binary Input Current Clockwise or BE1 (in case of parameter)
15	binary Input Clockwise or BE2 (in case of parameter)

Display of the device statuses in the Status-Word

The device statuses are indicated by the following bit combinations in the Status-Word



Not-Ready-to-switch-on	0	X	X	0	0	0	0
Switch-on-disabled	1	X	X	0	0	0	0
Ready-to-switch-on	0	1	X	0	0	0	1
Switched-on	0	1	X	0	0	1	1
Operation-enabled	0	1	X	0	1	1	1
Malfunction	0	X	X	1	0	0	0
Malfunction-Reaction-act	0	X	X	1	1	1	1
Quick-stop-active	0	0	X	0	1	1	1

The bits marked X are irrelevant here.

Description of the remaining bits in Status-Word

Bit 4 Voltage-Disabled

The disabled voltage request ist active when bit 4 = 0, or at the binary Input "Enable" has a L-Level.

Bit 5 Quick-stop

The quick stop request is indicated if bit 5 = 0. Alternatively, the status of the device control is indicated depending on other bits.

Bit 7 Warning

A Warning is active. The kind of the warning is shown in the display of the FDS. The FDS keeps up operation for the adjusted time. Afterwards the status is changed to Malfunction-Reaction-Active.

Bit 8 Message

The FDS is not able to perform the required drive load condition. The kind of the message is shown in the display .

Bit 9 Local/Remote

Remote indicates that parameters can be modified through the Interbus-S if bit 9 is = 1

Bit 10 Face-value-reached

This bits indicates that the face-value has been reached when bit 10 is = 1. This bit-function is only active, if the status is Operation-enabled and the bits 4, 5 and 6 are all set to = 1. Otherwise this bit always is set to 1.

Bit 11 Limit-value

This bit indicates that the speed-limit is active when bit 11 is = 1.

Bit 12 u. 13 are reserved

Bit 14 Level of binary Input "Counter Clockwise" or BE1
Bit 15 Level of binary Input "Clockwise" or BE2
In case of the adjustment of the parameter Mapping-Status-Word-Bit14-15 the Level of the selected Inputs will be shown here. A H-Level is represented by a 1.

5.5 Nominal-Speed-Value and Actual-Speed-Value

The notions Nominal-Speed-Value and Actual-Speed-Value have by the FDS the same meaning as Nominal-Revolutions-Value and Actual-Revolutions-Value. These parameters are a part of the Drive Function. The complete description of these parameters follows in the next chapter.

The Nominal-Speed-Value ist the Control-Input for the Speed-Function. This parameter is signed 16bit-Number (Interger16) and has the unit Revolutions per minute. It can be accessed over the process data. The result of the Speed-Function is shown in the parameter Actual-Speed-Value. It has the same unit. Both parameters can also be accessed by PCP-Communication.

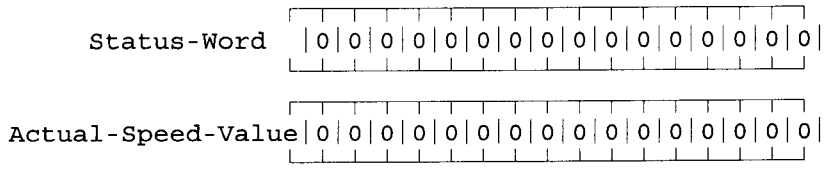
For all signed speed parameters, the positive value denotes rotation to the clockwise and the negative value denotes counter clockwise rotation (when viewing the drive shaft).
The minimum or maximum values (e. g. -32768 and +32767 for Integer16) are output if the value of an object exceeds the value range that can be represented.

5.6 Example of simple Control with Interbus-S

To cotrol a frequency inverter with the Control-Word, many Device Control Commands and bitcombinations exists. In most applications not all Commands are neccessary. Here a simple example will be shown with use of the minimal amount of commands and the Nominal-Speed-Value:
A SPS has to control a FDS 1040/B with a motor. The motor has an effective power of 1,1 kW, 4 poles and no gear. The acceleration- and deceleration-time in this application are irrelevant. So we do not need to manipulate any parameter with the display and keys or with the PCP-Communication. We can start up the drive with the basic adjustment from the FDS. The complete control can be done by process data.
The binary Input "Enable" at X1 is fixed to H-Level.

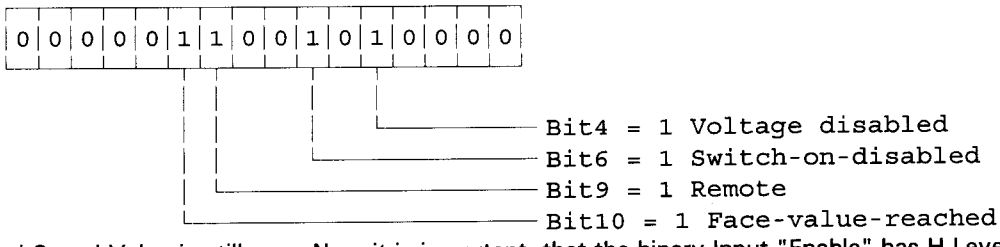
1 Switch on the 230V Voltage supply for the Optional board Interbus-S at X6 of FDS 1040/B, 1070/B, 1085/B and start the SPS and Interbus-S. In this moment all bits in Status-Word and Nominal-Speed-Value are still zero. (The FDS 1110/B ... 1200/B have no separate power supply for the Optional board Interbus-S. Then Point 1 is irrelevant)

Observing the *"Status-Word"* and *"Actual-Speed-Value"* shows:



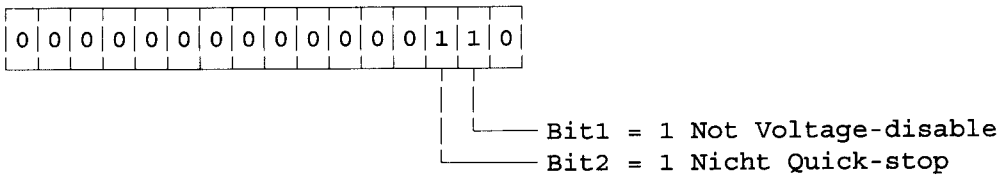
Because the 400V Voltage supply of the FDS is not yet switched on all bits are zero.

2 Switch on the 400V Voltage supply to FDS
After appropiately 2 seconds "inhibited" appears in the display . The content of *"Status-Word"* changes to:

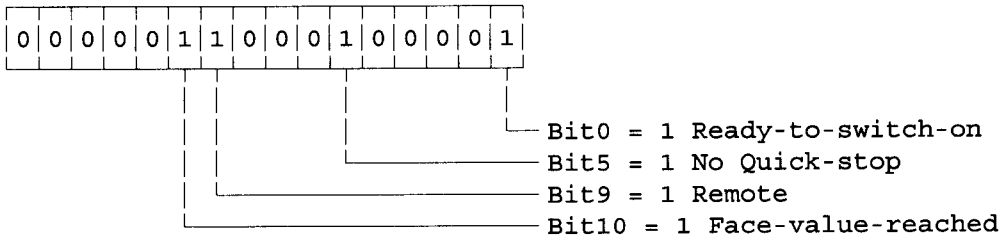


The Actual Speed Value is still zero. Now it is important, that the binary Input "Enable" has H-Level present, otherwise bit 10 is 0 and all following steps in this example are not executable. In this case the FDS would be remain in status Switch-on-disabled.

3 Setting the Device Control Command Shutdown in *"Control-Word"*:

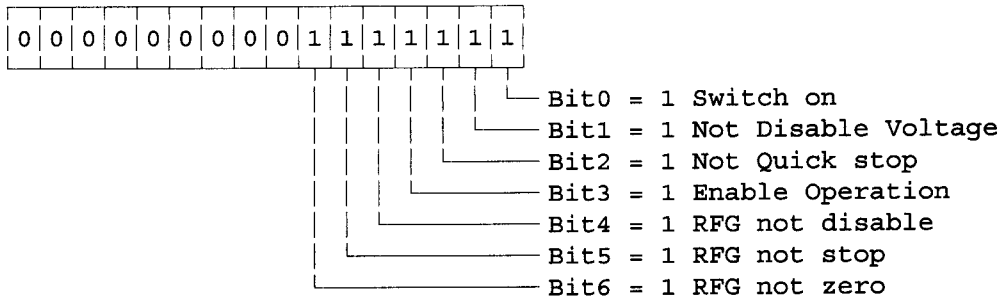


Reaction of FDS: display will read: "ready / not enabled"
The *"Status-Word"* changes to:



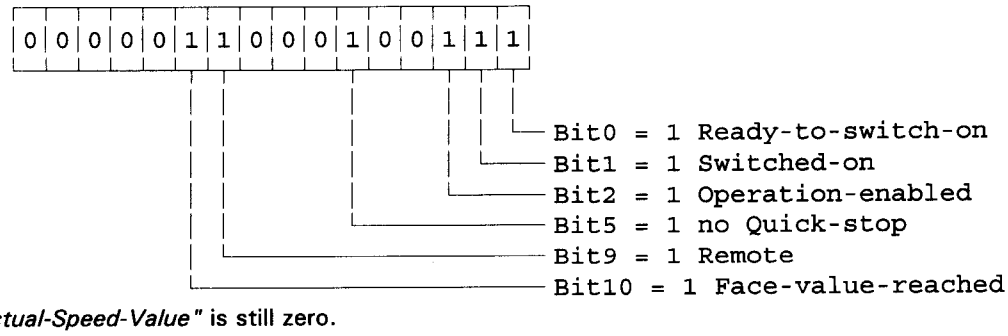
The *"Actual-Speed-Value"* is still zero.

4 Setting the Device Control Command "Enable operation" and the bits 4, 5, 6 für the RFG



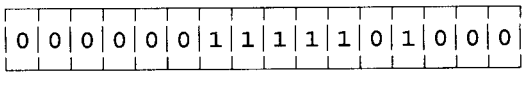
Reaction of FDS: The power section is switched on and the display reads:
"....rpm ....A" / "clockwise ".

The answer of FDS in *"Status-Word"*:



The *"Actual-Speed-Value"* is still zero.

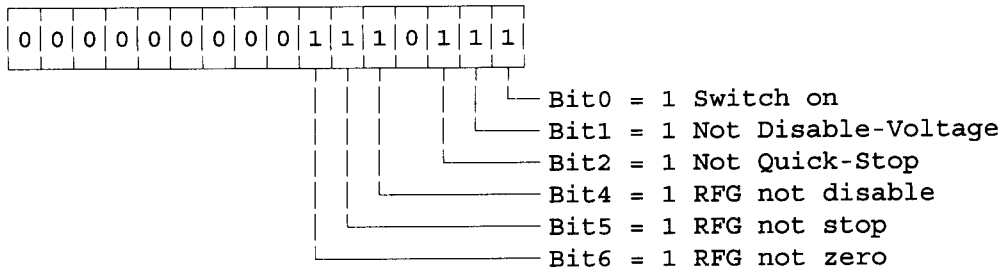
5 Setting of a positive *"Nominal-Speed-Value"*: of 1000 rpm



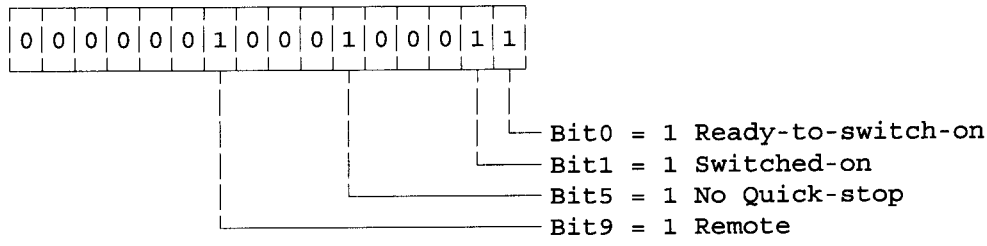
Because the values of the parameters *"Pole-Number" and "Gear-Factor"* are still at basic adjustment, the unit of the Nominal-Speed-Value is "rpm". (1000 = 3E8 hex)

Reaction of FDS: The motor starts at the ramp Accel1. During that time the bit 10 of the Status-Word is zero. After reaching the Actual-Speed-Value the Nominal-Speed-Value the bit 10 changes to 1. In the Actual-Speed-Value you can always see the actual value of speed.

6 Setting the Device Control Command "Disable operation"



Reaction of FDS: The motor will slow down at the ramp Decel1 and by reaching the Actual-Speed-Value zero the state changes from Operation-enabled to Switched-on. The display reads:
" 0 rpm 0 A" / "enabled ".
After that is the answer of FDS in *"Status-Word"*:

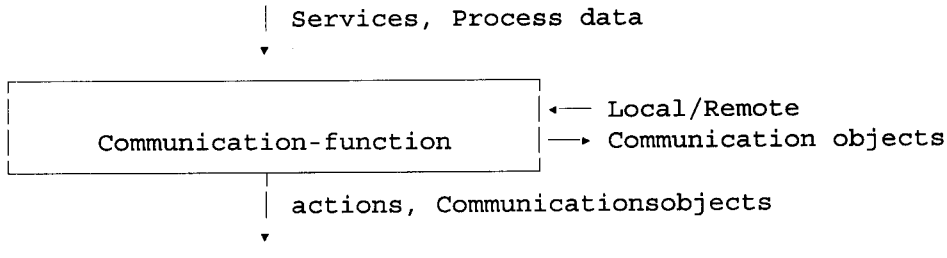


6 Parametrization of FDS with Interbus-S

The Parametrization of FDS with Interbus-S is done, as defined in DRIVECOM-Pofile No. 21. In that profile the functions of PCP-Communication of Interbus-S are used. You can find a detailed description of that functions in "Referencemanual for Interbus-S Peripherals Communication Protocol" from Phönix Contact . In the following subchapters these functions of communication are described, that are used in DRIVECOM-Profile.

6.1 Communication-function

The Communication-function contents of several Subfunctions:
- Execution of services
- Control of communication
- mapping of process data to the communication objets
- Process data monitoring
- Communication monitoring



Services

The FDS represents a server, that executes the received service-requests. Following services are executed by the communication function in the drive contoler.

Initiate (establish connection)
After a positive answer of the initiate service, all following services are done by the FDS

Read (read communication-objects)
The contents of parameters are read out with this service.

Write (write communication-objects)
The contents of parameters are written with this service.

Status (read device status)
The drive controllers's status is read out with this service.

Identify (read manufacturer's name, type and version)
With this service the manufacturer's name, type and the version of Software are read out.

Get-OV (read of the objet descriptions)
The object descriptions of the drive controller's communications objects are read out with this service..

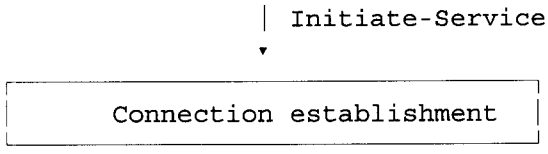
Abort (abort connection)
The connection establlished with the initiate service is aborted. From now on, all following services are rejected by the communication layer.

Process data
Process data consists of the data transferred trough the process data channel and mapped to communication objects. Process data is mapped to communication objects by means of the "Process-Data-Control" function.

Communication-objects
With the aid of services, communication objects can be manipulated through the bus. The communication objects are mapped to the real parameters in the drive controller. The data pertaining to the communication objects can be transferred by means of services or process data. All comminication objects accessible by means of the Read and Write services are described in the object dictionary.

6.1.1 Connection Establishment

This function is executed when a request to establish a connection (Initiate-Service indication) is send over the bus to the FDS. Then the connection will be established.



Initiate-Service
The Initiate-Service is the request to establish a connection. Following Service-parameters will be set from the FDS with values and send back to the client.

Version OV
By this SW-Version the value 0013 hex will be shown. By every update of the software this value will be incremented.

Profile-Nummer
The value is 0021 hex, so as defined in DRIVECOM profile No. 21

Access-Protection-Supported
This service parameter specifies whether or not access protection is realized. For FDS this value is true.

Password
The Value of this service parameter is 0 hex.

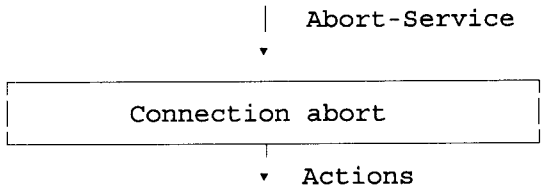
Access-Group
In this profile the value of access-group is 0 hex.

Error message
By FDS no error messages are possible here

Additional-Code
no

6.1.2 Connection Abort

This function is called when an Abort service is send over the bus to the FDS.

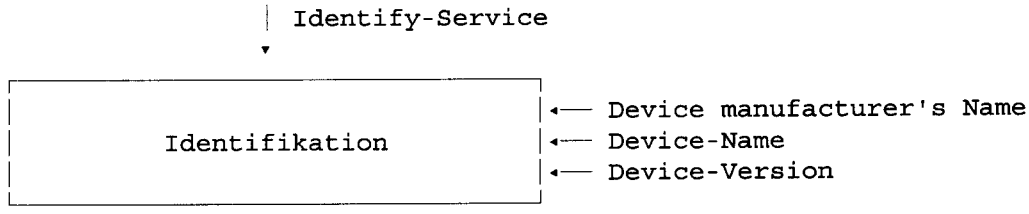


Aktions
- From this moment no more services were worked from FDS. The process data are still running.
- see chapter 6.3.3

Error message
no

6.1.3 Identification

For recognizing which devices are connected to the Interbus-S the Identify-Service is used.



Then the FDS is sending following data to the client:

Device manufacturer's name
here: "STÖBER".

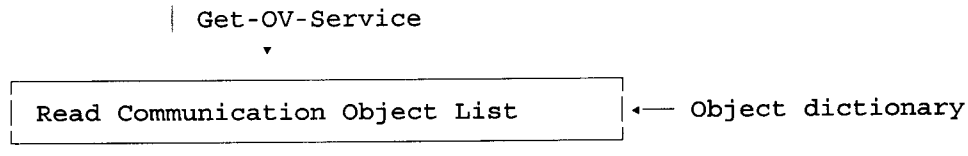
Device-Name
here appears the kind of the FDS:
"FDS 1040", "FDS 1040/B", ..."FDS 1200" or "FDS1200/B".

Device-Version
This parameter contains the number of software-version from "Base-Device-FDS" and from the Optional board Interbus-S: z.B.: "D3.1_4 /SV3.1_4". In case of softwareupdates other numbers appears.

Error message
no

6.1.4 Read Communication Object List

All descriptions of the device parameters can be read out with this function.



Get-OV-Service
The object dictionay is read out with this service.

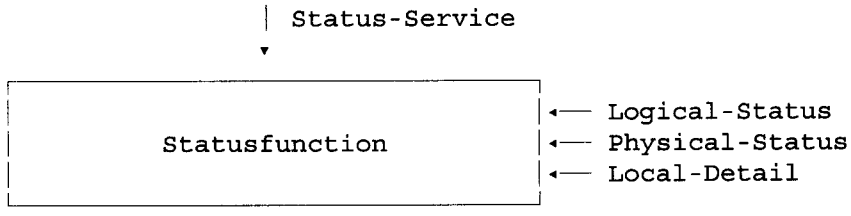
Objekt dictionary
The object dictionary contains the description of all communication objects.

Error message
The service parameter *"Error-Type"* is composed of the following parameters:
- *"Error-Class"*
- *"Error-Code"*
- *"Additional-Code"* *This parameter is here always zero*

Error-Class	Error-Code	Meaning
5 Service	5 Illegal-Parameter	The service parameter *"Access-Specification"* has an invalid value
6 Access	7 Object-Non-Existent	No object exists under this index
6 Access	8 Type Conflict	The data of the *"Name"* are not of the Visible-String data type

6.1.5 Status Function

The Status function block interrogates the status of the device.



Logical-Status
This parameter defines the capability of the drive device.

	Logical-Status	Device status
0	Ready to communicate	Remote
2	Limited number of services	Local

Physical-Status
This parameter provides a rough overviews of the device's operating state.

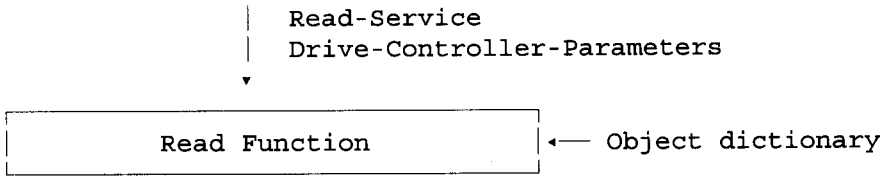
	Physical-Status	Status of the device control
0	Ready	Operation enabled
1	Partly ready	all other states
2	Not ready	Not ready to switch on
3	Maintenache required	not used

Local-Detail
In the bits 0 ... 15 of this paramter are mapped the bits of the *Status-Word"*

Error message
no

6.1.6 Read Function

Read access to communication objects is realized with this function. The data communicated when using the Read service is read out of the corresponding drive prameters. The assignment of drive controller parameters to communication objects can be fount in the object dictionary.



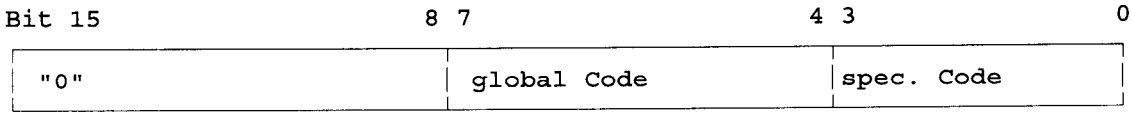
Read-Service
The Read-Service communicates the data to be read out of a communication object.

Drive Conroller-Paramters
The data communicated by means of the Read service is read out of the Drive-Controller-Parameters.

Error messages
The service parameter *"Error-Type"* is composed of the following parameters:
- *"Error-Class"*
- *"Error-Code"*
- *"Additional-Code"*

Error-Class	Error-Code	Meaning
6 Access	3 Object-Access-Denied	The client's access rights are not adequate
6 Access	5 Object-Attribut-Inconsistent	A service parameter has assumed an invalid value angenommen. See the reason in "Additional Code"
6 Access	6 Object-Access-Unsupported	The object is not a Variable Access Object
6 Access	7 Object-Non-Existent	No object exists under this index
8 Other	0 Other	The service has not been executed. See the reason in "Additional-Code"

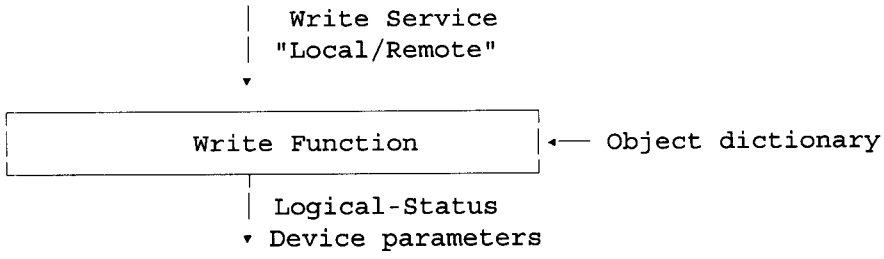
Additional-Code
2 octets with following format:



Code [hex] glob.	spez.	Meaning
0	0	no precise details of the reason for the error
1	0	Service-Parameter with an invalid value
1	1	Subindex does not exist
2	0	Service can currently executed because of the Device status

6.1.7 Write Function

Write acces to communication objects is executed with this function. The data communicated in the Write service is written to the corresponding drive controller parameters. The assignment of the drive controller parameters to communication objects can be found in the object dictionary.



Write-Service
The Write Service communicates the data to be written to a communication object. The length of the data is been checked with the corresponding parameters in the object dictionary from the FDS. The service is acknowledged negatively if the data length do not agree.

Local / Remote
When the Local / Remote signal is set ot local, each write access by means of a Write service to a communication object is acknowledged negatively.

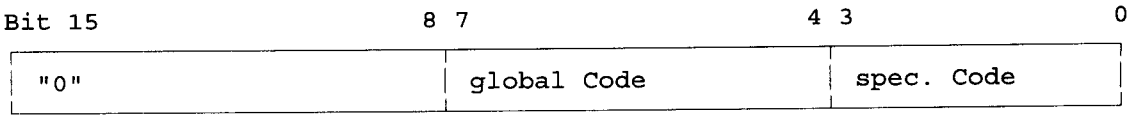
Logical-Status
This parameter is set from the FDS to Limited-Number-of-Services" value when the internal Local / Remote signal is set to local.

Device-Controller-Parameters
The data communicated with the Write service is written to the device parameters.

Error message
The service parameter *"Error-Type"* is composed of the following parameters:
- *"Error-Class"*
- *"Error-Code"*
- *"Additional-Code"*

Error-Class	Error-Code	Meaning
6 Access	3 Object-Access-Denied	The client's access rights are not adequate
6 Access	5 Object-Attribut-Inconsistent	A service parameter has assumed an invalid value The reason is specified in Additional Code
6 Access	6 Object-Access-Unsupported	The object is not an Variable-Access-Object
6 Access	7 Object-Non-Existent	No object exists under this index
6 Access	8 Type-Conflict	The data does not correspond to the data type of the object
8 Other	0 Other	The service has not been executed. The reason is specified Additional-Code

Additional-Code
2 octets mit following format:



Code [hex] glob.	spez.	Meaning
0	0	no precise details of the reason for an error
1	0	service parameter with an invalid value
1	1	Sub-Index not available
1	2	Length of service parameter *"Data"* too high
1	3	Length of service parameter *"Data"* too low
2	1	Service can currently not be executed because of local-Control
2	2	Service can currently not be executed because of Device Status
3	0	Value range exceeded
3	1	Value of parameter too high
3	2	Value of parameter too low
3	4	Value of a subparameters too high
3	5	Value of a subparameters too low
3	6	max. Value less than min. Value
4	0	Collision with other values
4	7	General parameter collision reason

Note:
After sending the positve Acknowledge of a Write service back to the client, it continues up to 256 ms, til the new values became an effect in the device. Because this, it is possible that a read service executed just after a write service can send the old data.

6.1.8 Acknowledging Parameter Accesses

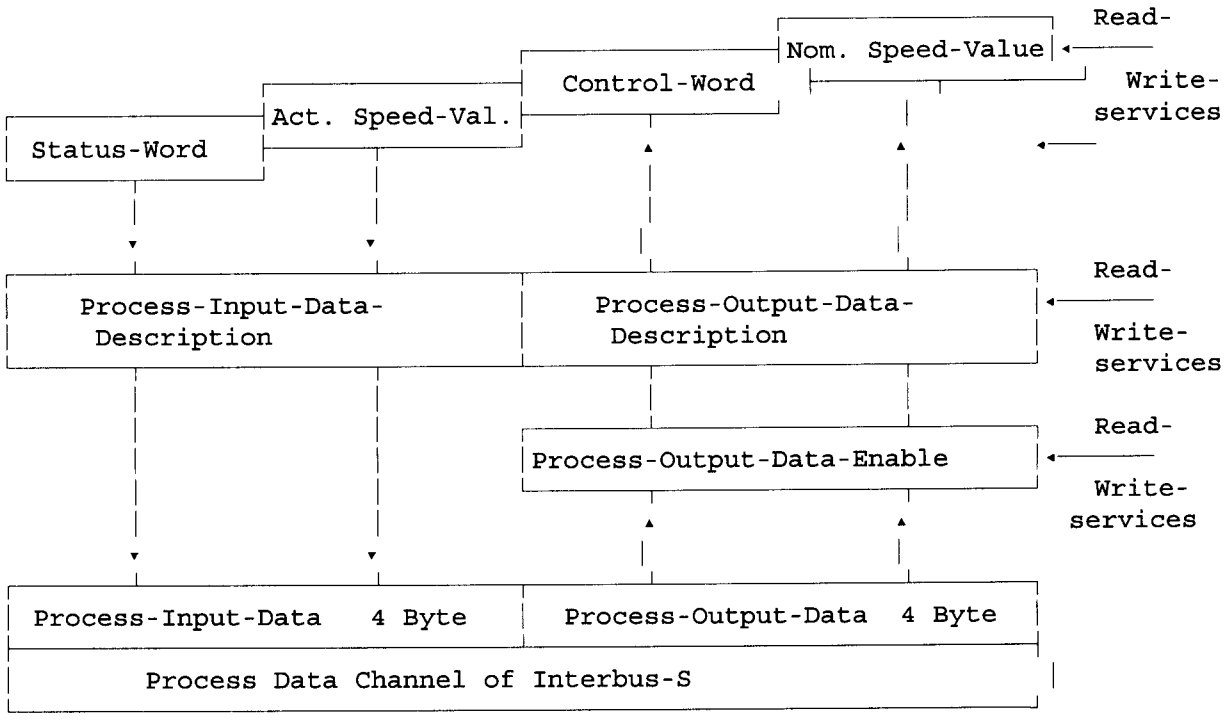
Each access to parameters through the PCP-Communiation channel is acknowledged. Access to inexistent prameters or fromal incorrectly access (subindex or datalength) is acknowledged negatively. When specified, parameters that can be mapped to the process data channel must always be acknowledged positively. These may, however be limited by specified limit values during the further course of processing. However, this does not modify the parameter itself.
By the FDS there are only the parameters *"Nominal-Speed-Value"* and *"Control-Word"*, which can not be acknowledged negatively.

Example: *"Speed-Max-Amount"* : 1500 rpm
"Nominal-Speed-Value" : 3000 rpm

These specified values are acknowledged positively, but the Nominal-Speed-Value is limited to 1500 rpm during execution. Reading back the Nominal-Speed-Value parameter returns a value of 3000 rpm.

6.2 Process Data Control

The process data transferred through the process data channel is mapped to the communication objects by means of this function. By FDS the length of the process data channel is fixed to 4 Byte.
Several parameters works to this function. That is shown in following picture:



Process-Output-Data-Enable

This paramter controls the Copy-Function of the Process-Outpun-Data to the Device-Paramter. All values from 0 to 255 can be written in this parameter. Only when the value 255 is written into the parameter, then the Process-Output-Data will be copied from the FDS to the device paramters Control-Word and Nominal-Speed-Value. The basic adjustment of this paramter is 255. By this value the writing a value to the above named paramters with a Write-service has no effect, because with every Interbus-S-Cycle new data from the process data channel appears.

Process-Input-Data-Description

This parameter discribes, which communication objects (Device parameters) are mapped to the Process-Input-Data. Here by the FDS with Interbus-S Communication is this mapping fixed to Status-Word and Actual-Speed-Value. This can not be modified.

The structure of the Process-Input-Data-Description by the FDS is following:

Nr.	Content	Meaning
1.	4	Length of Process-Input-Data-Channel in byte
2.	6041hex	Index for 1. byte: here *"Status-Word"*, that has a length of 2 bytes
3.	0	Subindex for 1. byte: *"Status-Word"* has no subindex, also 0
4.	0	Index for 2. byte: here 0 because this byte is reserved with *"Status-Word"*
5.	0	Subindex for 2. byte: here 0 because this byte is reserved
6.	6044hex	Index for 3. byte: here *"Actual-Speed-Value"*, that has a length of 2 bytes
7.	0	Subindex for 3. byte: *"Actual-Speed-Value"* has no subindex
8.	0	Index for 4. byte: here 0, because this byte is reserved from *"Actual-Speed-Value"*
9.	0	Subindex for 4. byte: here 0 because this byte is reserved

Process-Output-Data-Description

This parameter contains the data defining the communication objects to which the process output data is mapped. By the FDS with Option Interbus-S Communication that are *"Control-Word"* and *"Actual-Speed-Value"*. This mapping can not be modified by the FDS.

The structure of the Process-Output-Data-Description by the FDS is following:

Nr.	Content	Meaning
1.	4	Length of Process-Output-Data-Channel in byte
2.	6040hex	Index for 1. byte: here *"Control-Word"*, that has a length of 2 bytes
3.	0	Subindex for 1. byte: *"Control-Word"* has no subindex, also 0
4.	0	Index for 2. byte: here 0 because this byte is reserved with *"Control-Word"*
5.	0	Subindex for 2. byte: here 0 because this byte is reserved
6.	6042hex	Index for 3. byte: here *"Nominal-Speed-Value"*, that has a length of 2 bytes
7.	0	Subindex for 3. byte: *"Nominal-Speed-Value"* has no subindex
8.	0	Index for 4. byte: here 0, because this byte is reserved from *"Nominal-Speed-Value"*
9.	0	Subindex for 4. byte: here 0 because this byte is reserved

6.3 Check Functions

The Check Functions contains important tools for triggering defined actions by any faults from the Interbus-S, the SPS or faults from the application-programmer.
For the difference possibilities of faults exists difference parameters in which one of several actions can be triggered.

After every switch off of the voltage supply of the Optional board Interbus-S this pararmeters lose there actual values. So all parameters of the Check Functions have to been written with the correct values after every switch on of the voltage supply of the Optional board Interbus-S.

6.3.1 Process Data Monitoring

Data transfer through the process data channel is monitored by means of process data monitoring. A parameter-defined action is triggered if data transfer is longer than the set monitoring time:

Process-Data-Monitoring-Time

This parameter defines the time that may elapse maximal before new process data is transferred through the process data channel. The monitoring time values are specified in milliseconds. The value 65535 deactivates process data monitoring.

Value range:	100ms ... 65535 ms
Basic adjustment:	65535 (deactivate) (after switch on the voltage supply of Optional board)

Process-Data-Monitoring-Option-Code

This parameter defines the action that is triggered when no new process data is transferred within the monitoring time.

Option-Code	Meaning of the action
0	no action
1	Malfunction
2	Device Control Command "Disable Voltage"
3	Device Control Command "Quick Stop"

The basic adjustment is 0. (after switch on the voltage supply of Optional board Interbus-S)

6.3.2 Communication Monitoring

Data transfer through the communication channel is monitored by communication monitoring. A parameter-definable action is triggered if data transfer is incative for longer than the set monitoring time. This can only be happend during the established connection.

Communications-Monitoring-Time

This parameter defines the time that may elapse maximally before new data is transfered through the communication channel. The monitoring time values are specified in milliseconds. The value 65535 deactivates communication monitoring. This time should be adapted to the bus transfer time because one service through the communication needs several buscycles in the Interbus-S!

Value range: 100ms ... 65535 ms
Basic adjustment: 65535 (deactivate) (after switch on the voltage supply of Optional board)

Communication-Monitoring-Option-Code

This parameter defines the action that is triggered when no data is transferred through the communication within the monitoing time.

Option-Code	Meaning of the action
0	no action
1	Malfunction
2	Device Control Command "Disable Voltage"
3	Device Control Command "Quick Stop"

The basic adjustment is 0. (after switch on the voltage supply of Optional board Interbus-S)

6.3.3 Connection Abort Option Code

The parameter Connection-Abort-Option-Code defines the action, which is triggered, when the Connection Abort Serveice is received from the FDS.

Option-Code	Meaning of the action
0	no action
1	Malfunction
2	Device Control Command "Disable Voltage"
3	Device Control Command "Quick Stop"

The basic adjustment is 0. (after switch on the voltage supply of Optional board Interbus-S)

Description of that actions

By all the above named functions of monitoring exists the same actions. The effect of this actions will here detailed described:

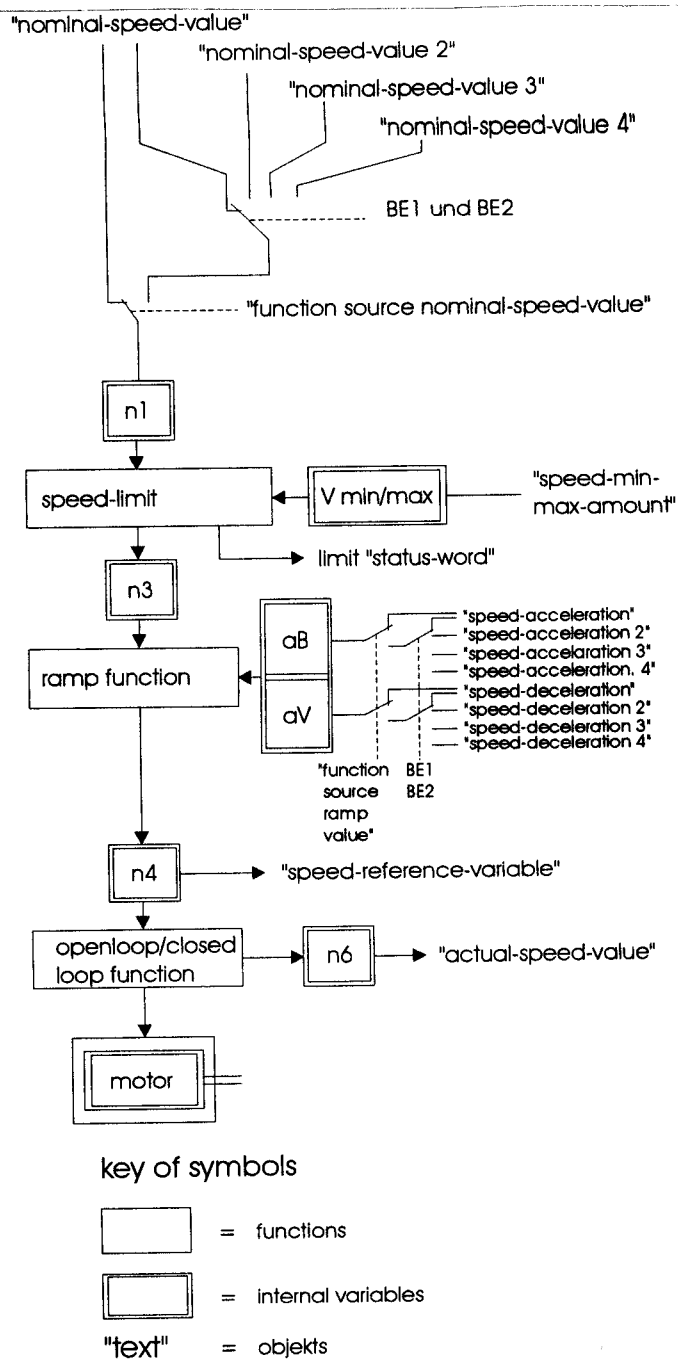
Start Status	Option-Code	Reaction	transit.
Ready-To-Switch-On	Disable-Voltage	-> Switch-On-Disabled	7
Ready-To-Switch-On	Quick-Stop	-> Switch-On-Disabled	7
Switched-On	Disable-Voltage	-> Switch-On-Disabled	10
Switched-On	Quick-Stop	-> Switch-On-Disabled	10
Operation-Enabled	Disable-Voltage	-> Switch-On-Disabled	9
Operation-Enabled	Quick-Stop	-> Quick-Stop-Active	11

Important Hints:

By the FDS 1040/B ... 1085/B works this monitoring functions only, when the Optional board Interbus-S is always supplied with its voltage supply. This Voltage (230V at X6) may not been switched off! Otherwise the drive is running with its last Nominal-Speed-Value an Control-Word further! In such a case the drive can only be stopped by disconnect the H-Level of binaray Input "Enable" or the power supply for the FDS.
This hint is not valid for the FDS 1110/B ... 1200/B.

6.4 Drive Function

The Drive Function has the same meaning as the Speed Function by the FDS.
Structure of the Speed Function of the FDS:



The Speed Function is composed of the folowing subfunctions:
- Speed limit
- Ramp function
- Closed-loop / open-loop control function

The Speed Function can be influenced by the following parameters:
- *"Nominal-Speed-Value"*
- *"Speed-Min-Max-Amount"*
- *"Speed-Acceleration (2, ..., 4)"*
- *"Speed-Deceleration (2, ..., 4)"*

The Speed Function returns the following parameters:
- *"Speed-Reference-Variable"*
- *"Actual-Speed-Value"*

General agreements for the Speed Function

For all signed communication objects, the positive value denotes rotation clockwise (to the right) and the negative value denotes rotation counter clockwise (to the left), when viewing the drive shaft.

The minimum or maximum values (e.g. -32768 and +32767 for integer 16) are output if the value of an object exceeds the value range that can be represented.

6.4.1 Nominal-Speed-Value
Nominal-Speed-Value 2
Nominal-Speed-Value 3
Nominal-Speed-Value 4

This parameters represents the setting values for the speed of the drive. The value ranges from -32768 to 32767 (integer 16), the unit is rpm.
This parameters are mapped internal in the FDS to the FDS-parameters "Fix. Ref. Values 1, -2, -3, -4". You can observe the values at that place in the display of FDS. These parameters can be saved non-volatile with the action "Save Values".
The selecting of the active Speed Value for the drive is influenced by the parameter "Source-Nominal-Speed-Value" and by the binary Inputs BE1 and BE2. Is after the switch on of the Voltage supply that parameter not modified, then the active Speed Value will be mapped from the parameter "Nominal-Speed-Value" which is transfered through the process data. This is the standard parameter defined from DRIVECOM:

If one of the bits "Disable-RFG", "Stop-RFG", or "RFG-zero" is zero, the parameter "Nominal-Speed-Value" will not more be mapped to the FDS-parameter "Fix. Ref. Value 1".

Error message no (formal correct services would not be acknowledged negatively)

6.4.2 Reference-Speed-Variable

The "Referecne-Speed-Variable" is the speed provided by the ramp function. The value ranges from -32676 to +32767. The parameter can only be read. The unit is rpm.

Error message no

6.4.3 Actual-Speed-Value

The Actual-Speed-Value is the speed on the motor spindle or load scaled in rpm. Depending on the system of a frequency inverter like FDS, speed deviations may occur between the Actual-Speed-Value and the physical speed. The value ranges from -32768 to 32767. The parameter is read-only.

Error message no

6.4.4 Speed-Min-Max-Amount

The parameter *"Speed-Min-Max-Amount"* consits of the subparameters *"Speed-Min-Amount"* und *"Speed-Max-Amount"* .
These subparameters have the value range from 0 to 4 294 967 295 (Unsigned 32). The unit is rpm.
This paramters sets the speed limits for the clockwise and counter clockwise rotation.

The value range is definde by the following rules:
- *"Speed-Min-Amount" < = "Speed-Max-Amount"*
- The frequency for the motor, calculated from the "Speed-Max-Amount" and the paramters "Pole-Number" and "Gear-Factor" may not exceed 150 Hz.

For calculating the maximum value the following formula is valid:

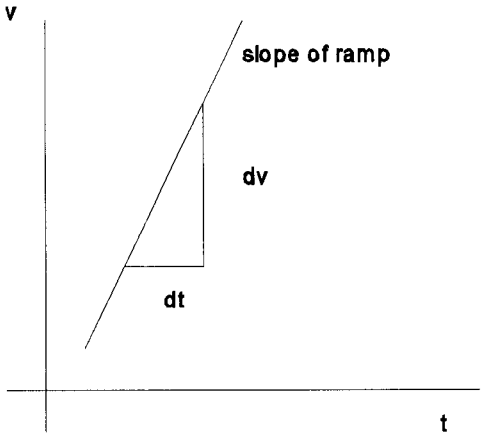
$$\text{maximum of Min-Amount, Max-Amount} = \text{minimum of } \frac{fmax * 60}{P/2 * i} \text{ and } 32767$$

With: i = 1 (*"Gear-Factor"* = 1 = basic adjustment in FDS)
p = 4 (*"Pole-Number"* = 4 = basic adjustment in FDS)
fmax = 150 Hz
results a maximal Value for parameter "Speed-Max-Amount" of 4500 rpm.

6.4.5 Speed-Acceleration
Speed-Acceleration 2
Speed-Acceleration 3
Speed-Acceleration 4

These parameters defines the slope of the Acceleration-ramps. They are generated as the quotient of *"Delta Speed"* and *"Delta Time"* subparameters.

$$\text{Speed-Acceleration (aB)} = \frac{\text{Delta Speed}}{\text{Delta Time}}$$



Delta Speed
Delta Speed has the unit rpm. This subparameter has the value range from 0 to 4 294 967 295 (Unsigned 32)

Delta Time
This subparameter is specifies in seconds and has the value range from 0 to 65 535 (Unsigned 16)

The value range of the complete acceleration consists of the two subparameters is defined with following rule:
- After the calculating of the quotient with the parameters "Pole-Number" and "Gear-Factor" into the FDS-internal unit the result may not leave the range from 0,1 to 300 sec/150Hz.

The selecting of the active acceleration ramp value for the drive is influenced by the parameter "Source-Ramp-Value" and by the binary inputs BE1 and BE2. Is after the switch on of the Voltage supply that parameter not modified, then the active acceleration ramp will be mapped from the parameter "Speed-Acceleration". This is the standard parameter which is defined from DRIVECOM.

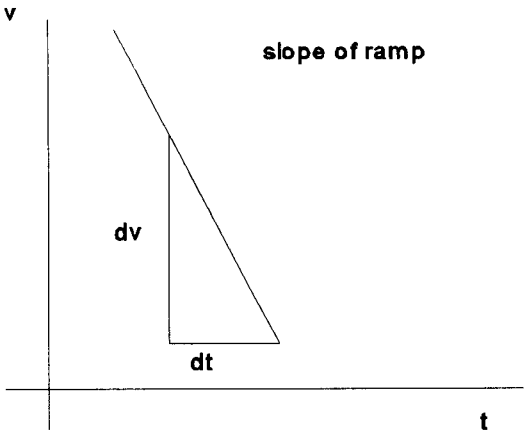
This parameters are mapped internal in the FDS to the FDS-parameters "Accel 1, -2, -3, -4". You can observe the values at that place in the display of FDS. These parameters can be saved with the action "Save Values".

Error message See Read- and Write Function

6.4.6 Speed-Deceleration
Speed-Deceleration 2
Speed-Deceleration 3
Speed-Deceleration 4

These parameters defines the slope of the Deceleration-ramps. They are generated as the quotient of *"Delta Speed"* and *"Delta Time"* subparameters.

$$\text{Speed-Deceleration (aV)} = \frac{\text{Delta Speed}}{\text{Delta Time}}$$



Delta Speed

Delta Speed has the unit rpm. This subparameter has the value range from 0 to 4 294 967 295

Delta Time

This subparameter is specifies in seconds and has the value range from 0 to 65 535

The value ranges of the Deceleration-ramps are the same as by the Acceleration-ramps.
The selecting of the active deceleration ramp value for the drive is influenced by the parameter "Source-Ramp-Value" and by the binary Inputs BE1 and BE2. Is after the switch on of the voltage supply that parameter not modified, then the active deceleration ramp will be mapped from the parameter "Speed-Deceleration".
This parameters are mapped internal in the FDS to the FDS-parameters "Decel 1, -2, -3, -4". You can observe the values at that plase in the display of FDS. These parameters can be saved with the action "Save Values".

Error message See Read- and Write Function

Calculating the ramps from FDS-uint into the DRIVECOM-unit

Definition of ramp by FDS $\dfrac{\Delta t_{FDS}}{150\ Hz}$ (start up time for 150Hz in seconds)

Definition of ramp by Drivecom-Profil: $\dfrac{\Delta n}{\Delta t_{IBS}}$ (in rpm per seconds)

Formula: $n = \dfrac{f * 60}{p/2 * i}$ (with p = Pole-Number, i = Gear-Factor)

$$\frac{\Delta n}{\Delta t_{IBS}} = \frac{\Delta f}{\Delta t_{FDS}} * \frac{60}{p/2 * i}$$

Formula for calculating the DRIVCOM-Values by given FDS-Values:

$$a = \frac{\Delta n}{\Delta t_{IBS}} = \frac{150Hz * 60}{p/2 * i * \Delta t_{FDS}}$$

Formula for calculating the FDS-Values by given DRIVECOM-Values:

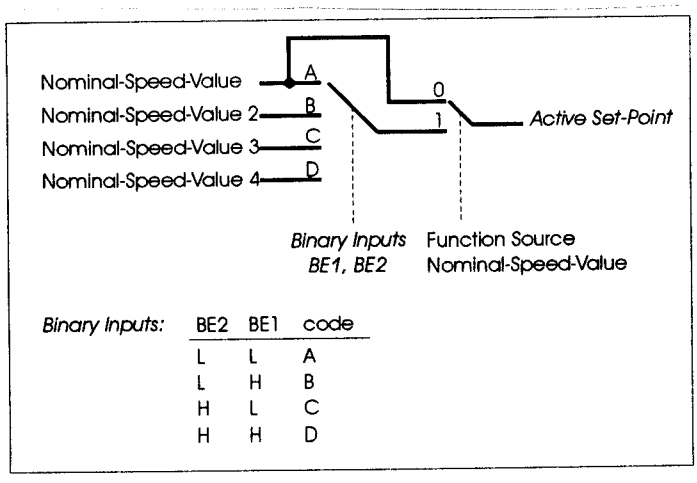
$$\Delta t_{FDS} = \frac{150Hz * 60 * \Delta t_{IBS}}{p/2 * i * \Delta n}$$

Value ranges:	FDS-Values:	[0,1 ... 3 ... 300] s/150Hz	
	DRIVECOM-Values:	Delta Speed:	[981000 ... <u>98100000</u> ... 4294967195]
		Delta Time:	[2180 ...<u>65400</u> ... 65535]

Note: It shoult be read always the whole parameter, because intern in the FDS both subparameters will be calculated new by every Read- and Write-Service.

6.4.7 Function Source Nominal Speed Value

Usually in an application, which use drives with the DRIVECOM Profile, the drive will be run with the parameter "Nominal-Speed-Value" . This parameter is defined in the DRIVECOM Profile and is transferred through the process data. If you want to select the other Nominal-Speed-Values (2, 3, 4), you have to set the parameter Source-Nominal-Speed-Value.

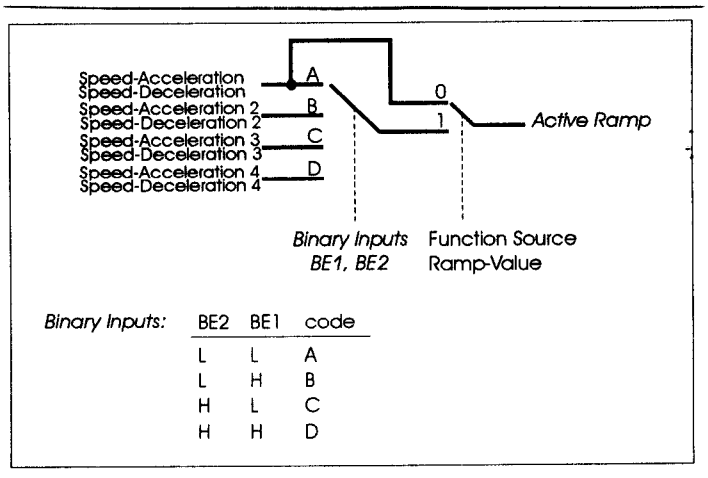


In that case, that the value of this parameter is set to 0, the FDS works so, as defined in DRIVCOM Profile. This is the basic adjustment of the parameter. If you set the value to 1, the one of the Nominal-Speed-Values is selectet. This selecting depends of the binary Inputs BE1 and BE2. (see picture)
After every "Power-On" of the FDS this parameter is set to its basic adjustment and has to be set new, if wanted.
This paremeter is manufactorer specific.

Error message See Read- and Write function

6.4.8 Function Source Ramp Value

Usually in an application, which use drives with the DRIVECOM Profile, the drive will be run with the parameters "Speed-Acceleration" and "Speed-Deceleration". This parameters are defined in the DRIVECOM Profile. If you want to select the other Accelerations and Decelerations (2, 3, 4), you have to set the parameter Source-Ramp-Value.



If this parameter is set to 0, the FDS works so, as defined in DRIVCOM Profile. This is the basic adjustment of the parameter. If you set the value to 1, one of the other accelerations and decelerations is selected. This selecting depends of the binary Inputs BE1 and BE2. (see picture)
After every "Power-On" of the FDS this parameter is set to its basic adjustment and has to be set new, if wanted.
This paremeter is manufactorer specific.

Error message See Read- and Write function

6.5 Motor Data Function

This function is used to adjust the connected motor to the FDS so that the Speed Function works as above described. This function is composed of the parameters U/f-Characteristic-Form, U/f-Characteristic-Curve, I-Motor, I-Max, Pole-Number and Gear-Factor. All these parameters are mapped in the FDS-internal parameters in the groups motor data and machine data. They can be saved non-volatile with the action "Save Values", which can be startet with the keys beneth the display.

6.5.1 U/f Characteristic Form

With this paramter the form of the U/f Curve can be selected:
- linear (0),
- square (1).

Other values are not possible.
This paremeter is manufactorer specific. It can only be modified in the Device status Switch-On-Disabled.

Error message See Read- and Write function

6.5.2 U/f Characteristic Curve

This parameter consits of the following subparameters:

1. subparameter : *"Boost"*
2. subparameter : *"Start-Frequency"*
3. subparameter : *"Nominal-Voltage"*
4. subparameter : *"Nominal-Frequency"*

The unit of **subparameter *"Boost"*** is Volt. The value ranges from 0 V to 110 Volt. This maximal value represents the Boost in FDS-unit of 100%.

The unit of **subparameter *"Start-Frequency"*** is 0,1 Hz. This value has to be set to 0. Other values an not possible. This subparameter represents that frequency, where the subparameter Boost is valid.

The **subparameter *"Nominal-Voltage"*** has the unit Volt. Only the value of 400 V is possible by FDS.

The **subparameter *"Nominal-Frequency"*** has the unit 0,1 Hz. The value ranges from 10 Hz to 400 Hz.

This paremeter is manufactorer specific. It can only be modified in the Device status Switch-On-Disabled.

Error message See Read- and Write function

6.5.3 I Motor

The parameters I-Motor and I-Max together defines the Current limitation. The parameter I-Motor should be set with the value of the nominal motor current of the connected motor. The unit is 0,01 A.

Value range:
- minimum value 0
- maximum value depends on the type of the FDS. (see the Installation and start-up instructions for FDS)

During the Write-Service to the parameter I-Motor the FDS checks also the parameter I-Max to its value range. So it can be neccessary to modify I-Max at first before writing an new value to I-Motor.
This paremeter is manufactorer specific.

Error message See Read- and Write function

6.5.4 I-Max

The parameter I-Max defines the value of motor current that is possible for maximal 30 seconds. The value ranges from 100 % of I-Motor to 200% of I-Motor. The absolute maximum value of I-Max is 150% of the nominal current of the FDS. (see the Installation and start-up instructions for FDS)

Error message See Read- and Write function

6.5.5 Pole-Number

This parameter is neccessary to adjust the connectet motor to the FDS. The correct number of poles has to be written in this parameter, otherwise the unit of all speed-paramters is not correctly rpm. This paramter has no unit. This is *an optional* parameter in the DRIVECOM Profile
Value Range: 2, 4, 6, ... 16
The Write-Service to this paramter will be acknowledged negatively, if an internal Variable of the FDS which are calculated from the parameters "Speed-Min-Max-Amount", "Speed-Acceleration, (2 ,....4)" and "Speed-Deceleration, (2, ... 4)" exceeds the frequency of 150 Hz.
This parameter can only be modified in the Device status Switch-On-Disabled.

Error message See Read- and Write function

6.5.6 Gear-Factor

This parameter is neccessary to adjust the connectet motor with gear to the FDS. This parameter has the same meaning as the FDS-Parameter "Gear-Factor" In the display of FDS the factor is shown like 0,1 ... 999,9. Therefor here is the unit of the "Gear-Factor" 0,1.
The correct factor of the gear has to be written in this parameter, otherwise the unit of all speed-paramters is not correctly rpm.
This is a manufactorer specific parameter.

Value range: 1 to 9999
The Write-Service to this paramter will be acknowledged negatively, if an internal Variable of the FDS which are calculated from the parameters "Speed-Min-Max-Amount", "Speed-Acceleration (2, ... 4)" and "Speed-Deceleration (2, ... 4)" exceeds the frequency of 150 Hz.
This parameter can only be modified in the Device status Switch-On-Disabled.

Error message See Read- and Write function

6.6 Device Description

Device description data is stored by means of this function. The following data is stored: Parameter-Set-Identifier, Code-Number. Both parameters can be read and written.

6.6.1 Parameter-Set-Identifier

This parameter serves to identify the currently active drive controler parameter set. It can be observed in the display of FDS in the group "display value" under "device info". The parameter can be saved with the action "Save values".

0	The parameter set belonging to the FDS has not been initialized through the bus.
1..254	The parameter set belonging to the device has been initialized through the bus. It has been assigned the freely selectable code number 1 ... 254 as its identifier.
255	The device was or is switched to local mode and there is no guarantee that the parameter set previously loaded is unchanged.

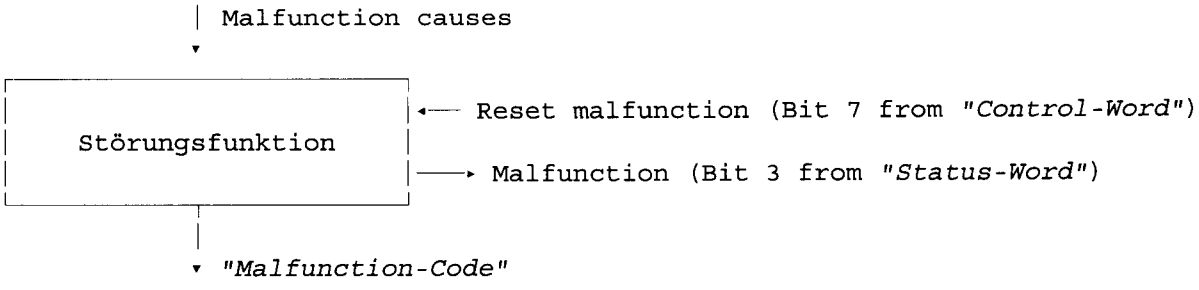
Error message See Read- and Write function

6.6.2 Code-Number

By means of this function, the user can store a code number in the FDS (non-volatile). The value can bee chosen freely from 0 to 65535. Therefore, in certain circumstances, any number may occur multiply on the bus. By appropiately assigning code numbers on the bus, exchange of a device or swapping of bus terminals, for example, can be recognized.

Error message See Read- and Write function

6.7 Fault Function



The fault function manages the *"Malfunction-Code"* parameter. As the result of FDS-malfunction, this parameter is set to the corresponding value (see list below).

6.7.1 Malfunction-Code

The *"Malfunction-Code"* is a String with the length of 2 Bytes. It is coded hierarchically, ranges from a coarse distinction to one that becomes increasingly finer.

Bit	Gruppierung
15 ... 12	Main groups
11 ... 8	Subgroups
7 ... 0	Details

The parameter is assigned a value unequal to zero if the FDS is in the malfunction state. The parameter is assigned the value 0 if the drive controller is not in the malfunction state.
If there is precisely one cause of a malfunction, the value assigned to its cause in the Malfunction-Code parameter can be read out unchanged until the malfunction state no longer applies. This is the case whenever the cause of the malfunction has been remedied and the Reset-Malfunction command has been issued.

Are there several causes of a malfunction, the value assigned to one of this causes. If only this cause has been remedied and the Reset-Malfunction command has been issued, so the Device status Malfunction is not terminated because of the other causes. Then one of the other causes is assigned to the parameter Malfunction-Code.

The Malfunction-Code is a read-only parameter.

Code (hex)	Meaning
0000	no malfunction
2310	Continuous overcurrent (Overload)
2320	Short circuit / short to earth
3210	Overvoltage
3220	Undervoltage
4210	Device overtemperature
4310	Motor overtemperature
5000	Device Hardware
5112	24V power supply
6010	Watchdog (3 s ausschalten)
7120	Motor not connected
8000	n-monitor or ref. val. wire break
8100	Communication

6.8 Display Function

The Display Function supplies two manufactory specific parameters.

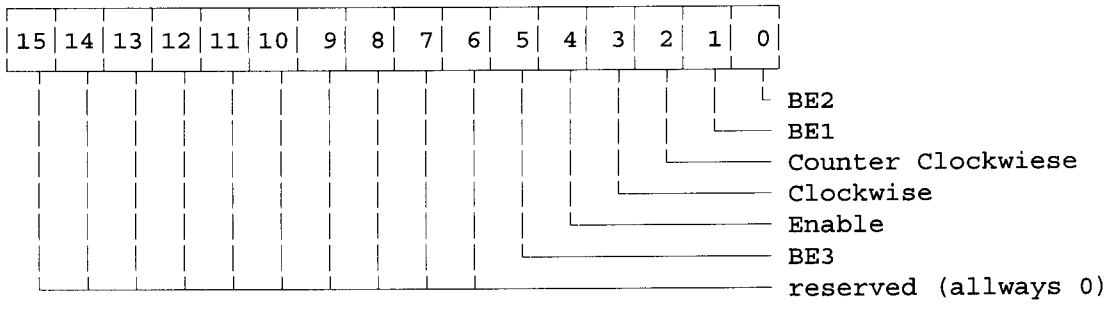
6.8.1 Motor-Current

The parameter *"Motort-Current"* assumes the value of the actual current between FDS and Motor. This value can observed in the display of FDS under Standard Display.
The unit of this parameter is 0,01A. It a read-only parameter.

6.8.2 Binary-Inputs

With this parameter, it is possible to observe the current input level of all binary inputs of the device FDS. Every binary input is represented by one bit. This parameter is read-only.

Mapping of the bits:



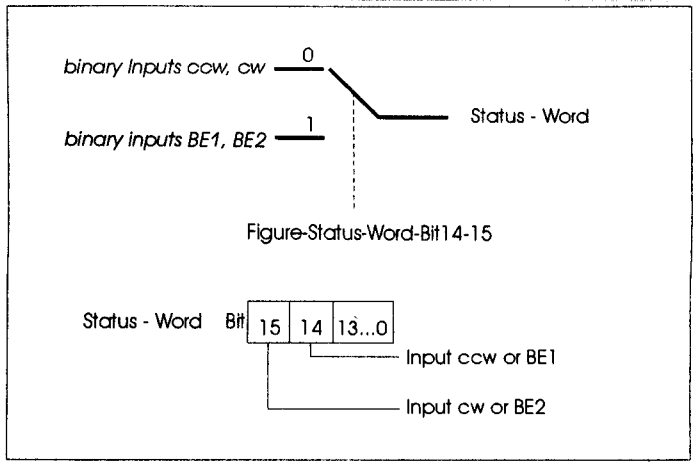
Meaning of the bits:
1 H-Level at binary input
0 L-Level "

6.8.3 Mapping-Status-Word-Bit14-15

Information about the status of the device and messages is indicated in the Status-Word. A complete description of all bits is found in chapter 6.1.5.

The bits 14 and 15 nomally shows the levels of the binary inputs Clockwise and Counter clockwise:
Bit 14 Level at binary Input Counter clockwise
Bit 15 Level at binary Input Clockwise

It is possible to map other sinals to that two bits of the Status-Word. That can be done with the parameter Mapping-Status-Word-Bit14-15. After every switch on of power supply of the FDS the value of this parameter is set to its basic adjustment (Clockwise / Counter clockwise). To use a other mapping as this basic adjustment it is neccessary to write the correct value every time after a power on of the FDS.



Error message See Read- and Write-Function

6.9 Compare of FDS-Parameters with Drivecom-Parameters

All neccessary adjustments for the operation of the FDS under control of Interbus-S are possible to set by the implemented parameters.

This parameters (communication objects) corresponds to the parameters of the Standard-FDS which are descibed in the "Installation and start-up instruction for FDS". But here some parameters have other names as known from the display of FDS, because of the mandatory definitions from the DRIVECOM-Profile.

In the display of Standard-FDS exists several parameters that are not neccessary for the correct operation of FDS under control of Interbus-S. This parameters can not be accessed through the Interbus-S.

The table below shows all parameters, displays and actions, that are known of the Standard-FDS and the corrresponding communications objects for the access through Interbus-S.

All the parameters, which can be find under Group, Parameter / Action and also under Object-Name for Access through Interbus-S, can be saved non-volatile in the FDS with the Action "Save Values".

Not all parameters of Standard-FDS can be accessed through FDS. That parameters have no Objet-Name in the table.

Some parameters of Standard-FDS have no signification for the control under Interbus-S. They are marked with "no signification by IBS".

Other parameters can not be accessed through Interbus-S, but the adjustment that has been done with the keys and display are used by operation the FDS under control of Interbus-S. They are marked in the table with "used".

Group	Parameter/Action	Objektname(s) by Access throug Inter-bus-S	Notes
motor data	poles	Pole-Number	
	type, boost	U/f-Charac.-Curve	
	U/f-character	U/f-Charac.-Form	
	current limit	I-Motor, I-Max	I-hold is used
	nominal speed	----------	used
	motor overtemp.	----------	used
	whisper automatic	----------	used
machine data	gear factor	Gear-Factor	
	sense of rotation	----------	not possible through IBS
	speed limits	Speed-Min-Max-Amount	
	start-upmode	----------	not possible through IBS
	n-blance 1 ... 4	----------	used
	load start	----------	not possible throught IBS
setting ref. val	all parameters	----------	no signification by IBS
fix ref. value	fix ref.-values 1 .. 4	Nominal-Speed-Value " " 2,3,4	
ramp generator	ramp shape	----------	used
	ramp set 1 ... 4	Speed-Acceleration, Speed-Decelration, Speed-Accel. 2..4 Speed-Decel. 2..4	
stop functions	quick stop	----------	not possible through IBS
	n-monitor	----------	used
	auto-start	----------	no signification by IBS
	auto-enable	----------	no signification by IBS
cntrl.-interface	input BE 3	----------	nicht möglich über IBS
	analog-output	----------	used
	relay-output 2	----------	used
display values	device info, device values, cntrl.-interface	Parameter-Set-Ideti-fier, Code-Number, Motor-Current, Binary-Inputs	rest not possible through IBS
operation	all parameters	----------	not possible through IBS
aktions	all aktions	----------	not possible through IBS
device data	all parameters	----------	not possible through IBS
	fault acknowldg.	----------	no signification by IBS

6.10 Not implemented DRIVECOM-parameters

The parameters which are defined in DRIVECOM-Profile, are devided in two groups: mandatory and optional paramters: The mandatory parameters has to be implemented in all devices that works with the DRIVECOM-Profile. The optional parameters only may be implemented. Because that, many parameters has been defined which have nearly the same function. So it was not neccessary to implement all the DRIVECOM-parameters. For that parameters which are not implemented, DRIVECOM defines Substitute-Values. This parameters and there substitute values has to been list in the documentation of the devices which operates with the DRIVECOM-Profile.

By the FDS all important parameters for a correct operation has been implemented. For a easier work with the profile here the not implemented parameters of DRIVECOM-Profile are desribed.

Factor Function

This function mulitplies for example the Nominal-Speed-Value with the Dimension-Factor and the Face-Value-Factor to an internal variable with the unit Rotation per minutes. For both factors the Substitue-Value is 1. So results for the parameters Nominal-Speed-Value, Actual-Speed-Value and Speed-Reference-Value the fixed unit of rpm. Another effect of not implementing of the Factor-Function is that also the paremeter Manipulated-Speed-Variable is not implemented.

Percentage Function

This function makes it possible to reference the nominal-, actual- and reference-values to the parameter Speed-Reference-Value. This function is not implemented in FDS. So the parameters Speed-Reference-Value, Nominal-Percentage, Percentage-Reference-Variable, Actual-Percentage and Percentage-Manipulated-Variable does not exist in FDS. Here no Substitute-Values are defined.

Speed Motor Limit

The limitation of speed for protecting the motor is done by the FDS only with the parameter Speed-Min-Max-Amount. Therefor the parameters Speed-Min-Max, Speed-Motor-Min-Max-Amount, Speed-Motor-Min-Max, Frequency-Motor-Min-Max-Amount, Frequency-Motor-Min-Max are not implemented. Substitute-Values are not neccessary.

Ramp Function

By the FDS the Ramp Function is controled by the parameters Speed-Acceleration (2 ... 4) and Speed-Decelration (2 ... 4). The parameters Speed-Quick-Stop, Ramp-Function-Time, Slow-Down-Time and Quick-Stop-Time does not exist. Substitute-Values are not defined.

Drive Disabling Function

The parameter Shutdown-Option-Code is not implemented. It desribes the action, which is performed, when the device status changes from Operation-Enabled to Ready-To-Swtich-On. The Substitute-Value is zero. That means the Drive-Function is disabled. The power section is switched off. This action is also desribed in the device state machine in transition No. 8.
The parameter Disabling-Operation-Option-Code is not implemented. It defines the action of the transition No. 5 in the device state machine. The Substitue-Value is 1. That means, that the drive will slow down on the Speed-Deceleration ramp. The power section is still switched on.

Stop Function

The parameters Stop-Option-Code and Quick-Stop-Option-Code are not implemented in FDS. For both parameters the Substitute-Value of 1 is valid. That means that the drive will slow down on Speed-Deceleration ramp if the corresponging control commands are send to FDS.

7 Mapping of Device Function to Communication

The parameters, which can be accessed through Interbus-S, are mapped from objects from the PCP-Communication. The sum of all objects that are supported from a device are listed in the "Static Object Dictionary". This dictionary describes all informations für PCP-Communication like index, typ, ...
The "type" of an object means one of the data type, which are supported by the PCP-Communication. The sum of that data types are listed in the "Static Type Dictionary".

All implemented parameters and data types of FDS with Option Interbus-S Communication will be descibed in the following tables.

Note:
A part of the communication objects are manufacturer specific defined. They are an addition to the objects defined in the DRIVECOM Profile. The indicies of this objects are lower than 6000 (hex). It is possible that other manufacturers has implemented at that indices other manufacturer specific objects.
Therefor it is very important to identify the device at first. Only if the device is recognized as a **FDS xxxx** from **Stöber Antriebstechnik** it is guarannteed, that the correct parameters exist at the corresponding indecies

For differentiation of these characteristics the objects in the tables are marked with a letter under "Objectclass".

m	=	mandatory
o	=	optional in DRIVECOM-Profile
h	=	manufacturer specific implemeted in FDS

7.1 Static Type Dictionary

Index (hex)	Object	Meaning	Objectclass
1	Data type	Boolean	o
2	Data type	Integer8	o
3	Data type	Integer16	m
4	Data type	Integer31	o
5	Data type	Unsigned8	o
6	Data type	Unsigned16	o
7	Data type	Unsigned32	m
8 - 9	Null object		o
A	Octet-String	Octet-String	m
1B - 20	Null object		o
21	Type Struc	Process data description structure	o
22	Type Struc	Ramp	m

Boolean:

The TRUE or FALSE is represented in one octet

Notation:	Boolean
Value range:	TRUE or FALSE
Coding:	FALSE is represented by the value 0, TRUE is represented by the value FF (hex).

Representation of the TRUE value at the Communication interface:

	MSB							LSB
Byte in addr= n	1	1	1	1	1	1	1	1

Representation of the FALSE value at the Communication interface:

	MSB						LSB
Byte in addr= n	0	0	0	0	0	0	0	0

Integer:
Integer-Values are signed variables
Notation: Integer8, Integer16, Integer32
Value range:

data type	Value range	length
Integer8	-128 <= i <= 127	1 Octet
Integer16	-32786 <= i <= 32767	2 Octet
Integer32	-2^{31} <= i <= $2^{31}-1$	4 Octet

Coding: Two's complement notation
VZ = 0: positive numbers including zero
VZ = 1: negative numbers

Representation of an Integer8 at the Communication interface:

	MSB						LSB
Byte in addr= n	VZ	6	5	4	3	2	1	0
		2	2	2	2	2	2	2

Representation of an Integer16 at the Communication interface:

	MSB						LSB
Byte in addr= n	VZ	14	13	12	11	10	9	8
		2	2	2	2	2	2	2
Byte in addr= n+1	7	6	5	4	3	2	1	0
	2	2	2	2	2	2	2	2

Representation of an Integer32 at the Communication interface:

	MSB						LSB
Byte in addr= n	VZ	31	30	29	27	26	25	24
		2	2	2	2	2	2	2
Byte in addr= n+1	23	22	21	20	19	18	17	16
	2	2	2	2	2	2	2	2
Byte in addr= n+2	15	14	13	12	11	10	9	8
	2	2	2	2	2	2	2	2
Byte in addr= n+3	7	6	5	4	3	2	1	0
	2	2	2	2	2	2	2	2

Object description: Integer8

Index	2 hex = Integer8
Object-Code	5 hex = data type
Description	Integer8

Object description: Integer16

Index	3 hex	= Integer16
Object-Code	5 hex	= data type
Description	Integer16	

Object description: Integer32

Index	4 hex	= Integer32
Object-Code	5 hex	= data type
Description	Integer32	

Object description Unsigned:
Unsigned values are variables that have no sign.
Notation: Unsigned8, Unsigned16, Unsigned32
Value range:

data type	Value range	length
Unsigned8	0 <= i <= 255	1 Octet
Unsigned16	0 <= i <= 65535	2 Octet
Unsigned32	0 <= i <= 4294967295	4 Octet

Coding: Binary

Representation of an Unsigned8 at the Communication interface:

	MSB						LSB
Byte in addr= n	7 2	6 2	5 2	4 2	3 2	2 2	1 2	0 2

Representation of an Unsigned16 at the Communication interface:

	MSB							LSB
Byte in addr= n	15 2	14 2	13 2	12 2	11 2	10 2	9 2	8 2
Byte in addr= n+1	7 2	6 2	5 2	4 2	3 2	2 2	1 2	0 2

Representation of an Unsigned32 at the Communication interface:

	MSB							LSB
Byte in addr= n	32 2	31 2	30 2	29 2	27 2	26 2	25 2	24 2
Byte in addr= n+1	23 2	22 2	21 2	20 2	19 2	18 2	17 2	16 2
Byte in addr= n+2	15 2	14 2	13 2	12 2	11 2	10 2	9 2	8 2
Byte in addr= n+3	7 2	6 2	5 2	4 2	3 2	2 2	1 2	0 2

Object description: Unsigned8

Index	5 hex	= Unsigned8
Object-Code	5 hex	= data type
Description	Unsigned8	

Object description: Unsigned16

Index	6 hex = Unsigned16
Object-Code	5 hex = data type
Description	Unsigned16

Object description: Unsigned32

Index	7 hex = Unsigned32
Object-Code	5 hex = data type
Description	Unsigned32

Object description Octet-String:

Representation of Strings with variable length.
Notation: Octet-String
Coding: binär

Representation of an Octet-Strings at the Communication interface:

	MSB						LSB
Byte in addr= n	first Octet						
Byte in addr= n+1	second Octet						
Byte in addr= n+2	usw.						
Byte in addr= n+3							

Object description: Octet-String

Index	A hex = Octet-String
Object-Code	5 hex = data type
Description	Octet-String

Object description: Process data description structure

The Process data description structure is composed of nine elements:

subindex	meaning	data type
1	Processdatalength	Unsigned8
2	1.index Process-In/Outdata	Unsigned16
3	1.subindex "	Unsigned8
4	2.index Process-In/Outdata	Unsigned16
5	2.subindex "	Unsigned8
6	3.index Process-In/Outdata	Unsigned16
7	3.subindex "	Unsigned8
8	4.index Process-In/Outdata	Unsigned16
9	4.subindex "	Unsigned8

Object description

Index	20 hex = PDD strucuture
Object-Code	6 hex = data type structure descr.
Description	PDD structure
Number-Of-Elements	9 hex
List of Element	
Data-Type-Index	5 hex = Unsigned8
Length	1 hex = 1 Byte
Data-Type-Index	6 hex = Unsigned16
Length	2 hex = 2 Byte
Data-Type-Index	5 hex = Unsigned8
Length	1 hex = 1 Byte
Data-Type-Index	6 hex = Unsigned16
Length	2 hex = 2 Byte
Data-Type-Index	5 hex = Unsigned8
Length	1 hex = 1 Byte
Data-Type-Index	6 hex = Unsigned16
Length	2 hex = 2 Byte
Data-Type-Index	5 hex = Unsigned8
Length	1 hex = 1 Byte
Data-Type-Index	6 hex = Unsigned16
Length	2 hex = 2 Byte
Data-Type-Index	5 hex = Unsigned8
Length	1 hex = 1 Byte

Object description Rampe:

The "Ramp" data type structure description contains two elements:

Subindex	Bedeutung	data type
1	Delta speed numerator	Unsigned32
2	Delta time denominator	Unsigned16

Object description: Ramp

Index	21 hex = Ramp
Object-Code	6 hex = data type structure
Description	Rampen-Struktur
Number-Of-Elements	2 hex
List of Element	
Data-Type-Index	7 hex = Unsigned32
Length	4 hex = 4 Byte
Data-Type-Index	6 hex = Unsigned16
Length	2 hex = 2 Byte

7.2 Static Object dictionary

Index	Type	Object	Name	Object-class
5FED	Rampe	Record	Speed-Deceleration4	h
5FEE	Rampe	Record	Speed-Acceleration4	h
5FEF	Rampe	Record	Speed-Deceleration3	h
5FF0	Rampe	Record	Speed-Acceleration3	h
5FF1	Rampe	Record	Speed-Deceleration2	h
5FF2	Rampe	Record	Speed-Acceleration2	h
5FF3	Integer16	Var	Nominal-Speed-Value4	h
5FF4	Integer16	Var	Nominal-Speed-Value3	h
5FF5	Integer16	Var	Nominal-Speed-Value2	h
5FF6	Usigned8	Var	Mapping-Status-Word-Bit14-15	h
5FF7	Usigned8	Var	Source-Ramp-Value	h
5FF8	Usigned8	Var	Source-Nominal-Speed-Value	h
5FF9	Usigned16	Var	Binary-Inputs	h
5FFA	Usigned16	Var	Motor-Current	h
5FFB	Usigned8	Var	U/f-Characteristic-Form	h
5FFC	Usigned16	Var	U/f-Characteristic-Curve	h
5FFD	Usigned16	Var	I-Max	h
5FFE	Usigned16	Var	I-Motor	h
5FFF	Unsigned16	Var	Gear-Factor	h
6000	PDD-Struc	Record	Process-Input-Data-Desription	o
6001	PDD-Struc	Record	Process-Output-Data-Desription	o
6002	Boolean	Var	Process-Output-Data-Enable	m
6003	Unsigned16	Var	Process-Data-Monitoring-Time	o
6004	Integer16	Var	Process-Data-Monitoring-Option-Code	o
6005	Unsigned16	Var	Communication-Monitoring-Time	o
6006	Integer16	Var	Communication-Monitoring-Option-Code	o
6007	Integer16	Var	Connection-Abort-Option-Code	o
6008	Unsigned16	Var	Code-Number	o
6009	Unsigned8	Var	Parameter-Set-Identifier	o
...				
603E		Null		m
603F	Octet-String	Var	Malfunction-Code	m
6040	Octet-String	Var	Control-Word	m
6041	Octet-String	Var	Status-Word	m
6042	Integer16	Var	Nominal-Speed-Value	m
6043	Integer16	Var	Speed-Reference-Variable	m
6044	Integer16	Var	Actual-Speed-Value	m
6045		Null		o
6046	Unsingned32	Var	Speed-Min-Max-Amount	m
6047		Null		o
6048	Ramp	Record	Speed-Acceleration	m
6049	Ramp	Record	Speed-Deceleration	m
604A		Null		o
604B		Null		o
604C		Null		o
604D	Unsigned8	Var	Pole-Number	m
...				o
605D		Null		o

Object description	
Index	5FED hex = **Speed-Deceleration4**
Variable-Name	= not available
Object-Code	09 hex = Record
Number-Of-Elements	02 hex = 2 elements
Data-Type-Index	16 hex = ramp
Password	00 hex = no Password
Access-Groups	00 hex = no access groups
Access-Rights	0003 hex = Read-All, Write-All
Local-address	= not used
Extension	= not available

Subindex	01 hex = Delta speed numerator
Data-Type-Index Length	04 hex = Integer32 04 hex = 4 Byte
Subindex	02 hex = Delta speed denominator
Data-Type-Index Length	03 hex = Integer16 02 hex = 2 Byte

Objectfunction	
Processdata- Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Object description	
Index	5FEE hex = **Speed-Acceleration4**
Variable-Name	= not available
Object-Code	09 hex = Record
Number-Of-Elements	02 hex = 2 elements
Data-Type-Index	21 hex = ramp
Password	00 hex = no Password
Access-Groups	00 hex = no access groups
Access-Rights	0003 hex = Read-All, Write-All
Local-address	= not used
Extension	= not available

Subindex	01 hex = Delta speed numerator
Data-Type-Index Length	04 hex = Integer32 04 hex = 4 byte
Subindex	02 hex = Delta time denominator
Data-Type-Index Length	03 hex = Integer16 02 hex = 2 byte

Objectfunction	
Processdata- Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Object description	
Index	5FEF hex = **Speed-Deceleration3**
Variable-Name	= not available
Object-Code	09 hex = Record
Number-Of-Elements	02 hex = 2 elements
Data-Type-Index	21 hex = ramp
Password	00 hex = no Password
Access-Groups	00 hex = no access groups
Access-Rights	0003 hex = Read-All, Write-All
Local-address	= not used
Extension	= not available

Subindex	01 hex = Delta speed numerator
Data-Type-Index Length	04 hex = Integer32 04 hex = 4 byte
Subindex	02 hex = Delta time denominator
Data-Type-Index Length	03 hex = Integer16 02 hex = 2 byte

Objectfunction	
Processdata- Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Object description	
Index	5FF0 hex = **Speed-Acceleration3**
Variable-Name	= not available
Object-Code	09 hex = Record
Number-Of-Elements	02 hex = 2 elements
Data-Type-Index	21 hex = ramp
Password	00 hex = no Password
Access-Groups	00 hex = no access groups
Access-Rights	0003 hex = Read-All, Write-All
Local-address	= not used
Extension	= not available

Subindex	01 hex = Delta speed numerator
Data-Type-Index Length	04 hex = Integer32 04 hex = 4 byte
Subindex	02 hex = Delta speed denominator
Data-Type-Index Length	03 hex = Integer16 02 hex = 2 byte

Objectfunction	
Processdata- Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Object description	
Index	5FF1 hex = **Speed-Deleration2**
Variable-Name	= not available
Object-Code	09 hex = Record
Number-Of-Elements	02 hex = 2 elements
Data-Type-Index	21 hex = ramp
Password	00 hex = no Password
Access-Groups	00 hex = no access groups
Access-Rights	0003 hex = Read-All, Write-All
Local-address	= not used
Extension	= not available

Subindex	01 hex = Delta speed numerator
Data-Type-Index Length	04 hex = Integer32 04 hex = 4 byte
Subindex	02 hex = Delta times denominator
Data-Type-Index Length	03 hex = Integer16 02 hex = 2 byte

Objectfunction	
Processdata- Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Object description	
Index Variable-Name Object-Code Number-Of-Elements	5FF2 hex = **Speed-Acceleration2** = not available 09 hex = Record 02 hex = 2 elements
Data-Type-Index	21 hex = ramp
Password Access-Groups Access-Rights	00 hex = no Password 00 hex = no access groups 0003 hex = Read-All, Write-All
Local-address Extension	= not used = not available

Subindex	01 hex = Delta speed numerator
Data-Type-Index Length	04 hex = Integer32 04 hex = 4 byte
Subindex	02 hex = Delta speed denominator
Data-Type-Index Length	03 hex = Integer16 02 hex = 2 byte

Objectfunction	
Processdata- Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Object description		
Index	5FF3 hex	= **Speed-Nominal-Value4**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	03 hex	= Integer16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Processdata-Mapping	not possible
Error-Codes by negative acknowledge.	cannot be acknowledged negatively

Objectdescription		
Index	5FF4 hex	= **Speed-Nominal-Value3**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	03 hex	= Integer16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Processdata-Mapping	not possible
Error-Codes by negative acknowledge.	cannot be acknowledged negatively

Objectdescription		
Index	5FF5 hex	= **Speed-Nominal-Value2**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	03 hex	= Integer16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Processdata-Mapping	not possible
Error-Codes by negative acknowledge.	cannot be acknowledged negatively

Objectdescription		
Index	5FF6 hex	= **Mapping-Status-Word-Bit14-15**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	05 hex	= Unsigned8
Length	01 hex	= 1 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Processdata-Mapping	not possible
Error-Codes by negative acknowledge.	see Read- or Write-Funktion

Objectdescription		
Index	5FF7 hex	= **Source-Ramp-Value**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	05 hex	= Unsigned8
Length	01 hex	= 1 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Processdata-Mapping	not possible
Error-Codes by negative acknowledge.	see Read- or Write-Funktion

Objectdescription		
Index	5FF8 hex	= **Source-Nominal-Speed-Value**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	05 hex	= Unsigned8
Length	01 hex	= 1 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Processdata-Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Funktion

Objectdescription		
Index	5FF9 hex	= **Binary-Inputs**
Variable-Name		= not available
Object-Code	08 hex	= Simple-Variable
Data-Type-Index	06 hex	= Unsigned16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0001 hex	= Read-All
Local-address		= not used
Extension		= not available

Objectfunction	
Objektclass	manufactuer specific
Processdata-Mapping	not possible
Error-Codes if negative acknowledge.	see Read-Function

Objectdescription		
Index	5FFA hex	= **Motor-Current**
Variable-Name		= not available
Object-Code	08 hex	= Simple-Variable
Data-Type-Index	06 hex	= Unsigned16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0001 hex	= Read-All
Local-address		= not used
Extension		= not available

Objectfunction	
Objektclass	manufactuer specific
Processdata-Mapping	not possible
Error-Codes if negative acknowledge.	see Read-Function

Objectdescription		
Index	5FFB hex	= **U/f-Characteristic-Form**
Variable-Name		= not available
Object-Code	08 hex	= Simple-Variable
Data-Type-Index	0A hex	= Unsigned8
Length	01 hex	= 1 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Objektclass	manufactuer specific
Processdata-Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Objectdescription		
Index	5FFC hex	= **U/f-Characteristic-Curve**
Variable-Name		= not available
Object-Code	08 hex	= Array
Number-of-Elements	04 hex	= 4 elements
Data-Type-Index	06 hex	= Unsigned16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Objektclass	manufactuer specific
Processdata-Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Objectdescription		
Index	5FFD hex	= **I-Max**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	06 hex	= Unsigned16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Objektclass	manufactuer specific
Processdata-Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Objectdescription		
Index	5FFE hex	= **I-Motor**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	06 hex	= Unsigned16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Objektclass	manufactuer specific
Processdata-Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Objectdescription		
Index	5FFF hex	= **Gear-Factor**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	06 hex	= Unsigned16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Objectfunction	
Objektclass	manufactuer specific
Processdata-Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Objectdescription		
Index	6000 hex	= **Process-Input-Data-Description**
Variable-Name		= not available
Object-Code	09 hex	= Record
Number-Of-Elements	09 hex	= 9 Elemente
Data-Type-Index	20 hex	= PDD-Struct
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-address		= not used
Extension		= not available

Subindex	01 hex = Processdatalength
Data -Type-Index Length	05 hex = Unsigned8 01 hex
Subindex	02 hex = 1.Index PI-Data
Data -Type-Index Length	06 hex = Unsigned16 02 hex
Subindex	02 hex = 1.Subindex PI-Data
Data -Type-Index Length	05 hex = Unsigned8 01 hex
Subindex	04 hex = 2.Index PI-Data
Data -Type-Index Length	06 hex = Unsigned16 02 hex
Subindex	05 hex = 2.Subindex PI-Data
Data -Type-Index Length	05 hex = Unsigned8 01 hex
Subindex	06 hex = 3.Index PI-Data
Data -Type-Index Length	06 hex = Unsigned16 02 hex
Subindex	07 hex = 3.Subindex PI-Data
Data -Type-Index Length	05 hex = Unsigned8 01 hex
Subindex	08 hex = 4.Index PI-Data
Data -Type-Index Length	06 hex = Unsigned16 02 hex
Subindex	09 hex = 4.Subindex PI-Data
Data -Type-Index Length	05 hex = Unsigned8 01 hex

Objectfunction	
ProcessData - Mapping	not possible
Error-Codes	see Read- or Write-Function

Objectdescription	
Index	6001 hex = **Process-Output-Data -** **Description**
Variable-Name	= not available
Object-Code	09 hex = Record
Number-Of-Elements	09 hex = 9 elements
Data -Type-Index	20 hex = PDD-Struc
Password	00 hex = no Password
Access-Groups	00 hex = no access groups
Access-Rights	0003 hex = Read-All, Write-All
Local-Address	= not used
Extension	= not available

Subindex	01 hex = Processdatalength
Data -Type-Index Length	05 hex = Unsigned8 01 hex
Subindex	02 hex = 1.Index PO-Data
Data -Type-Index Length	06 hex = Unsigned16 02 hex
Subindex	02 hex = 1.Subindex PO-Data
Data -Type-Index Length	05 hex = Unsigned8 01 hex
Subindex	04 hex = 2.Index PO-Data
Data -Type-Index Length	06 hex = Unsigned16 02 hex
Subindex	05 hex = 2.Subindex PO-Data
Data -Type-Index Length	05 hex = Unsigned8 01 hex
Subindex	06 hex = 3.Index PO-Data
Data -Type-Index Length	06 hex = Unsigned16 02 hex
Subindex	07 hex = 3.Subindex PO-Data
Data -Type-Index Length	05 hex = Unsigned8 01 hex
Subindex	08 hex = 4.Index PO-Data
Data -Type-Index Length	06 hex = Unsigned16 02 hex
Subindex	09 hex = 4.Subindex PO-Data
Data -Type-Index Length	05 hex = Unsigned8 01 hex

Objectfunction	
ProcessData - Mapping	not possible
Error-Codes	see Read- or Write-Function

Objectdescription		
Index	6002 hex	= **Process-Output-Data-Enable**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data -Type-Index Length	0A hex 01 hex	= Octet-String = 1 Byte
Password Access-Groups Access-Rights	00 hex 00 hex 0003 hex	= no Password = no access groups = Read-All, Write-All
Local-Address Extension		= not used = not available

Objectfunction	
Objektclass	mandatory
ProcessData - Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Objectdescription		
Index	6003 hex	= **Process-Data-Monitoring Time**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	06 hex	= Unsigned16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-Address Extension		= not used = not available

ObjektFunction	
Objektclass	optional
Processdata-Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Objectdescription		
Index	6004 hex	= **Process-Data-Monitoring-Option-Code**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	03 hex	= Integer16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-Address Extension		= not used = not available

ObjektFunction	
Objektclass	optional
Processdata-Mapping	not possible
Error-Codes if negative acknowledge.	see Read- or Write-Function

Objectdescription		
Index	6005 hex	= **Communication-Montoring Time**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	06 hex	= Unsigned16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-Address		= not used
Extension		= not available

ObjektFunction	
Objektclass	optional
Processdata-Mapping	not possible
Error-Codes if negative acknowledge	see Read- or Write-Function

Objectdescription		
Index	6006 hex	= **Communication-Monitoring-Option-Code**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	03 hex	= Integer16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-Address		= not used
Extension		= not available

ObjektFunction	
Objektclass	optional
Processdata-Mapping	not possible
Error-Codes if negative acknowledge	see Read- or Write-Function

Objectdescription		
Index	6007 hex	= **Connection-Abort-Option-Code**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	03 hex	= Integer16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-Address		= not used
Extension		= not available

ObjektFunction	
Objektclass	optional
Processdata-Mapping	not possible
Error-Codes if negative acknowledge	see Read- or Write-Function

Objectdescription		
Index	6008 hex	= **Code-Number**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	06 hex	= Unsigned16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-Address		= not used
Extension		= not available

ObjektFunction	
Processdata-Mapping	not possible
Error-Codes if negative acknowledge	see Read- or Write-Function

Objectdescription		
Index	6009 hex	= **Parameter-Set-Identifier**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	05 hex	= Unsigned8
Length	01 hex	= 1 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-Address		= not used
Extension		= not available

ObjektFunction	
Processdata-Mapping	not possible
Error-Codes if negative acknowledge	see Read- or Write-Function

Objectdescription		
Index	603F hex	= **Malfunction-Code**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	0A hex	= Octet-String
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0001 hex	= Read-All
Local-Address		= not used
Extension		= not available

ObjektFunction	
Processdata-Mapping	not possible
Error-Codes if negative acknowledge	see Read-Function

Objectdescription		
Index	6040 hex	= **Control-Word**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	0A hex	= Octet-String
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-Address		= not used
Extension		= not available

Valuedescription	
Bit-No.	meaning
0	Switch-On
1	Disable-Voltage
2	Quick-Stop
3	Enable-Operation
4	Disable RFG
5	Stop-RFG
6	RFG-zero
7	Reset-Malfuncion
8..15	Reserved

ObjektFunction	
Processdata-Mapping	PO-Data
Error-Codes if negative acknowledge	Can not be acknowledged negatively

Objectdescription		
Index	6041 hex	= **Status-Word**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	0A hex	= Octet-String
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0001 hex	= Read-All
Local-Address		= not used
Extension		= not available

Valuedescription

Bit-No.	meaning
0	Ready-To-Switch-On
1	Switched-On
2	Operation-Enabled
3	Malfunction
4	Voltage-Disabled
5	Quick-Stop
6	Switch-On-Disabled
7	Warning
8	Message
9	Remote
10	Face-Value-Reached
11	Limit-Value
12..15	Reserved

ObjektFunction

Processdata-Mapping	PI-Data
Error-Codes if negative acknowledge	can not be acknowledged negatively

Objectdescription

Index	6042 hex	= **Nominal-Speed-Value**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	03 hex	= Integer16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-Address		= not used
Extension		= not available

ObjektFunction

Processdata-Mapping	PA-Daten
Error-Codes if negative acknowledge	can not be acknowledged negatively

Objectdescription

Index	6043 hex	= **Speed-Reference-Variable**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	03 hex	= Integer16
Length	02 hex	= 2 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0001 hex	= Read-All
Local-Address		= not used
Extension		= not available

ObjektFunction

Processdata-Mapping	not possible
Error-Codes if negative acknowledge	see Read- or Write-Function

Objectdescription	
Index	6044 hex = **Actual-Speed-Value**
Variable-Name	= not available
Object-Code	07 hex = Simple-Variable
Data-Type-Index	03 hex = Integer16
Length	02 hex = 2 Byte
Password	00 hex = no Password
Access-Groups	00 hex = no access groups
Access-Rights	0001 hex = Read-All
Local-Address	= not used
Extension	= not available

ObjektFunction	
Processdata-Mapping	PI-Data
Error-Codes if negative acknowledge	can not be acknowledged negatively

Objectdescription	
Index	6046 hex = **Speed-Min-Max-Amount**
Variable-Name	= not available
Object-Code	08 hex = Array
Number-Of-Elements	02 hex = 2 elements
Data-Type-Index	07 hex = Unsigned32
Length	04 hex = 4 Byte
Password	00 hex = no Password
Access-Groups	00 hex = no access groups
Access-Rights	0003 hex = Read-All, Write-All
Local-Address	= not used
Extension	= not available

ObjektFunction	
Processdata-Mapping	not possible
Error-Codes if negative acknowledge	see Read- or Write-Function

Objectdescription	
Index	6048 hex = **Speed-Acceleration**
Variable-Name	= not available
Object-Code	09 hex = Record
Number-Of-Elements	02 hex = 2 elements
Data-Type-Index	21 hex = ramp
Password	00 hex = no Password
Access-Groups	00 hex = no access groups
Access-Rights	0003 hex = Read-All, Write-All
Local-Address	= not used
Extension	= not available

Subindex	01 hex = Delta speed numerator
Data-Type-Index	04 hex = Integer32
Length	04 hex = 4 Byte
Subindex	02 hex = Delta time denominator
Data-Type-Index	03 hex = Integer16
Length	02 hex = 2 Byte

ObjektFunction	
Processdata-Mapping	not possible
Error-Codes if negative acknowledge	See Read- or Write-Function

Objectdescription	
Index	6049 hex = **Speed-Deceleration**
Variable-Name	= not available
Object-Code	09 hex = Record
Number-Of-Elements	02 hex = 2 elements
Data-Type-Index	21 hex = ramp
Password	00 hex = no Password
Access-Groups	00 hex = no access groups
Access-Rights	0003 hex = Read-All, Write-All
Local-Address	= not used
Extension	= not available

Subindex	01 hex = Delta speed numerator
Data-Type-Index	04 hex = Integer32
Length	04 hex = 4 Byte
Subindex	02 hex = Delta time denominator
Data-Type-Index	03 hex = Integer16
Length	02 hex = 2 Byte

ObjektFunction	
Processdata-Mapping	not possible
Error-Codes	See Read- or Write-Function

Objectdescription		
Index	604D hex	= **Pole-Number**
Variable-Name		= not available
Object-Code	07 hex	= Simple-Variable
Data-Type-Index	05 hex	= Unsigned8
Length	01 hex	= 1 Byte
Password	00 hex	= no Password
Access-Groups	00 hex	= no access groups
Access-Rights	0003 hex	= Read-All, Write-All
Local-Address		= not used
Extension		= not available

ObjektFunction	
Objektklasse	optional
Processdata-Mapping	not possible
Error-Codes	see Read- or Write-Function