COMBICOM

CAN in AUTOMATION

GB INSTRUCTION MANUAL CANopen Interface

Mat.No.	Rev.
CCF50E0-K002	1B

GB

The pictograph used in this instruction manual has following meaning:

Attention, observe at all costs

There is a bibliography on page 77 in this instruction manual with reference books which explain certain standards and declarations of this manual. Marked digits with brackets [ ] are placed at the appropriate passages.

Table of Contents

1. General Information

The presented documentation as well as the herein mentioned hard- and software are developments of Karl E. Brinkmann GmbH. Errors and ommissions excepted! The Karl E. Brinkmann GmbH have prepared the documentation, hardware and software to the best of their knowledge, however, no guarantee is given that the specifications will provide the efficiency aimed at by the user. The Karl E. Brinkmann GmbH reserves the right to change the specifications without prior notification or further obligation.

This instruction manual describes the new version of the F5-CAN operator. With regard to the old version we refer you to the instruction manual CC.F5.010-K001.

2. Order Informations

This instruction manual:	CC.F5.0D0-K002
F5 CAN operator with display and keyboard:	00.F5.060-5010
F5 CAN operator without display and keyboard:	00.F5.060-5110
F5 CAN operator with display and keyboard (terminal strip):	00.F5.060-5011
Utilities for the diagnostic interface	
HSP5 cable between PC and adapter:	00.F5.0C0-0001
Adapter DSUB9 / Western:	00.F5.0C0-0002

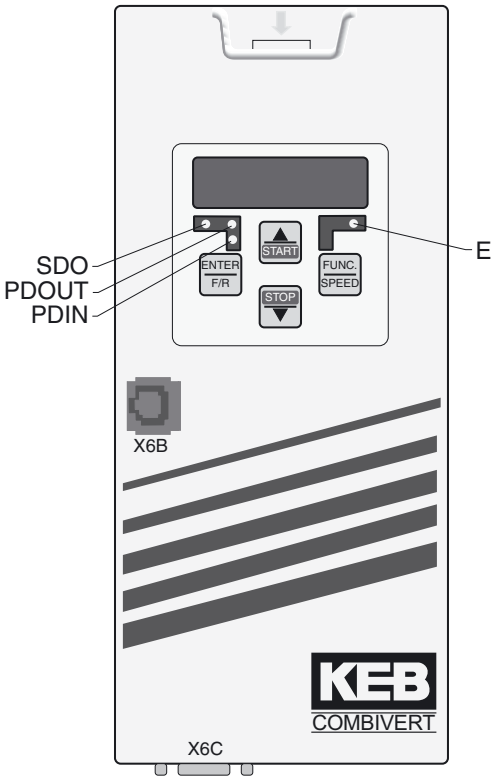
3. F5 CAN Operator

KEB-Antriebstechnik develop, produce and sell static frequency inverters worldwide in the industrial power range. The inverters of type F5 can be equipped optionally with a CAN (Controller-Area-Network) interface. It concerns an intelligent interface, that controls the access to the parameters of the frequency inverter via CAN.

The F5 CAN operator is integrated into the FI housing by simple plug-in and fits into all KEB F5 frequency inverters. Parallel to the field bus operation the operation over the integrated display/keyboard as well as another interface for diagnosis/parameterization (KEB COMBIVIS) is possible.

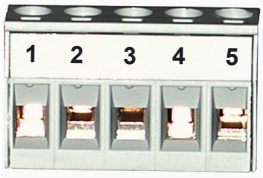
For programming the KEB F5 inverter by CAN the user requires in additon to this manual the instruction manual of the respective frequency inverter control [1].

4 Hardware Description



Keyboard and display only at: 00.F5.060-5010 and 00.F5.060-5011	
SDO (green):	SDO communication active
PDOUT (green):	PDOUT data are written to the FI control
PDIN (green):	• PDIN data are read by the FI control • Flashes shortly after reset command via CAN.
E (red):	on → Inverter ready for operation flashes → Inverter at fault off → No power supply
X6B:	Diagnostic interface to the PC (see chapter 4.1)
X6C:	CAN interface (socket-plug)

Alternativley available with teminal strip	
X6D:	CAN interface as 5pole terminal strip (optional available)



Pin assignment CANo:	
Pin	Signal
1	V- (reference potential for external power supply)*
2	CAN_L
3	Shield
4	CAN_H
5	V+ (external power supply) *

* not connected here

4.1 Diagnostic Interface

To prevent the desctruction of the PC interface, the diagnostic interface must be connected to the PC with a special HSP5 cable with voltage adaption only !

A HSP5 cable is connected via adapter to the diagnostic interface (see chapter 2 order designations). Access to all inverter parameters can be done with the PC software KEB COMBIVIS 5 .Also the operator internal parameters can be read out and partly adjusted or parameterized by download.

4.2 CAN interface

The CAN interface consists of a D-SUB-9pole pin connector (according to DIN41652 part 1). Assignment of the CAN connector according to [2]:

Pin	Signal	Description
1	-	reserved
2	**CAN_L**	**CAN bus signal dominant low**
3	CAN_GND	not connected here
4	-	reserved
5	(CAN_SHLD)	not connected here
6	(GND)	not connected here
7	**CAN_H**	**CAN bus signal dominant high**
8	-	reserved
9	(CAN_V+)	not connected here

Transmission level on CAN:	according to ISO/DIS 11898, ISO high speed
Transmission rate on CAN:	adjustable via CAN (10, 20, 25, 50, 100, 125, 250, 500, 800, 1000 Kbit/s)
Potential separation:	safe disconnection according to VDE0160.
Bus termination:	124 Ohm , must be made externally (between pin 2 and 7).

5. Basics of the CAN BUS

Here we like to introduce the system of the CAN (Controller-Area-Network) BUS and also explain some terms that are frequently used in the following.

The CAN is a multi master system, i.e. each user has access to the BUS and can send telegrams. In order to avoid invalid conditions during simultaneous access of two users, the CAN BUS knows a so-called arbitration phase, which defines the telegram beginning. In case of access conflicts all users recognize during this arbitration, who sends the lowest telegram number (identifier). Then this user can continue to send his telegram completely, without having to start from the beginning again. Now all other (willing-to-send) users pass over into the receiving status and abort their telegram for the time being. Thus it is specified that lower telegram numbers automatically have priority over higher numbers. The number of telegram numbers is limited to 2032 identifier (0...2031) at CAN version 2.0A.

CAN telegrams can contain a maximum of 8 byte user data.

The term logical CAN master used in the following, refers to the CAN user, who is responsible for the control of the entire CAN system. Even if there are physically only masters at CAN, in most applications there will be one or several users who exercise control. In this combination the KEB frequency inverter is considered as recipient of orders (logical slave).

6 Functions

The CAN protocol is uniformly standardized for the data backup layer. The processing of this protocol is taken over completely by a CAN controller. Furthermore, the CAN in Automation association (CiA) has passed a standard for the higher protocol layer that was named CAN Application Layer (CAL). Based on this standard the „CAL-based Communication Profile“ (CiA,DS301) was published in September 1995. This standard provides the basis for all CANopen unit profiles. In this standard a certain subset of the CAL standard is selected. The communication profile defines, among other things, a minimum capability device. That is the minimum required functionality, which a CANopen node must make available. The present CAN interface connection realizes such a minimum capability device.

An important point for every CAN network is the assignment of the telegram numbers (Identifier), especially since the numbers are limited to 2032 CAN V2.0 A. In the CAL standard an own procedure has been defined, which processes this assignment dynamically over an own protocol. This relative complex procedure for the assignment of identifiers is not mandatory for a minimum capabiltity device and is not integrated into the KEB CAN interface connection . For this case a more simpler procedure for the arrangement of the identifier assignment is defined in the communication profile. This procedure is also supported by the KEB CAN interface connection and looks as follows:

Each frequency inverter receives an explicit CAN address, the Node_Id. KEB provides two possibililties for the source of this Node_Id.

- If the value of the parameter OP_Node_Id has the value 255 (dec):
  → Node_Id = inverter address (SY.06) +1
- In all other cases the value of the parameter OP_Node_Id itself specifies the value of the Node_Id:
  → Node_Id = OP_Node_Id

On delivery all KEB frequency inverters have the inverter address = 1. In case several KEB frequency inverters shall be networked over CAN, they all must first receive different inverter addresses . This is done, e.g. via the keyboard of the operator.

Six identifiers are assigned to each frequency inverter.

Over an identifier any CAN node can request the reading or writing of a parameter value (Request-Identifier).

A further identifier is reserved for the appropriate response of the frequency inverter (Response-Identifier). The mechanism of request and response is also referred to as acknowledged service. The CANopen communication profile combines these functions under the term Service Data Object (SDO):

SDO(rx) = Request-Identifier = 1536+Node_Id
SDO(tx) = Response-Identifier = 1408+Node_Id

Example: Node_Id= 30 → Write/read requests over identifier = 1566(dec)
→ Write/read requests over identifier = 1438(dec)

Note: Basically the function of the SDO is completely sufficient to control the KEB F5 frequency inverter by CAN. Each parameter value in the inverter can be changed or inquired herewith.

Over the 3th identifier the CAN master can give the frequency inverter unaddressed and unacknowledged data. In dependence on the data direction from the master to the slave, it is referred to in the following as OUT1-Iden-

tifier.

Over the 4th identifier the frequency inverter transfers new data unadressed and unacknowledged to the CAN master (IN1-Identifier).

This function is called Process Data Object (PDO) by the communication profile. The two object parts are named PDO1(rx) and PDO1(tx).

PDO1(rx)	=	Out-Identifier	=	512 + Node_Id
PDO1(tx)	=	IN-Identifier	=	384 + Node_Id

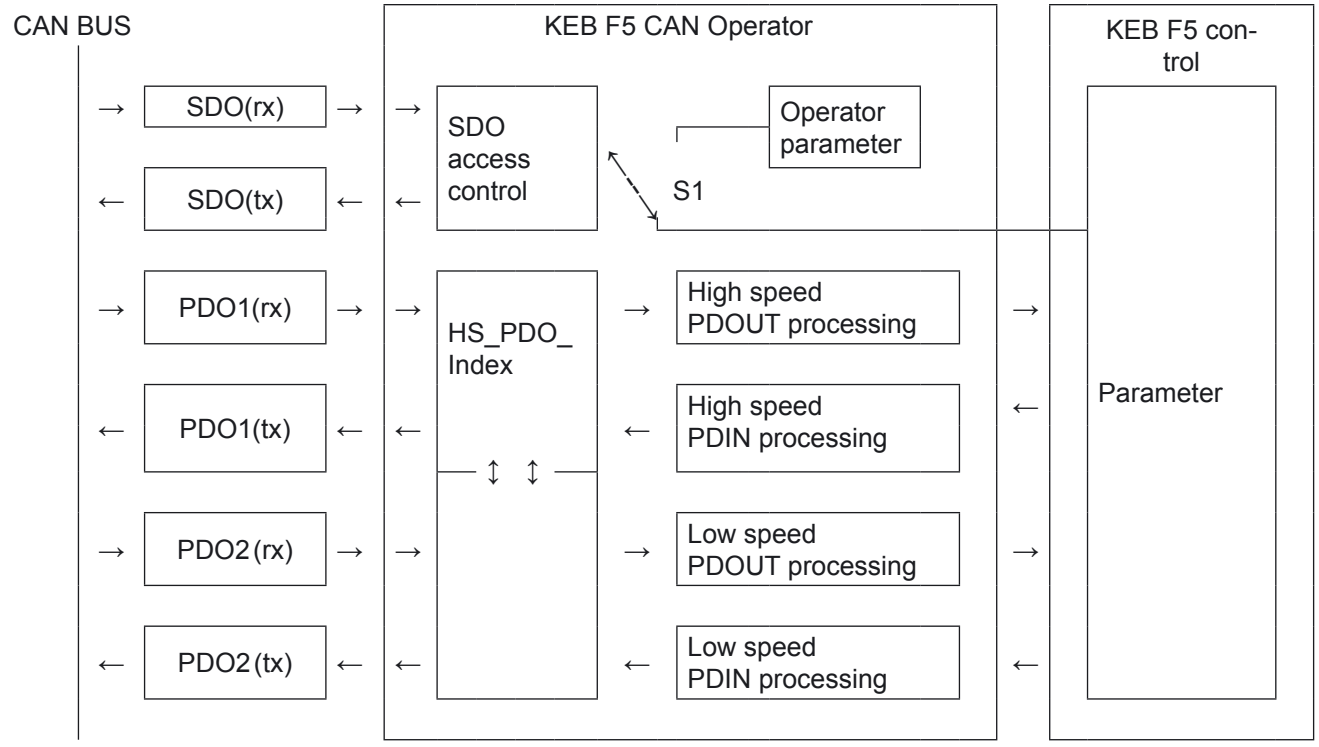
Starting with the Software version 1.3 the PDO functionality exists twice in the KEB F5 CAN interface connection. This so-called 2. PDO occupies the identifier five to six:

PDO2(rx)	=	Out-Identifier	=	768 + Node_Id
PDO2(tx)	=	IN-Identifier	=	640 + Node_Id

Over the 5th identifier the CAN master can give the frequency inverter unaddressed and unacknowledged data. In dependence on the data direction from the master to the slave, it is referred to in the following as OUT2-Identifier.

Over the 6th identifier the frequency inverter transfers new data unaddressed and unacknowledged to the CAN master (IN2-Identifier).

The two PDO's are with regard to the management identical but differ clearly in the form of internal processing. Only one of the two can be processed like in the previous software versions as high speed PDO. With regard to the processing the added PDO is on equal terms with the SDO commands and is referred to as low speed PDO. It is adjustable, which of the two PDO's is to be the high speed PDO. Just as before the first PDO 'high speed' and the second PDO are switched off after delivery. Therefore it is not necessary to change existing CAN applications.
The CAN interface connection controls the data flow from CAN BUS SDO(rx), PDO1(rx) and PDO2(rx) up to the frequency inverter control and also from the frequency inverter to the CAN BUS SDO (tx), PDO1(tx) and PDO2(tx):



The above figure shows the function of the CAN interface connection. The position of the switch S1 is exclusively defined by the parameter address (16 bit index plus 8 bit subindex) contained in the CAN SDO(rx) telegram. Within a certain index range lie the so-called configuration data of the CAN interface connection. These parameters define the behaviour of the CAN interface connection and therefore are realized in this. Access to parameters in the index range 2000(hex) to 5EFF(hex) are passed on as write/read requests to the inverter control.

6.1 Characteristics of the high speed PDO

- The process data mapping is located in the inverter control. The corresponding parameters are in the system parameter group (SY). Since the coding of the PD mapping of the inverter differs from CANopen, it is automatically converted by the CAN operator accordingly.
- The setting of new process output data by CAN is converted by only one special process data service to the inverter control.
- The minimum cycle time for new process output data is appr. 3ms.
- The cyclic reading of process input data is executed by only one special process data read service.
- The minimum achievable cycle time for the reading of process input data is appr. 3ms.
- Not all parameters of the inverter control can be mapped onto the high speed PDO.

6.2 Characteristics of the low speed PDO

- Process data mapping is exclusively managed by the CAN operator.
- The setting of new process output data by CAN is converted to ‘n’ single services (like SDO commands) to the inverter control, at that ‘n’ corresponds to the number of mapped parameters in the PDO mapping.
- The minimum cycle time for new process output data is appr. ‘n’ * 5 ms.
- The cyclic reading of process input data is executed by ‘n’ single read services, at that ‘n’ corresponds to the number of mapped parameters in the PDO mapping.
- The minimum achievable cycle time for the reading of process input data is appr. ‘n’ * 5 ms.
- All parameters of the inverter control can be mapped onto the low speed PDO.

6.3 Process data mapping

The definition of the target for the data in the PDO(rx) telegrams respectively the source for the data in the PDO(tx) telegrams completely abides by the regulations of the CANopen communication profile [12]. Here a complex structured object (parameter) defines the PDO mapping for every data direction.

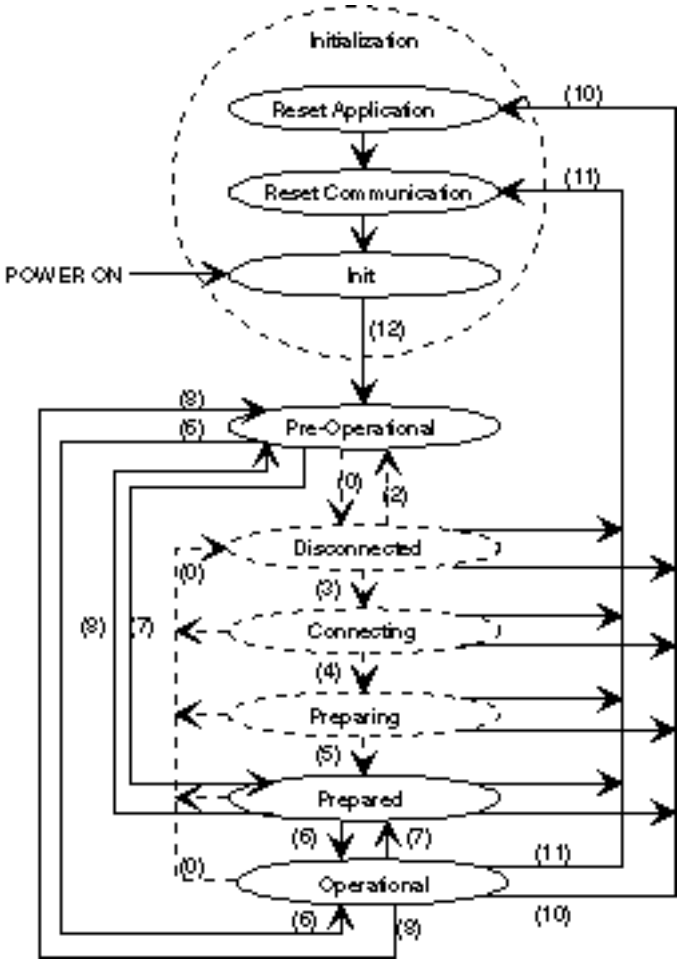
Another object per data direction defines the communication definition (PDO Communication Parameter). See parameter description of

- **1st receive PDO Mapping**
- **1st transmit PDO Mapping**
- **1st receive PDO Parameter**
- **1st transmit PDO Parameter**
- **2nd receive PDO Mapping**
- **2nd transmit PDO Mapping**
- **2nd receive PDO Parameter**
- **2nd transmit PDO Parameter**

in this instruction manual.

6.4 CANopen bootup sequence

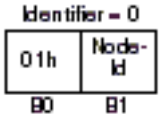
After the initialization phase the KEB CAN interface connection goes automatically into status pre-operational. In this status the communication over SDO(rx) and SDO(tx) with the services domain download (parameter write) and domain upload (parameter read) is already activated. Only the process data communcation is still inactive in this status. It is released by the NMT command Start_Remote_Node() (Fig.). The target of this start sequence is the operating condition operational. In this status the communication is completely activated. With the NMT protocol certain CAN nodes are addressed by the above mentioned Node-Id.



The KEB CANopen interface connection realizes following transitions, illustrated in the above diagram by a solid line:

6: Start_Remote_Node()

CAN-telegram: :



Node_Id = 0 (all NMT-Slaves are addressed) or

Node_Id = inverter-address + 1 (only 1 frequency inverter is addressed)

7: Stop_Remote_Node()

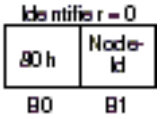
CAN-telegram: :

Node_Id = 0 (all NMT slaves are addressed) or

Node_Id = inverter-address + 1 (only 1 frequency inverter is addressed)

8: Enter_Pre-Operational_State()

CAN-telegram: :

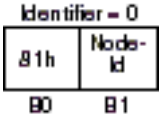


Node_Id = 0 (all NMT slaves are addressed) or

Node_Id = inverter-address + 1 (only 1 frequency inverter is addressed)

10: Reset_Node(): During the execution of this function a software reset is carried out in the KEB CAN interface connection.

CAN-telegram: :

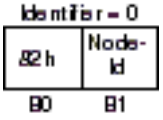


Node_Id = 0 (all NMT slaves are addressed) or

Node_Id = inverter-address + 1 (only 1 frequency inverter is addressed)

11: Reset_Communication(): function like at Reset_Node().

CAN-telegram: :



Node_Id = 0 (all NMT-Slaves are addressed) or

Node_Id = inverter-address + 1 (only 1 frequency inverter is addressed)

12: Enter Pre-Operational automatically(): see above

6.5 Bootup message

The KEB F5 CAN operator releases a bootup message, if the initialization phase is completed after power on. It is a telegram to identifier = 1792 + Node_Id with the data length = 1 and the value = 0.

6.6 Node guarding

A protocol is provided in [12], with which a CAN node can inquire the current status of any node. It belongs to the network management functionality (NMT) of the CAN node and is referred to as node guarding . The KEB CANopen interface connection supports the node guarding. The node guarding request is deposited on the node guarding identifier by a remote frame. The response arrives as data telegram with 1 byte data on the same identifier. The data byte contains the node status plus a toggle bit (MSBit), which is inverted from message to message. Each node has its special node guarding identifier.

At the minimum capability device this identifier is a direct result from the Node-Id:

Node guarding identifier = 1792 + Node-Id

Value of the node status	Meaning
1	DISCONNECTED
2	CONNECTING
3	PREPARING
4	PREPARED
5	OPERATIONAL
127d	PRE_OPERATIONAL

6.7 Life guarding

The F5 CANopen operator supports the life guarding. It concerns the monitoring of the cyclic guarding of the CAN master. For that reason the life guarding should be activated only with the cyclic node guarding. The life guarding operates completely detached from all other monitoring functions. It is activated by the product of the two parameter values guard time and life time factor. Shows the product = 0, then the life guarding is not activated. Otherwise the product specifies the life guarding timeout time. With activated life guarding the node guarding monitoring starts as soon as the first node guard request is received. The function, that is executed upon occurrence of the life guarding timeout case, is adjustable by two further parameters (LifeGuardTout.Addr, LifeGuardTout.Data). It concerns on one hand a write access to any parameter in the inverter control and on the other hand a function code that defines, which action shall be executed in the operator. On delivery the CAN operator is adjusted in such a way, that on occurrence of life guarding timeout the value 1 is written in set 0 of the parameter SY.50 (control word). In addition the CAN oerator switches into the state Pre_Operational.

6.8 Emergency object

The CANopen communication profile DS301 defines a mechanism, after which the nodes signal independently, if the important events incidents. This emergency message is also supported by the KEB F5 CANopen operator. The function is deactivated in default setting. The emergency message is activated by changing the parameter EmergencyCycle to a value unequal 0. Then the CAN operator reads during this cycle time the value of the parameter inverter status (ru.00) from the inverter control and converts it into the ErrorCode value after [13]. Has the value changed an emergency message is send to identifier 128d + Node_Id. That means, that the transition from an error state to normal operating conditions is also announced by an emergency message. The contents of the telegram is only in part firmly set by the profile. All in all the contents of the emergency message looks as follows at the KEB F5 CAN :

Identifier = 128 + Node_Id

B0	B1	B2	B3	B4	B5	B6	B7
Error code		Error register	Inverter status		00h	00h	00h
LB	HB		LB	HB			

All errors are stored in the ‚Predefined ErrorField' defined by the profile. At the KEB F5 CANopen operator this field contains maximal five entries. Whereby the first entry always contains the last error that occurred. Please take the coding of the entries from the description of the parameter of the same name.

7 Coding of the data in the four CAN telegram types

Via this telegram the logical CAN master can inquire (read) or change (write) the value of a parameter. In the communication profile a write-service is referred to as domain download and a read service as domain upload. The KEB CAN interface connection supports only the short form of these two services, thus only one telegram can be exchanged for the service request and another for the service acknowledgement between logical CAN master and the KEB CAN interface connection.

7.1 SDO(rx) telegram

The addressing of the parameter is done via unsigned 16 bit index and unsigned 8 bit subindex. The parameters of the frequency inverter control are in the index range from 2000(hex) to 5EFF(hex). The CAN index is a result of the parameter address (see parameter description of the employed FI control) by adding the offset 2000(hex):

CAN index = KEB parameter address + 2000 (hex)

The subindex serves as additional addressing for complex parameters of the operator. It can also be used for the set-addressing of parameters of the frequency inverter control. The following applies:

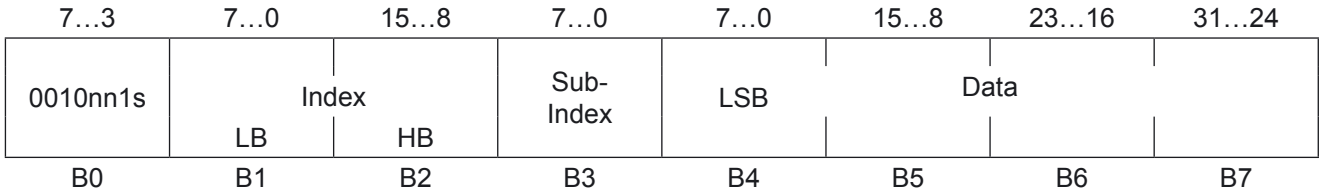
Subindex = 0
For set-programmable parameters the value of the parameter FR.09 specifies the selected set.

Subindex unequal 0
For set-programmable parameters the subindex specifies the selected set. Keep in mind that the set is bit-coded. Thus it is possible to change the value of the parameters in several sets at the same time during the writing. If during the reading several sets are addressed at the same time, then the value of the parameter is returned only under the provision that it is the same in all addressed sets. In case not all values are equal an error signal is returned.

Subindex (if unequal 0):

set 7	set 6	set 5	set 4	set 3	set 2	set 1	set 0
B7	B6	B5	B4	B3	B2	B1	B0

7.1.1 Initiate domain download request (write request of the master)



nn: Only valid with s=1: Contains the number of bytes of the data field, that contains no data.

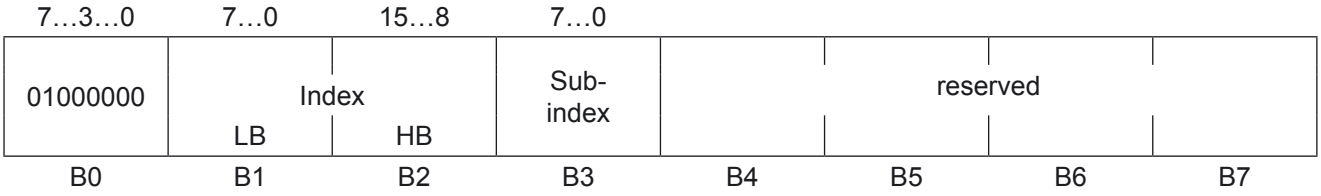
s: If it is equal 1, then nn contains the number of bytes in the data field, that contains no data. Otherwise no display of the data length in nn.

Index: 16 bit (unsigned) addressing of the parameter (see above).

Subindex: 8 bit (unsigned) subaddressing for complex parameters and the direct set-addressing.

Data: Data to be transmitted. The LS byte is transmitted first.

7.1.2 Initiate domain upload request (read request of the master)



Index: 16 bit (unsigned) addressing of the parameter (see above).

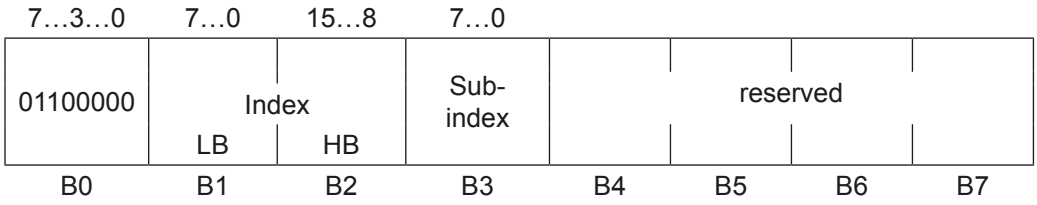
Subindex: 8 bit (unsigned) subaddressing for complex parameters and the direct set-addressing.

7.2 SDO(tx) telegram

7.2.1 Initiate domain download response (write acknowledgement from the FI)

This response is transmitted by the KEB CAN interface connection, if the requested write service was performed error-free.

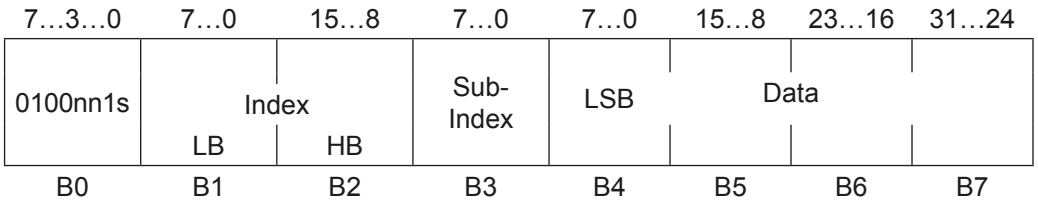
Index: see above
Subindex: see above



7.2.2 Initiate domain upload response (read acknowledgement from the FI)

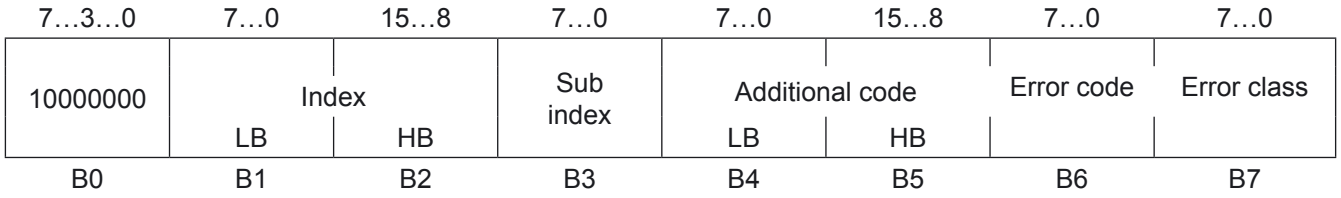
This response is transmitted by the KEB CAN interface connection, if the requested read service was performed error-free.

nn: see above
s: see above
Index: see above
Subindex: see above
Data: see above



7.2.3 Abort domain transfer (error response from the FI)

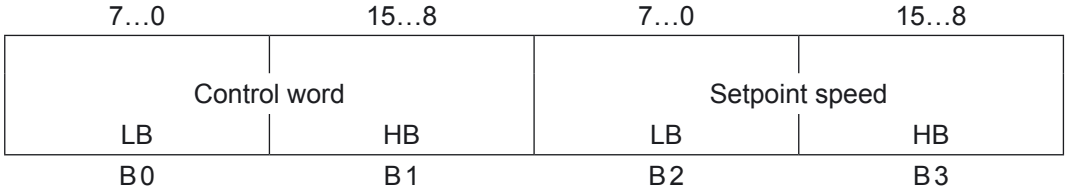
This response is transmitted by the KEB CAN interface connection, if the requested write or read service could not be carried out. In this case an error description is returned.



Error class	Error code	Addi-tional code	Meaning
6	1	0000h	Invalid access to a parameter, e.g. write to a Read_Only parameter.
6	1	0010h	Invalid password.
6	1	0011h	Operation not possible
6	4	0000h	The addressed parameter does not exist.
6	4	0041h	Invalid PD assignment
6	6	0000h	Internal communication between operator and FI control is disturbed.
6	7	0010h	Invalid data length
6	9	0011h	Invalid subindex
6	9	0012h	Invalid language identifier
6	9	0030h	The written value is outside of the valid value range
8	0	0022h	Inverter busy

7.3 PDO1(rx) telegram

With this telegram the logical CAN master transfers new process output data to the inverter. With the default setting the KEB CAN interface connection expects a telegram of > 4 byte data with following content:



The length and assignment of the PDO1(rx) telegram can be changed by different operator parameters . This change can be made only by the SDO(tx) telegram (see above).

Following operator parameters affect the structure of the process output data:

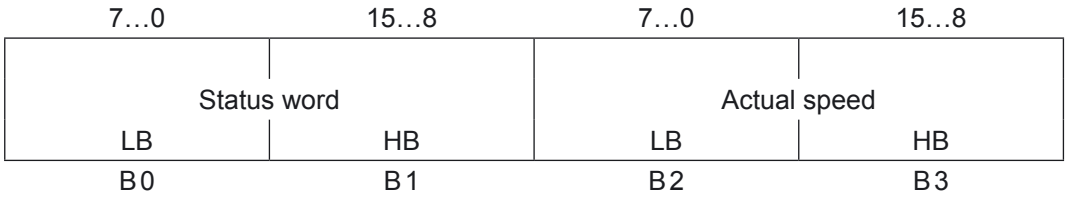
- 1st receive PDO Mapping
- 1st receive PDO Parameter

7.4 PDO1(tx) telegram

With this telegram the KEB CAN interface connection announces process input data to the (logical) CAN master. The length, assignment and control of this telegram is affected by following operator parameters:

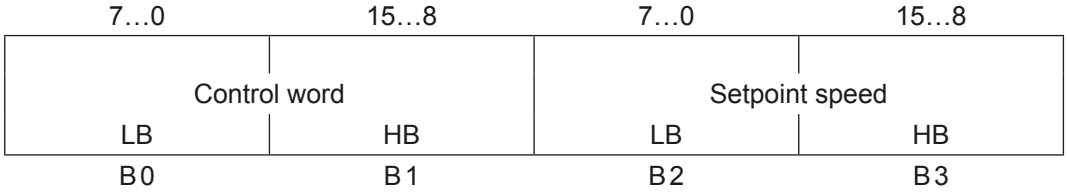
- 1st transmit PDO Mapping
- 1st transmit PDO Parameter

The default setting produces following telegram structure:



7.5 PDO2(rx) telegram

With this telegram the logical CAN master transfers new process output data to the inverter. With the default setting the KEB CAN interface connection expects a telegram of > 4 byte data with following content:



The length and assignment of the PDO2(rx) telegram can be changed by different operator parameters . This change can be made only by the SDO(tx) telegram (see above).
Following operator parameters affect the structure of the process output data:

- 2nd receive PDO Mapping
- 2nd receive PDO Parameter

7.6 PDO2(tx) telegram

With this telegram the KEB CAN interface connection announces process input data to the (logical) CAN master. The length, assignment and control of this telegram is affected by following operator parameters:

- 2nd transmit PDO Mapping
- 2nd transmit PDO Parameter

The default setting produces following telegram structure:

7…0	15…8	7…0	15…8
Status word		Actual speed	
LB	HB	LB	HB
B0	B1	B2	B3

8. Operator Parameters

8.1 Parameters defined by KEB

These parameters define the configuration of the KEB F5 CAN interface connection and therefore are realized here and not in the frequency inverter control:

PD_Stored

CAN SDO Index	**5FE2h**
Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Determines whether the current process data assignment is read from the EEPROM or processed with the standard PD assignment.
Coding	FFh → works with the saved PD assignment other-wise → works with the standard PD assignment.
Default setting	FFh
Notice	A changed value takes effect immediately and is stored non-volatile.

OP_NodeId

CAN SDO Index	**5FE3h**
Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Enables the setting of the CANopen node address in the CAN operator, independent of the inverter address.
Coding	255 : As hitherto the node address is defined from the inverter address (SY.06): Node_Id = inverter address + 1 1…127 : The node address is maintained in the operator and stored: Node_Id = OP_NodeId
Default setting	255
Notice	A value change is effective immediately and stored non-volatile.

Watchdog activation

CAN SDO Index	**5FDAh**
SDO-Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Enables the delay of the activation of the fieldbus watchdog after power on respectively a reset command via CAN.
Coding	0 : The fieldbus watchdog is active immediately. Values unequal zero are bit-coded and have following significance: Bit 0 : Activation of the fieldbus watchdog after the first SYNC telegram Bit 1 : Activation of the fieldbus watchdog after the first node guarding Bit 2 : Activation of the fieldbus watchdog after the first transition into the node status OPERATIONAL Bit 3 : Activation of the fieldbus watchdog after the first PDOUT1 telegram Bit 4 : Activation of the fieldbus watchdog after the first PDOUT2 telegram Bit 5 : Activation of the fieldbus watchdog after the first SDO telegram
Default setting	0
Permitted PDO mapping	no mapping
Notice	A value change is effective immediately and stored non-volatile. Several occurrences can be defined as fieldbus watchdog activation. In that case watchdog becomes active as soon as one of the defined occurrences takes place.

B7	B6	B5	B4	B3	B2	B1	B0
		1. SDO	1. PDOUT2	1. PDOUT1	1. OPERATI- ONAL	1. Node Guar- ding	1. SYNC

Watchdog inhibit

CAN SDO Index	**5FF9h**
Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Defines upon which events the fieldbus watchdog is triggered. The fieldbus watchdog is used to put the frequency inverter into the error status, if no more activities take place on CAN. The real activation and programming of watchdog is adjusted in the FI control. Take the parameters to be adjusted from the instruction manual of the FI control.
Coding	Bit-coded: Bit 0 = 1 When starting a PDOUT telegram to the FI control the watchdog is reset. Note, that the occurrence of this event also depends on the adjustment of the parameter 1st Receive PDO ParameterTx_type as well as on the value of the parameter PDOUT_Wr_Mode. Bit 1 = 1 At the beginning of the processing a SDO job the watchdog is reset. Bit 2 = 1 If the node does not detect any transmission problems onto CAN, the watchdog is reset. Bit 3 = 1 At every receipt of a SYNC telegram the watchdog is reset. Bit 4 = 1 At every receipt of a node guard request telegram the watchdog is reset. Bit 5 = 1 At every receipt of a SYNC telegram the watchdog is reset, provided that at least once process output data have been transmitted to the frequency inverter control.
Default setting	07h The watchdog is reset, if: - process output data are written to the FI control , - an SDO job is started and - no transmission problems to CAN are detected. .
Notice	A changed value takes effect immediately and is stored non-volatile.

PDOUT_WrMode

CAN SDO Index	**5FE4h**
Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Defines the conditions under which PDOUT data are written to the FI control. Herewith the communication between CAN operator and FI control can be relieved.
Coding	0 : All PDOUT data are always written to the FI control, whether they are changed or not. 255 : All PDOUT data are always written to the inverter, if at least one of the values was changed. other-wise : Only the changed values are written.
Default setting	255
Notice	A value change is effective immediately and stored non-volatile.

HS_PDO_Index

CAN SDO Index	**5FE5h**
Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	With this parameter it is defined which PDO should be the high-speed PDO.
Coding	0 : 1. PDO is high-speed PDO 1 : 2. PDO is high-speed PDO
Default setting	0
Notice	A changed value is stored non-volatile, but becomes active only after the next switch-on or reset node command. Moreover, in case of a change of value all PDO's are deactivated (bit 31 from PDO parameter CobID = 1).

PDIN1_Cycle_Time

CAN SDO Index	**5FE6h**
Subindex	0
Object type	Single variable (var)
Data length	2 byte
Access	READ_WRITE
Meaning	Specifies the cycle time in which the process input data of the PDO1 are read in status OPERATIONAL by the FI control.
Coding	1 ms
Default setting	25 = 25 ms
Notice	A changed value takes effect immediately and is stored non-volatile.

PDIN2_Cycle_Time

CAN SDO Index	**5FE7h**
Subindex	0
Object type	Single variable (var)
Data length	2 byte
Access	READ_WRITE
Meaning	Specifies the cycle time in which the process input data of PDO2 are read in status OPERATIONAL by the FI control.
Coding	1 ms
Default setting	100 = 100 ms
Notice	A changed value takes effect immediately and is stored non-volatile.

SAVE_CAN_Baud

CAN SDO Index	**5FFEh**
Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Serves for the non-volatile storing of the adjusted CAN transmission rate.
Coding	Write: FFh : Non-volatile storing of CAN_Baud 0 : no storing Read: FFh : Adjusted value corresponds with the stored value 00h : Adjusted value is not equal to stored value

CAN_Baud

CAN SDO Index	**5FFFh**
Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Index for CAN transmission rate
Coding	0 = 10 Kbit/s 1 = 20 Kbit/s 2 = 50 Kbit/s 3 = 100 Kbit/s 4 = 125 Kbit/s 5 = 250 Kbit/s 6 = 500 Kbit/s* 7 = 1000 Kbit/s* 8 = 800 Kbit/s* 9 = 25 Kbit/s
Default setting	1
Notice	A changed value takes effect immediately but is not automatically stored non-volatile. The bit-timing abides by the specifications of the working committee Physical-Layer of CiA [2]. See Annex regarding bit-timing. What kind of transmission rates are possible depends on the line length, the sum of the deceleration times and the bit-timing and must be cleared up for each individual case.

CAN_Baud2

CAN SDO Index	**5FECh**
Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Index for CAN transmission rate alternatively to CAN_Baud (s.o.)
Coding	0 = 10 Kbit/s 1 = 20 Kbit/s 2 = 50 Kbit/s 3 = 100 Kbit/s 4 = 125 Kbit/s 5 = 250 Kbit/s 6 = 500 Kbit/s* 7 = 1000 Kbit/s* 8 = 800 Kbit/s* 9 = 25 Kbit/s
Default setting	1
Notice	Contrary to parameter CAN_Baud a changed value is immediately stored non-volatile, but becomes active only after a reset node command or after the next switch-on..

* Please observe chapter ‘Important Warning Notice’

8.2 Parameters defined by the communication profile [12]

Device type (according to CANopen [13])

CAN SDO Index	**1000h**
Subindex	0
Object type	Single variable (var)
Data length	4 byte
Access	READ_ONLY
Meaning	Decribes the unit type according to CANopen communication profile.
Coding	No predefinition up to now.
Default setting	0
Notice	This parameter is constant, therefore it can be read only.

Error register (according to CANopen [13])

CAN SDO Index	**1001h**
Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_ONLY
Meaning	Indicates the Error-Status of the CANopen user.
Coding	Bit 0 = 1 → Error exists
Default setting	0
Notice	This parameter can be read only. The inverter parameter status (ru.00) serves as source for this parameter. The conversion of the ru.00 values into values of the error register is found in the table in the annex. Note, the value of the Error-Register is updated only with activated Emergency-Processing (→ EmergencyCycle).

Manufacturer Status Register

CAN SDO Index	**1002h**				
Subindex	0				
Object type	Single variable (var)				
Data length	4 byte				
Access	READ_ONLY				
Meaning	Direct mapping of the parameter inverter status (ru.00) in DS301 parameter range.				
Coding	See description of the parameter inverter status (ru.00) in the instruction manual of the inverter control.				
Default setting	0				
Permitted PDO mapping	High speed PDO		Low speed PDO		
	rx	tx	rx	tx	
	NO	YES	NO	YES	
Notice	Is mapped internally on the parameter ru.00.				

Manufacturer Device Name

CAN SDO Index	**1008h**
Subindex	0
Object type	Single variable (var)
Data length	4 Byte
Access	READ_ONLY
Meaning	Outputs the value of the parameter Inverter_Identification (SY.02) of the FI control as 4-character hexadecimal-string.
Coding	The value 1234h would be transferred in the SDO response telegram as follows:
Default setting	Depending on the inverter type
Permitted PDO-mapping	no mapping

B4	B5	B6	B7
31h	32h	33h	34h

Identify Object

CAN SDO Index	**1018h**
Subindex	0
Object type	structured variable (record)
Data length	1 byte
Access	READ_ONLY
Meaning	Specifies the number of entries in this object.
Coding	1
Default setting	2
Notice	The value of this parameter can be read only.

Subindex	1
Data length	4 byte
Meaning	Manufacturer identification assigned by the CAN in automation user group.
Coding	Bit 31…Bit 24 : Department Bit 23…Bit 0 : Company
Default setting	00000014h
Notice	The value of this parameter can be read only.

Subindex	2
Data length	4 byte
Meaning	Product description
Coding	00000004h = Type F4 00000005h = Type F5
Default setting	00000005h
Notice	The value of this parameter can be read only.

Manufacturer software version

CAN SDO Index	**100Ah**
SDO-Subindex	0
Object type	Single variable (var)
Data length	4 byte
Access	READ_ONLY
Meaning	Outputs the value of the parameter software version (In.06) of the FI control as 4-character hexadecimal string.
Coding	The value 140h/260d (= version 2.60) would be transferred in the SDO response telegram as follows:
Default setting	Depending on the software version of the inverter control
Permitted PDO mapping	no mapping

B4	B5	B6	B7
30h	31h	30h	34h

1st receive PDO Parameter

CAN SDO Index	**1400h**
Subindex	0
Object type	structured variable (record)
Data length	1 byte
Access	READ_WRITE
Meaning	Specifies the number of entries that can be addressed under this object.
Coding	1
Default setting	2
Notice	The value of this parameter can be read only.

Subindex	1
Data length	4 byte
Meaning	Indicates to which identifier the PDO(rx) for the transfer of the process output data is transferred. In addition to it control information for this PDO are contained in the higher bits.
Coding	Bit 31(MSB) = 0 → The processing of the process output data is activated Bit 31(MSB) = 1 → The processing of the process output data is deactivated. Bit 30 = 0 → Remote frame on the correspon-dingidentifier is responded. Bit 30 = 1 → Remote frame is not answered. Bit 29 = 0 → 11-bit identifier (CAN V2.0A) Bit 29 = 1 → 29-bit ientifier (CAN V2.0B), not adjustable here. But 29-bit identifier telegrams are received and processed. Bit 28...Bit 0 = Identifier (Bit0 = LSB), here for bit28 to bit11=fixed=0.
Default setting	00000200h + Node_Id
Notice	A changed value takes effect immediately and is stored non-volatile. When activating the process data processing (Bit31 from “31 1” to “0”) the setting of the parameter 1st receive PDO mapping (Index = 1600h) is transferred to the inverter control. If the FI control does not accept the map-ping an error response is returned here and the process output data processing remains disabled. If the PD map-ping is accepted by the FI, it is automatically stored non-volatile and the process output data processing is enabled as desired. Since the identifier assignment of the PDOs is derived directly from the Node_Id, the Bits Bit28 to Bit0 can only be read. During writing these Bits are ignored.

further on next side

Subindex	2
Data length	1 byte
Meaning	Defines, when and how this object is transmitted on the CAN Bus .
Coding	0 ... 240: On receipt of a SYNC command (Identifier = 128d, data length = 0) the current process output data are transferred to the FI control. 254 (asynchronous, manufacturer-specific): The process output data are transferred to the FI control as soon as at least one has changed. . 255 (asynchronous, profile-specific): See asynchronous, manufacturer-specific.
Notice	A changed value takes effect immediately and is stored non-volatile. Please also consider the effect of the parameter PDOUT_ WrMode.

2nd receive PDO Parameter

CAN SDO Index	**1401h**
Subindex	0
Object type	structured variable (Record)
Data length	1 byte
Access	READ_WRITE
Meaning	Specifies the number of entries that can be addressed under this object.
Coding	1
Default setting	2
Notice	The value of this parameter can be read only.

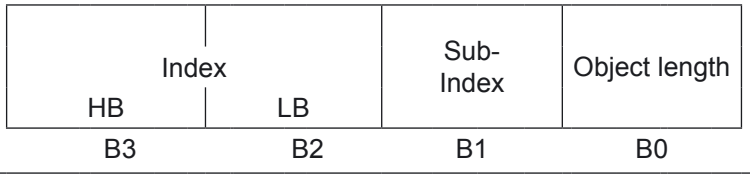
Subindex	1
Data length	4 byte
Meaning	Indicates to which identifier the PDO(rx) for the transfer of the process output data is transferred. In addition to it control information for this PDO are contained in the higher bits.
Coding	Bit 31(MSB) = 0 → The processing of the process output data is activated Bit 31(MSB) = 1 → The processing of the process output data is deactivated. Bit 30 = 0 → Remote frame on the corresponding identifier is responded. Bit 30 = 1 → Remote frame is not responded. Bit 29 = 0 → 11-bit identifier (CAN V2.0A) Bit 29 = 1 → 29-bit identifier (CAN V2.0B), not adjustable here. But 29-bit identifier telegrams are received and processed. Bit 28...Bit 0 = Identifier (Bit0 = LSB), here for Bit28 to Bit11=fixed=0.
Default setting	80000300h + Node_Id
Notice	A changed value takes effect immediately and is stored non-volatile. On activation of the process data processing (Bit 31 from “1” to “0”) the setting of the 2nd receive PDO mapping is converted to a corresponding inverter mapping. If this could be executed successfully, the mapping is automatically stored non-volatile. Since the identifier assignment of the PDOs is derived directly from the Node_Id, bits 28 to bit 0 can only be read. During writing these Bits are ignored.
Subindex	2
Data length	1 byte

further on next side

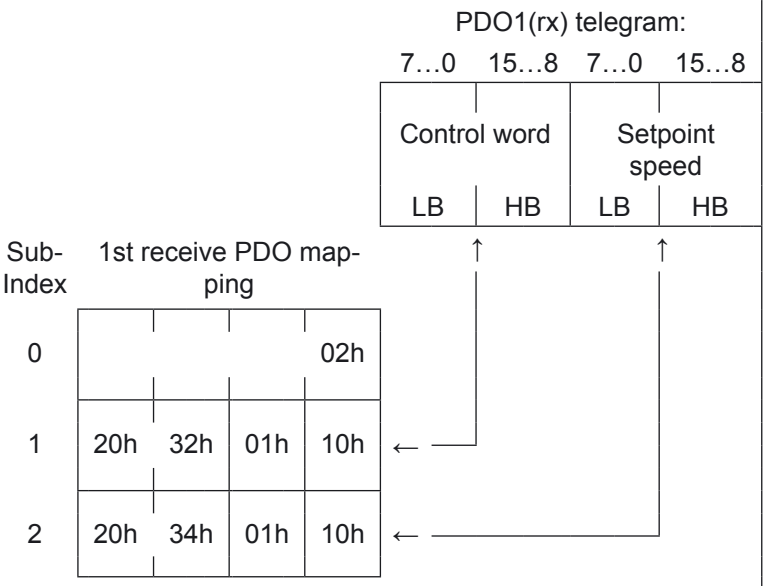
Meaning	Defines, when and how this object is transmitted on the CAN Bus .
Coding	0 ... 240: On receipt of a SYNC command (Identifier = 128d, data length = 0) the current process output data are transferred to the FI control. 254 (asynchronous, manufacturer-specific): The process output data are transferred to the FI control as soon as at least one has changed. 255 (asynchronous, profile-specific): See asynchronous, manufacturer-specific.
Notice	A changed value takes effect immediately and is stored non-volatile. Please also consider the effect of the parameter PDOUT_ WrMode.

1st receive PDO Mapping

CAN SDO Index	**1600h**
Subindex	0
Object type	structured variable (Record)
Data length	1 byte
Access	READ_WRITE
Meaning	Specifies the number of entries that can be addressed under this object.
Coding	1 (maximum valid value range 1.....4).
Default setting	2
Notice	A writing of this parameter causes the automatic deactivation of the process output data processing (Bit31 of Index 31 1400h, Subindex = 1 is set to “1”).

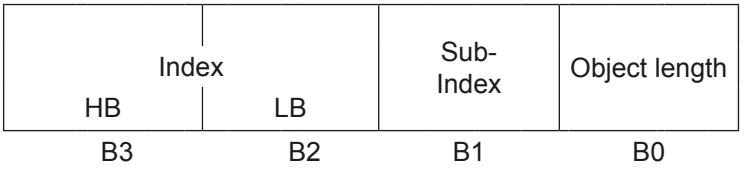
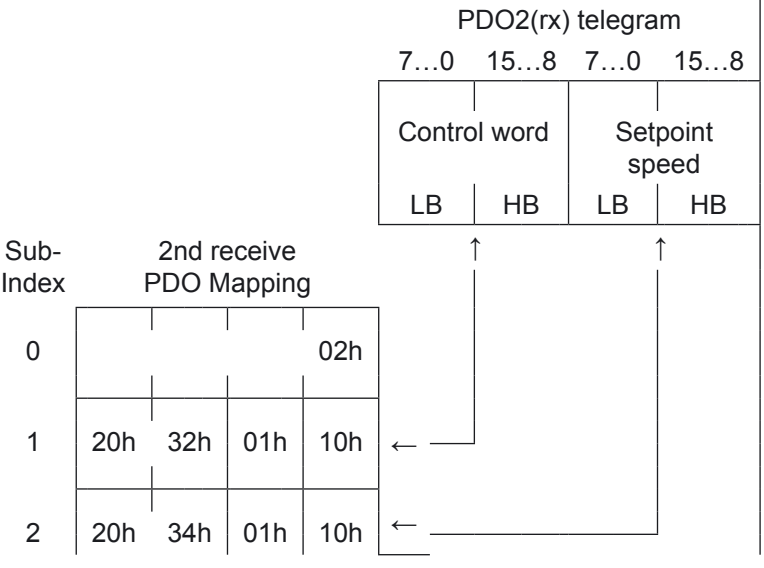
Subindex	1 upto maximum 4
Data length	4 byte
Meaning	Describes an object mapping. The index, subindex and the object length are specified in bits.
Coding	
Default setting	see below

further on next side

Notice	A writing of this parameter causes the automatic deactivation of the process output data processing (Bit31 of Index 31 1400h, Subindex = 1 is set to “1”). The correlation between process output data mapping and the corresponding PDO1(rx) telegram structure is shown once more in the default assignment:  Example for the coding The first mapped object in receive PDO shall not be the control word, but the parameter with the index 2302h and subindex = 1. In this case the 8 bytes of the initiate-domain-download-requests are to be filled as follows (data example on CAN BUS): 23h 00h 18h 01h 10h 01h 02h 23h B0 B1 B2 B3 B4 B5 B6 B7

2nd receive PDO Mapping

CAN SDO Index	**1601h**
Subindex	0
Object type	Structured variable (record)
Data length	1 byte
Access	READ_WRITE
Meaning	Specifies the number of entries that can be addressed under this object.
Coding	1 (maximal valid value range 1...4).
Default setting	2
Notice	A writing of this parameter causes the automatic deactivation of the process output data processing (Bit31 of index 1401h, subindex = 1 is set to “1”).

Subindex	1 upto maximum 4
Data length	4 byte
Meaning	Describes an object mapping. The index, subindex and the object length are specified in bits.
Coding	
Default setting	see below
Notice	A writing of this parameter causes the automatic deactivation of the process output data processing (Bit31 of index 1401h, subindex = 1 is set to “1”). The correlation between process output data mapping and the corresponding PDO2(rx)-telegram structure is shown once more in the default assignment: 

1st transmit PDO Parameter

CAN SDO Index	**1800h**
Subindex	0
Object type	Structured variable (record)
Data length	1 Byte
Access	READ_WRITE
Meaning	Specifies the number of entries in this object.
Coding	1
Default setting	3
Notice	The value of this parameter can be read only.

Subindex	1
Data length	4 byte
Meaning	Indicates to which identifier the PDO(tx) for the transfer of the process input data is transferred. In addition some additional information are contained in the higher bits.
Coding	Bit 31(MSB) = 0 → The processing of the process output data is activated. Bit 31(MSB) = 1 → The processing of the process input data is deactivated. Bit 30 = 0 → Remote frame on the corresponding identifier is responded. Bit 30 = 1 → Remote frame is not responded. Bit 29 = 0 → 11-Bit identifier (CAN V2.0A) Bit 29 = 1 → 29-Bit identifier (CAN V2.0B), not adjustable here. Bit 28...Bit 0 = Identifier (Bit0 = LSB), here for Bit28 to Bit11=fixed=0.
Default setting	00000180h + Node_Id
Notice	A changed value takes effect immediately and is stored non-volatile. On activation of the process data processing (Bit31 from “1” to “0”) the setting of the parameter 1st transmit PDO mapping (Index 1A00h) is transferred to the inverter control. If the FI control does not accept the mapping an error response is returned and the process input data processing remains switched off. If the PDO mapping is accepted by the FI, it is automatically stored non-volatile and the process input data processing is enabled as desired. Since the identifier assignment of the PDOs is derived directly from the Node_Id, the Bits Bit28 to Bit0 can only be read. During writing these bits are ignored.

Subindex	2
Data length	1 byte
	further on next side

Meaning	Defines, when and how this object is transmitted on the CAN Bus .
Coding	0 (synchronous acyclic): At every receipt of a SYNC a PDO(tx) telegram is transmitted on CAN. 1 - 240 (synchronous, cyclic): In this setting range it is adjusted by means of the value, how many SYNC telegrams must be received, before a PDO(tx) telegram is transmitted on CAN. 252 (synchronous-RTROnly): A PDO(tx) telegram is only transmitted after a remote request on the PDO(tx) identifier. 253 (asynchronous-RTROnly): A PDO(tx) telegram is only transmitted after a remote request on the PDO(tx) identifier. 254 (asynchronous, manufacturer-specific): A PDO(tx) telegram is transmitted as soon as at least one byte has changed. 255 (asynchronous, profile-specific): A PDO(tx) telegram is transmitted as soon as at least one byte has changed.
Notice	A changed value takes effect immediately and is stored non-volatile.

Subindex	3
Data length	2 byte
Meaning	Describes the minimal temporal distance between two CAN telegrams on this identifier.
Coding	100 µs
Default setting	150 (= 15 ms)
Notice	A changed value takes effect immediately and is stored non-volatile. The internal resolution for the Inhibit-Time is 1ms. Thus the adjusted value has an inaccuracy of +/- 1 ms.

2nd transmit PDO Parameter

CAN SDO Index	**1801h**
Subindex	0
Object type	Structured variable (record)
Data length	1 byte
Access	READ_WRITE
Meaning	Specifies the number of entries in this object.
Coding	1
Default setting	3
Notice	The value of this parameter can be read only.

Subindex	1
Data length	4 byte
Meaning	Indicates to which identifier the PDO(tx) for the transfer of the process input data is transferred. In addition some additional information are contained in the higher bits.
Coding	Bit 31 (MSB) = 0 → The processing of the process output data is activated. Bit 31 (MSB) = 1 → The processing of the process input data is deactivated. Bit 30 = 0 → Remote frame on the corresponding identifier is responded. Bit 30 = 1 → Remote frame is not answered. Bit 29 = 0 → 11-Bit identifier (CAN V2.0A) Bit 29 = 1 → 29-Bit identifier (CAN V2.0B), not adjustable here. Bit 28...Bit 0 = Identifier (Bit0 = LSB), here for bit28 to bit11=fixed=0.
Default setting	80000280h + Node_Id
Notice	A changed value takes effect immediately and is stored non-volatile. On activation of the process data processing (Bit31 from “1” to “0”) the setting of the parameter 2nd transmit PDO mapping is converted into a corresponding inverter mapping. If this could be executed successfully, the mapping is automatically stored non-volatile. Since the identifier assignment of the PDOs is derived directly from the Node_Id, the Bits Bit28 to Bit0 can only be read. During writing these bits are ignored.

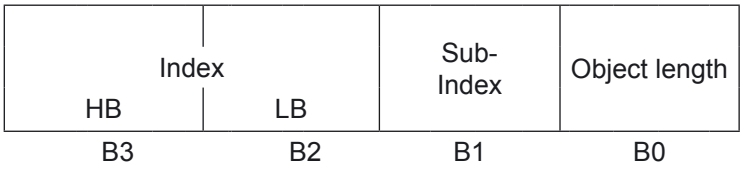
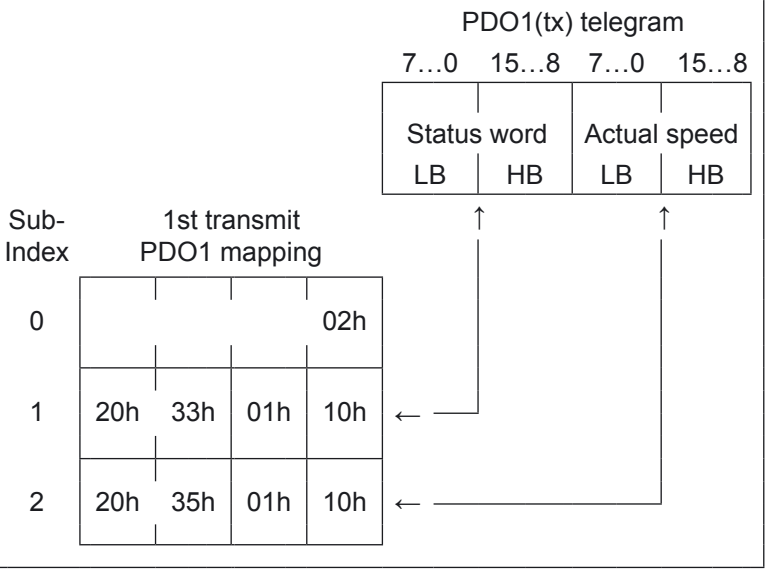
Subindex	2
Data length	1 byte
	further on next side

Meaning	Defines, when and how this object is transmitted on the CAN Bus .
Coding	0 (synchronous acyclic): At every receipt of a SYNC a PDO(tx) telegram is transmitted on CAN. 1 - 240 (synchronous, cyclic): In this setting range it is adjusted by means of the value, how many SYNC telegrams must be received, before a PDO(tx) telegram is transmitted on CAN. 252 (synchronous-RTROnly): A PDO(tx) telegram is only transmitted after a remote request on the PDO(tx) identifier. 253 (asynchronous-RTROnly): A PDO(tx) telegram is only transmitted after a remote request on the PDO(tx) identifier. 254 (asynchronous, manufacturer-specific): A PDO(tx) telegram is transmitted as soon as at least one byte has changed. 255 (asynchronous, profile-specific): A PDO(tx) telegram is transmitted as soon as at least one byte has changed.
Notice	A changed value takes effect immediately and is stored non-volatile.

Subindex	3
Data length	2 byte
Meaning	Describes the minimal temporal distance between two CAN telegrams on this identifier.
Coding	100 µs
Default setting	1000 (= 100 ms)
Notice	A changed value takes effect immediately and is stored non-volatile. The internal resolution for the Inhibit-Time is 1ms. Thus the adjusted value has an inaccuracy of +/- 1 ms.

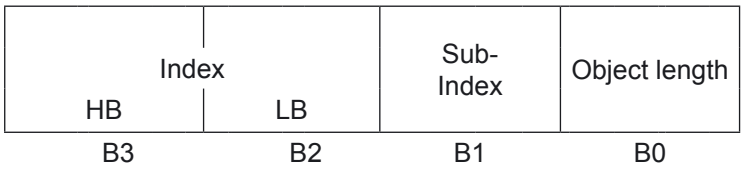
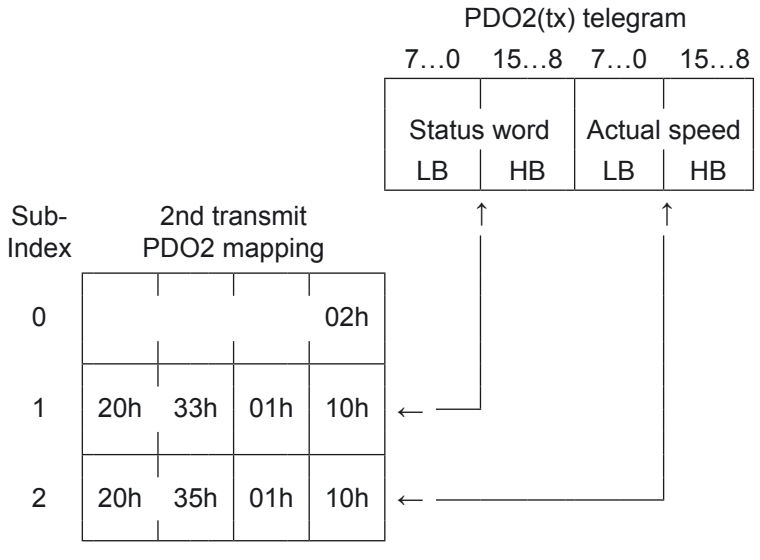
1st transmit PDO Mapping

CAN SDO Index	**1A00h**
Subindex	0
Object type	Structured variable (record)
Data length	1 byte
Access	READ_WRITE
Meaning	Specifies the number of entries that can be addressed under this object.
Coding	1 (maximal valid value range 1.....4).
Default setting	2
Notice	A writing of this parameter causes the automatic deactivation of the process input data processing (Bit31 of index 1800h, subindex = 1 is set to “1”).

Subindex	1 upto maximum 4
Data length	4 byte
Meaning	Describes an object mapping. The index, subindex and the object length are specified in bits.
Coding	
Default setting	see below
Notice	A writing of this parameter causes the automatic deactivation of the process input data processing (Bit31 of index 1800h, subindex = 1 is set to “1”). The correlation between process input data mapping and the corresponding PDO1(tx) telegram structure is shown once more in the default assignment: 

2nd transmit PDO Mapping

CAN SDO Index	**1A01h**
Subindex	0
Object type	Structured variable (record)
Data length	1 byte
Access	READ_WRITE
Meaning	Specifies the number of entries that can be addressed under this object.
Coding	1 (maximal valid value range 1.....4).
Default setting	2
Notice	A writing of this parameter causes the automatic deactivation of the process input data processing (Bit31 of index 1801h, subindex = 1 is set to “1”).

Subindex	1 upto maximum 4
Data length	4 byte
Meaning	Describes an object mapping. The index, subindex and the object length are specified in bits.
Coding	
Default setting	see below
Notice	A writing of this parameter causes the automatic deactivation of the process input data processing (Bit31 ofIndex 1801h, subindex = 1 is set to “1”). The correlation between process input data mapping and the corresponding PDO2(tx)-telegram structure is shown once more in the default assignment: 

8.3 Parameters for the life guarding

Guard Time

CAN SDO Index	**100Ch**
SDO-Subindex	0
Object type	Single variable (var)
Data length	2 byte
Access	READ_WRITE
Meaning	Defines together with the Life Time Factor the monitoring time for the life guarding.
Coding	0 = Life guarding switched off other-wise = 1 ms
Permitted PDO-mapping	not mappable
Notice	A changed value takes effect immediately and is stored non-volatile.

Life Time Factor

CAN SDO Index	**100Dh**
SDO-Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Defines together with the guard time the monitoring time for the life guarding.
Coding	0 = Life guarding switched off other-wise = 1
Permitted PDO-mapping	not mappable
Notice	A changed value takes effect immediately and is stored non-volatile.

LifeGuardTout.Addr

CAN SDO Index	**5FDFh**
SDO-Subindex	0
Object type	Single variable (var)
Data length	4 byte
Access	READ_WRITE
Meaning	Defines together with the LifeGuardTout.Data the function, that is executed once after the life guarding timeout occurred.
Coding	The value consists of the parameter address to be written and the parameter set as well as the function code for the operator. The mapping below shows the structure of the value, as it appears in the CAN SDO telegram: B0 B1 B2 B3 opfunc | Set | Parameter address LB | HB opfunc = 0 → no activity in the operator 1 → Transition into Pre_Operational Parameter address = Address of the parameter to be written Set = Set selection byte of the parameter to be written.
Permitted PDO-mapping	not mappable
Default setting	Parameter address = 0032h(SY.50) Parameter set = 1 (set 0) opfunc = 1: Transition into Pre_Operational
Notice	A changed value takes effect immediately and is stored non-volatile.

LifeGuardTout.Data

CAN SDO Index	**5FDEh**
SDO-Subindex	0
Object type	Single variable (var)
Data length	4 byte
Access	READ_WRITE
Meaning	Defines the value of the inverter parameter to be written in case life guarding timeout occurs.
Coding	Depending on the selected inverter parameter.
Permitted PDO-mapping	not mappable
Default setting	1
Notice	A changed value takes effect immediately and is stored non-volatile.

8.4 Parameters of the emergency processing

EmergencyCycle

CAN SDO Index	**5FDDh**
SDO-Subindex	0
Object type	Single variable (var)
Data length	2 byte
Access	READ_WRITE
Meaning	Serves for the activation/deactivation of the emergency processing. The value also shows the cycle time, during which the value of the parameter inverter status is read, to indicate a possible error.
Coding	0 = switched off other-wise = 1 ms
Permitted PDO-mapping	not mappable
Default setting	0 (switched off)
Notice	A changed value takes effect immediately and is stored non-volatile.

Predefined ErrorField

CAN SDO Index	**1003h**
SDO-Subindex	0, 1, 2, 3, 4, 5
Object type	Single variable (var)
Data length	4 Byte
Access	Subindex = 0 → READ_WRITE otherwise = READ_ONLY
Meaning	This field contains the last five error messages. The first error field (subindex=1) always contains the error that occurred last. According to this, entries with higher subindex have occurred earlier.
Coding	Subindex = 0 → Number of filled entries Subindex = 0 ! B0 B1 B2 B3 ErrorCode | Inverter state LB HB LB HB
Permitted PDO mapping	not mappable
Default setting	0 (switched off)
Notice	Only the subindex = 0 is writable. When writing with the value = 0 the entire field is reset.

8.5 Parameter for the synchronous mode

ComCycle

CAN SDO Index	**1006h**
SDO-Subindex	0
Object type	Single variable (var)
Data length	4 byte
Access	READ_WRITE
Meaning	Serves for the activation/deactivation of the synchronous mode. The value is coded in µs, but has an internal resolution of 1ms.
Coding	0 = OFF (normal mode) other-wise = 1 µs
Value range	0, 1000, 2000, 3000…65000
Notice	The parameter is available only over CAN SDO. The parameter is stored non-volatile and shows the value zero after every new start. The CAN operator switches back automatically into the normal mode, if after four-times ComCycle-time no SYNC telegram is received (also see HS_SyncToutDelay).

HS_SyncToutDelay

CAN SDO Index	**5FE0h**
SDO-Subindex	0
Object type	Single variable (var)
Data length	2 byte
Access	READ_WRITE
Meaning	With this parameter the SYNC timeout monitoring can be delayed by the desired number of SYNCs.
Coding	0 = The SYNC timeout monitoring is active immediately after change-over into synchronous mode. other-wise = Number of SYNC telegrams until the timeout monitoring becomes active.
Value range	0…65535
Notice	The parameter is stored non-volatile.

9. Access to Operator Parameters over the Diagnostic Interface

The operator parameters are managed in the operator itself. Access to these parameters is possible by the CAN interface as well as by the diagnostic interface (per COMBIVIS). Keep in mind, that a CAN parameter is partly distributed onto several operator parameters and thus the outer appearance of a parameter on CAN and in COMBIVIS can vary slightly. This chapter lists the operator parameters that are of relevance to the user with the reference to the corresponding CAN parameter. The complete description is found in the chapter operator parameters.

Following information to the addressing of operator parameters and parameters of the inverter control: All parameters of a KEB inverter (FI parameter + operator parameter) are addressed over the diagnostic interface with a 16-bit parameter address plus 8-bit set selection byte. For the parameter address applies, that it is divided into the parameter group address (high byte) and a consecutive number (low byte):

KEB Parameter address:

B15 B8	B7 B0
Group	Number

Every parameter group can be occupied with operator parameters as well as with parameters of the inverter control. For the distinction of the localization of the parameters the consecutive „number“ is divided into two ranges:

Number = 0…127 → Parameters of the inverter control
Number = 128…255 → Operator parameters

Moreover, it must be noted, that exactly as with the inverter parameters some operator parameters exist several times. These use the set selection byte in KEB in the familiar manner. Here applies, that over set 0 the first mapping of the parameter is addressed and in set 1 the second one. Presently the set-programmable operator parameters are limited to the process data parameters. Since at present the CAN operator supports two PDOs, these parameters exist in set 0 for the first PDO and in set 1 for the second one.

All set-programmable parameters are provided with a corresponding reference under notice. All other parameters exist only in set 0.

Example:
Parameter PD_In_Para_CobId of the first PDO is addressed over the diagnostic interface with set selection byte = 01 (hex). The same parameter of the second PDO is in set 1 (set selection byte = 02 (hex)).
Keep in mind that the simultaneous addressing of a operator parameter in several sets is generally prohibited.

Operator type

Parameter address	**0180h**
Parameter-Id	OS.00
Data length	2 byte
Meaning	Describes the operator type
Set-programming	Not set-programmable
Coding	The two most significant decimal places specify the customer, the next two decimal places name the type and the next two decimal places indicate the version resolved into 0.1. Example: Value 10421 (dec) means: customer = 1 (KEB) Type = 4 (CAN) Version = 21 (V 2.1)
Notice	Only for information.

Software date

Parameter address	**0182h**
Parameter-Id	OS.02
Data length	2 byte
Meaning	Configures the software date
Set-programming	Not set-programmable
Coding	The lowest decimal digit shows the year 2000. The next two decimal digits show the month. The next two decimal digits show the day.
Notice	The representation is as follows: day, month, year 2000

Diag Error Count

Parameter address	**0183h**
Parameter-Id	OS.03
Data length	2 byte
Meaning	Specifies the number of occurred errors on the diagnostic interface.
Coding	1
Notice	Only for information.

Diag response delay time

Parameter address	**0184h**
Parameter-Id	OS.04
Data length	1 byte
Meaning	Minimal response delay time for inquiries over the diagnostic interface.
Coding	1 ms
Notice	This parameter is not available over CAN.

Diag Baudrate

Parameter address	**0185h**
Parameter-Id	OS.05
Data length	1 byte
Meaning	Specifies the transmission rate on the diagnostic interface.
Coding	0 : 1200 Bit/s 1 : 2400 Bit/s 2 : 4800 Bit/s 3 : 9600 Bit/s 4 : 19200 Bit/s 5 : 38400 Bit/s
Notice	The value of the parameter is a mapping of the parameter SY.07 and therefore it is only readable over the diagnostic interface.

HSP5 Max InvBusy retries

Parameter address	**0186h**
Parameter-Id	OS.06
Data length	1 byte
Meaning	Indicates how often a HSP5 service to the inverter control is repeated, it the inverter rejects the service with the error ‘Inverter busy’ .
Coding	1

HSP5 Tout Count

Parameter address	**0187h**
Parameter-Id	OS.07
Data length	2 byte
Meaning	Counts the time exceedance at the internal communication between operator and FI control.
Coding	1
Notice	This parameter is not available over CAN and exists only in set 0.

OP_Node_Id

Parameter address	**0280h**
Parameter-Id	FB.00
Data length	1 byte
Notice	This parameter is identical with the CAN Parameter OP_ Node_Id and exists only in set 0.

CAN_Baud2

Parameter address	**0281h**
Parameter-Id	FB.01
Data length	1 byte
Notice	This parameter corresponds to the CAN Parameter CAN_ Baud2 (s.o.) and exists only in set 0.

Act_CAN_Baud

Parameter address	**0282h**
Parameter-Id	FB.02
Data length	1 bgyte
Meaning	Shows the currently adjusted CAN bitrate.
Coding	See CAN_Baud.
Notice	This parameter ist Read_Only and not available on CAN and exists only in set 0.

Watchdog activation

Parameter address	**0283h**
Parameter-Id	FB.03
Data length	1 byte
Notice	This parameter is identical with the CAN Parameter Watchdog activation and exists only in set 0.

Watchdog inhibit

Parameter address	**0284h**
Parameter-Id	FB.04
Data length	1 byte
Notice	This parameter is identical with the CAN Parameter Watchdog inhibit (s. o.) and exists only in set 0.

HS_PDO_Index

Parameter address	**0285h**
Parameter-Id	FB.05
Notice	This parameter corresponds to the operator parameter HS_PDO_Index and exists only in set 0.

DSP402_OpMode

Parameter address	**0286h**
Parameter-Id	FB.06
Data length	4 byte
Notice	This parameter corresponds to the operator parameter DSP402_ModesOfOperation and exists only in set 0.

PD_In_Para_CobID

Parameter address	**0287h**
Parameter-Id	FB.07
Data length	4 Byte
Notice	This parameter corresponds to the CAN Parameter „nth transmit PDO Parameter, Cob ID“ * and exists in set 0 and set 1.

PD_In_Para_TxType

Parameter address	**0288h**
Parameter-Id	FB.08
Data length	1 byte
Notice	This parameter corresponds to the CAN Parameter „nth transmit PDO Parameter, TxType“ and exists in set 0 and set 1.

PD_In_Para_Inhibit

Parameter address	**0289h**
Parameter-Id	FB.09
Data length	2 byte
Notice	This parameter corresponds to the CAN Parameter „nth transmit PDO Parameter, Inhibit Time“ and exists in set 0 and set 1.

PD_In_Cycle

Parameter address	**028Ah**
Parameter-Id	FB.10
Notice	This parameter corresponds to the CAN Parameter PDIN_ Cycle_and exists in set 0 (PDIN1_Cycle_Time) and set 1 (PDIN2_Cycle_Time).

Nr_PDIn_Objs

Parameter address	**028Bh**
Parameter-Id	FB.11
Data length	1 byte
Notice	This parameter corresponds to the least significant byte (LSB) of the CAN parameter nth transmit PDO mapping, Nr Mapped Objects and exists in Set 0 and Set 1.

PD_Inx Index (with x = 1…4)

Parameter address	**028Ch,028Fh,0292h,0295h**
Parameter-Id	FB.12, 15, 18, 21
Data length	2 byte
Notice	This parameters correspond to the most significant word (Bit31...Bit16) of the parameter ‚nth tramsmit PDO mapping‘. Thus the index for the application object of the nth PDO mapping is defined. This parameters exist respectively in set 0 (PDO1) and set 1 (PDO2).

PD_Inx Set (with x = 1…4)

Parameter address	**028Dh,0290h,0293h,0296h**
Parameter-Id	FB.13, 16, 19, 22
Data length	1 byte
Notice	This parameters correspond to the third most significant word (Bit15...Bit8) of the parameter ‚nth tramsmit PDO mapping‘. Thus the set for the application object of the nth PDO mapping is defined. This parameters exist respectively in set 0 (PDO1) and set 1 (PDO2).

PD_Inx _BitDlen (with x = 1…4)

Parameter address	**028Eh,0291h,0294h,0297h**
Parameter-Id	FB.14, 17, 20, 23
Notice	This parameters correspond to the least significant word (Bit 7...Bit 0) of the parameter ‚nth tramsmit PDO mapping'. Thus the index for the application object of the nth PDO mapping is defined. This parameters exist respectively in set 0 (PDO1) and set 1 (PDO2).

PDOUT_WrMode

Parameter address	**0299h**
Parameter-Id	FB.25
Notice	This parameter corresponds to the CAN Parameter „PDOUT_WrMode“ and exists only in set 0.

PD_Out_Para_Co-bID

Parameter address	**029Ah**
Parameter-Id	FB.26
Data length	4 byte
Notice	This parameter corresponds to the CAN Parameter „nth Receive PDO Parameter, Cob ID“ and exits in set 0 and set 1.

PD_Out_Para_Tx-Type

Parameter address	**029Bh**
Parameter-Id	FB.27
Data length	1 byte
Notice	This parameter corresponds to the CAN-parameter „nth Receive PDO Parameter, TxType“ and exists in Set 0 and Set 1.

Nr_PDOut_Objs

Parameter address	**029Ch**
Parameter-Id	FB.28
Data length	1 byte
Notice	This parameters correspond to the least significant word (Bit 7...Bit 0) of the parameter ‚nth receive PDO mapping'. Thus the index for the application object of the nth PDO mapping is defined. This parameters exist respectively in set 0 (PDO1) and set 1 (PDO2).

PD_Outx Index (with x = 1…4)

Parameter address	**029Dh,02A0h,02A3h,02A6h**
Parameter-Id	FB.29, 32, 35, 38
Data length	2 byte
Notice	This parameters correspond to the most significant word (Bit 31...Bit 16) of the parameter ‚nth receivet PDO mapping‘. Thus the index for the application object of the nth PDO mapping is defined. This parameters exist respectively in set 0 (PDO1) and set 1 (PDO2).

PD_Outx Set (with x = 1…4)

Parameter address	**029Eh,02A1h,02A4h,02A7h**
Parameter-Id	FB.30, 33, 36, 39
Data length	1 byte
Notice	This parameters correspond to the third most significant byte (Bit 15...Bit 8) of the parameter ‚nth receive PDO mapping‘. Thus the set for the application object of the nth PDO mapping is defined. This parameters exist respectively in set 0 (PDO1) and set 1 (PDO2).

PD_Outx_BitDlen (with x = 1…4)

Parameter address	**029Fh,02A2h,02A5h,02A8h**
Parameter-Id	FB.31, 34, 37, 40
Notice	This parameters correspond to the least significant word (Bit 7...Bit 0) of the parameter ‚nth receivet PDO mapping‘. Thus the index for the application object of the nth PDO mapping is defined. This parameters exist respectively in set 0 (PDO1) and set 1 (PDO2).

ProcessData Inx (with x = 1…4)

Parameter address	**02AAh - 02ADh**
Parameter-Id	FB.42…45
Data length	2 byte
Meaning	x. Process input data word
Coding	Depending on the mapped parameter
Notice	This parameter is Read_Only and corresponds to the n. word of the PDO (tx) telegram on CAN and exists in set 0 and set 1.

ProcessData Outx (with x = 1…4)

Parameter address	**02AEh - 02B1h**
Parameter-Id	FB.46…49
Data length	2 byte
Meaning	x. Process output data word
Coding	Depending on the mapped parameter
Notice	This parameter is Read_Only and corresponds to the n. word of the PDO (rx) telegram on CAN and exists in set 0 and set 1.

Take Stored PD-Map

Parameter address	**02B2h**
Parameter-Id	FB.50
Notice	This parameter corresponds to the CAN Pparameter PD_Stored (Index = 5FE2h), (see above) and exists only in set 0.

Check PD Setting

Parameter address	**02B3h**
Parameter-Id	FB.51
Data length	1 byte
Meaning	Indicates whether the last adjusted PD assignment change was executed error-free.
Coding	0 : Error occurred in the last PD-assignment change. 255d : Last PD assignment change was executed error-free.
Notice	This parameter is not available on CAN and exists only in set 0.

ComCycle

Parameter address	**02B4h**
Parameter-Id	FB.52
Notice	This parameter corresponds to the CAN-parameter Com-Cycle and exists only in Set 0.

HS_SyncToutDelay

Parameter address	**02B5h**
Parameter-Id	FB.53
Notice	This parameter corresponds to the CAN-parameter HS_ SyncToutDelay and exists only in Set 0.

LifeGuardTout.Addr

Parameter address	**02B6h**
Parameter-Id	FB.54
Notice	This parameter corresponds to the CAN-parameter LifeGuardTout.Addr and exists only in Set 0.

LifeGuardTout.Data

Parameter address	**02B7h**
Parameter-Id	FB.55
Notice	This parameter corresponds to the CAN-parameter LifeGuardTout.Data and exists only in Set 0.

EmergencyCycle

Parameter address	**02B8h**
Parameter-Id	FB.56
Notice	This parameter corresponds to the CAN-parameter EmergencyCycle and exists only in Set 0.

Save_VLRamps

Parameter address	**02B9h**
Parameter-Id	FB.57
Notice	This parameter corresponds to the CAN-parameter Save_VLRamps and exists only in Set 0.

VL_Ramp_CalcMode

Parameter address	**02BAh**
Parameter-Id	FB.58
Notice	This parameter corresponds to the CAN Parameter VL_Ramp_CalcMode and exists only in set 0.

10. Change-over of the Transmission Type of the PDOs

The transmission type of the parameter 1st/2nd receive PDO parameter as well as of the 1st/2nd transmit PDO parameter is changeable. The valid values are:

- Asynchronous manufacturer-specific (Value = 254 = default)
- Asynchronous profile-specific (Value = 255)
- Synchronous acyclic (Value = 0)
- Synchronous cyclic (Value = 1)
- Synchronous cyclic (Value = 1, 2…240)
- Synchronous RTROnly (Value = 252)
- Asynchronous RTROnly (Value = 253)

According to CANopen the values 0 to 240 possess identical behaviour at the PDO(rx). That means, that at the PDO(rx) the PDOUT-data are updated at every SYNC, regardless of the value the tx-type has. Please also keep in mind the influence of the parameter PDOUT_WrMode on the PDO1, PDO2(rx)-processing.

10.1 Asynchronous manufacturer-specific (Value = 254d/FEh) or asynchronous profile-specific (Value = 255d/FFh)

If in parameter 1st/2nd receive PDO Parameter the transmission-type is adjusted to one of these values, it means, that the process output data in status OPERATIONAL are transferred to the inverter-control on receipt of a valid PDO(rx)-telegram, if at least 1 Byte has changed. A valid PDO(rx)-telegram is a telegram on the corresponding identifier with a data length of > the data length that results from the PDO(rx)-mapping. In the standard case it means, that all telegrams on the OUT-identifier with a data length of > 4 Byte are accepted.
In status OPERATIONAL the process input data are also read cyclical by the FI-control. If the value 254d or 255d is adjusted in parameter 1st/2nd transmit PDO Parameter, it means that a PDO(tx)-telegram is transmitted to CAN, if the process input data have changed.

10.2 Synchronous acyclic (Value = 0) or synchronous cyclic (Value = 1...240)

If in parameter 1st/2nd receive PDO Parameter the transmission-type is adjusted to once of these values, it means, that the process output data in status OPERATIONAL are transferred to the inverter-control on receipt of a SYNC-telegram. Provided, that a valid PDO(rx)-telegram was received before.
For the parameter 1st/2nd transmit PDO Parameter the value transmission-type = 0/1 means, that in status OPERATIONAL a PDO(tx)-telegram is immediately transmitted on CAN after receiving a SYNC-telegram.
For all synchronous values of the tx_type applies, that the SYNC-telegram triggers the transmission of the appropriate PDO(tx) respectively the transmission of the PDO(rx). With the exact value it is determined how many SYNC-telegrams are necessary for it. With the values 0 and 1 every SYNC activates the corresponding event. The values 2 to 240 itself specify the number of required SYNC-telegrams. However, it must be observed that in this value range the behaviour of all PDO(rxx)-telegrams is identical with the value = 1.
Example:

PDO1(tx).tx_type	=	10 →	After 10 SYNC-telegrams the Slave transmits a PDO1(tx) on CAN with the current PDIN1-data.
PDO1(rx).tx_type	=	10 →	After everySYNC-telegram the current PDOUT1-data are forwarded.

10.3 Synchronous / asynchronous RTROnly (Value = 252, 253)

These values are only valid for Tx PDOs. The transmission of the current PDIN data via the corresponding PDO(tx) telegram on CAN is only started upon receipt of a remote frame request on the corresponding identifier.

11. Synchronous mode

In the synchronous mode the internal processing cycle of the CAN operator and the connected frequency inverter control is adjusted over the SYNC telegram on CAN. Shortest deceleration times and above all extremely little deviation in the deceleration times are the ultimate goal of this operating mode. It is achieved with simultaneous compatibility on CAN. But clear functional restrictions are connected with the synchronous mode. Though the general operation is maintained.
In the synchronous mode the highest priority lies on the fastest possible transfer of process data. The mapping of the process data is adjustable over the process data mapping and is only subject to the restrictions of the already known high-speed PDOs in the CAN operator.

Following conditions apply to the synchronous mode:
The synchronous mode operates only in OPERATIONAL status of the node.
Only the high-speed PDO may be active.
The PDO works in both directions synchronous.
The PDO mapping in both directions fulfills following conditions:

- Number of mapped parameters in both direction = 2 or 3
- The first mapping occupies 32-Bit
- Every further mapping occupies 16-Bit

The synchronous mode is activated by writing on the new parameter ComCycle with a value unequal zero.
In the synchronous mode the SYNC telegrams on CAN must be transmitted within the adjusted time (ComCycle). The maximal temporal deviation of two successive SYNC telegrams must not exceed approx. 80µs. If this maximal deviation is exceeded, it must be clarified in each individual case, whether the application functions in the desired manner. The CAN operator monitors the receipt of the SYNC telegrams. If no SYNC is received within the timeout time, the operator automatically switches back into the normal mode. The timeout time corresponds to four-times of the expected SYNC cycle time (ComCycle).

The necessary presettings are comprised in the following table:

Index	Subindex	Dlen	Value
5FE5h	0	1	0
1801h	1	4	8000XXXXh
1401h	1	4	8000XXXXh
1800h	2	1	0 or 1
1400h	2	1	0 or 1
1A00h	0	1	< 3
1A00h	1	4	XXXXXX20h
1A00h	2	4	XXXXXX10h
1A00h	3	4	XXXXXX10h
1600h	0	1	< 3
1600h	1	4	XXXXXX20h
1600h	2	4	XXXXXX10h
1600h	3	4	XXXXXX10h
1006h	0	4	multiple of 1000 (dec)

11.1 Functional restrictions in synchronous mode

In the synchronous mode all CAN-SDO jobs and jobs from the diagnostic interface are interlaced in the process data transfer. On this account only CAN-SDO accesses to parameters in the inverter control with subindex = 0 are possible. That means, parameters in the inverter can only be addressed in the set defined by the set indicator (Fr.09) (indirect set-addressing). Note, that every CAN telegram can shift the SYNC in time, even if the SYNC telegram has a very high priority due to its low identifier. Therefore, if at all possible, it should be abstained from any other CAN communication in the synchronous mode (Node-Guarding, SDO-commands, NMT-commands). Only PDO(rx) telegrams, PDO(tx) relegrams and the SYNC should be transmitted.

The keyboard is not processed in the synchronous mode. The display is static and shows ‚Synch'. The diagnostic interface continues to operate with similar restrictions that apply to the CAN-SDO communication: Parameters in the inverter can only be read or written over the HSP5 service = 1 with the set selection Byte = 1 (indirect set-addressing over Fr.09).

12. DSP402 Support

Please refer to the description of the parameter DSP402_Modes Of Operation to find out which DSP402 modes are supported.
The CAN in automation user group has published the version 2.0 of the DSP402 unit profile for drives on 26.07.2002. The KEB F5 CANopen interface connection supports a subset of the functions and parameters, that are defined in the DSP402. The CAN operator takes over the conversion of the DSP402 parameter into parameters of the inverter control. This conversion is, in part, complex and consequently intensive in running time. For that reason a mapping of such parameter, that must be transliterated on the high-speed PDO is not permitted in most cases. But the DSP402 parameters can be addressed over the SDO commands. Likewise, nearly all DSP402 parameters are mappable on the low-speed PDO .
Some of the parameters in the KEB F5 frequency inverter, that serve as basis for realized DSP402 parameters, are set-programmable. Since the DSP402 profile does not support any set-programming, the following provision was made for the DSP402 realization: All DSP402 profile parameters, that are converted to parameters in the frequency control, are stored in set 0. They have no influence on the parameters in other sets:

The DSP402 profile works exlusively in parameter set 0

12.1 Presettings for DSP402 operation

The DSP402 profile supports the differentiation of the ramps for clockwise and counter clockwise rotation. Therefore, the ramp times for clockwise and counter clockwise rotation must have the same values. That conditions the following presettings for the inverter control:

Parameter	Parameter address	Parameter set	Parameter value
oP.29	031Dh	set 0	-1
oP.31	31Fh	set 0	-1

For the operation over the DSP402 control and status word the following presettings are to be made in the inverter control:

Parameter	Parameter address	Parameter set	Parameter value
ud.01	0801h	set 0	440
oP.00	0300h	set 0	5
oP.01	0301h	set 0	6
oP.02	0302h	set 0	0
oP.60	033Ch	set 0	0
oP.61	033Dh	set 0	0
di.01	0B01h	set 0	Bit 0 = 1
di.02	0B02h	set 0	Bit 0 = 1
di.09	0B09h	set 0	2

12.2 Details to the DSP402 velocity ramps

The DSP402 profile defines a velocity ramp (VL ramp) as a structure developed from two parts:

- VL ramp.Dspeed : Delta speed value of the ramp in rpm
- VL ramp.Dtime : Delta time value of the ramp in seconds.

Internally in the inverter control a ramp is defined by a fixed part the so-called ramp reference value and an adjustable part the ramp time. The DSP402 ramp values are stored in the CAN operator. During a read access these buffer values are accessed. If one of the values is written an additional write access to the corresponding ramp time in the inverter control becomes necessary. The ramp values according to DSP402 coding are not automatically stored non-volatile in the operator. The user can explicitly request it over parameter Save_VL_Ramps.
The conversion of a VL ramp into a FI ramp time is clear. However, the likewise necessary conversion of a FI ramp time into a VL ramp is not clear. For that reason a method for the reversion of the conversion must be found. The KEB F5 CANopen operator supports different reversion modes, which are selectable by the parameter VL_ramp_CalcMode (see below).
Moreover, it must be noted that with each change of one of the two VL ramp parts a write access to the relevant FI ramp times is always effected. That means, that with the change of both VL ramp parts initially only one part is converted into a FI ramp time. At this point the FI ramp time does not correspond to the desired ramp. Only after the second VL ramp part has also been written, the desired ramp is preset as FI ramp time. This problem also applies to the low-speed PDO. The DSP402 profile does not provide any regulations for the consistent setting of VL ramps. The just illustrated problem must be solved by the user:

- For example, one of the two VL ramp times always remains unchanged and the ramp is varied only with the other part of the VL ramp.
- Another approach would be to never change the ramps while the FI drives ramps.

12.3 DSP402 profile and synchronous mode

It is generally possible to operate in the synchronous mode over DSP402 profile parameters, but it must be noted, that the synchronous mode does not support any conversion/reversion of parameter values. Therefore DSP402 parameters to be converted are not mappable on process data in the synchronous mode. The following parameters do not require a conversion:

- VL_TargetVelocity (Index=6042h)
- VL_ControlEffort (Index=6043h)

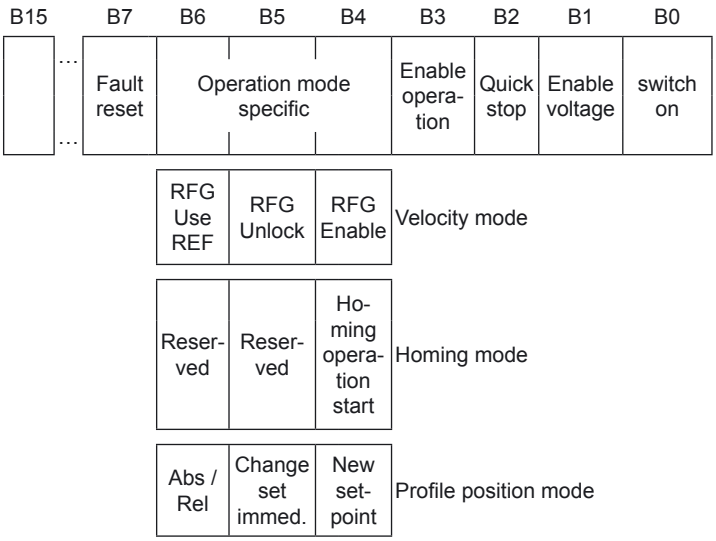
The direct set-addressed SDO access is also not permitted in the synchronous mode. Consequently the most SDO accesses on DSP402 parameters would be rejected with error.
For these reasons it is practically unrealistic to operate with DSP402 parameters in the synchronous mode.

12.4 General parameters of the DSP402 Profile

DSP402_ErrorCode

CAN-SDO-Index	**603Fh**			
SDO-Subindex	0			
Object type	Single variable (var)			
Data length	2 byte			
Access	READ_ONLY			
Meaning	Indicates the current error status of the unit.			
Coding	According DSP402 setting, see table in the annex			
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	NO	YES	NO	YES
Notice	Is mapped internally on the parameter RU.00. If the FI signals no error status, but the CAN operator has detected an error, it is returned.			

DSP402_Control-word

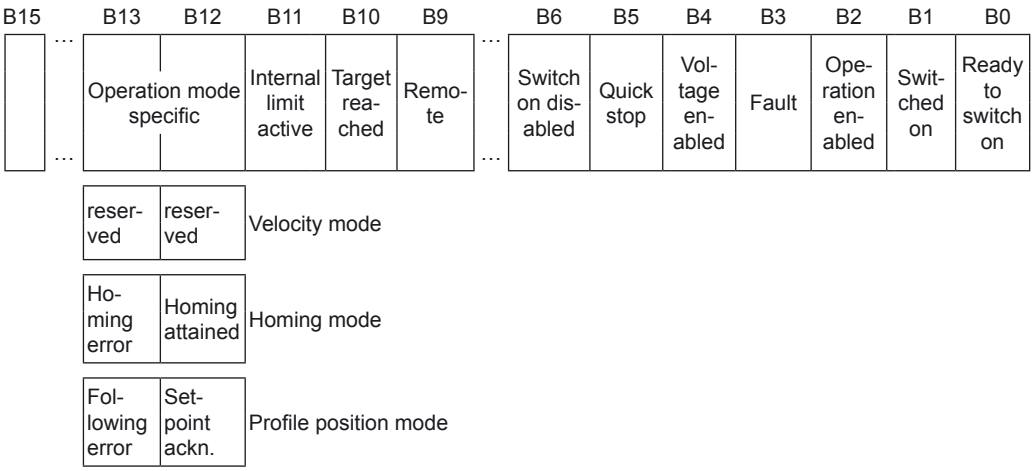
CAN-SDO-Index	**6040h**	
SDO-Subindex	0	
Object type	Single variable (var)	
Data length	2 byte	
Access	READ_WRITE	
Meaning	Serves for the setting of control commands. The parameter is bit-coded and is mapped in the inverter control on the parameter SY.50 (control word).	
Coding	Only the supported bits are listed in the following fig.:  The bits B6 to B4 are defined mode-dependent. Grey highlighted bits are currently not realized in the KEB CANopen interface connection.	
Permitted PDO mapping	High-Speed-PDO	Low-Speed-PDO
	rx / tx	rx / tx
	YES / YES	YES / YES
Notice	Is mapped internally on the parameter SY.50 .	

DSP402_Status-word

CAN-SDO-Index	**6041h**
SDO-Subindex	0
Object type	Single variable (var)
Data length	2 byte
Access	READ_ONLY
Meaning	Serves for the publication of the current condition. The parameter is bit-coded and mapped in the inverter control on parameter SY.51 (control word).

Coding

Only the supported bits are listed in the following fig.:



The bits B13, B12 are defined mode-dependent. Grey highlighted bits are currently not realized in the KEB CANopen interface connection.

Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	NO	YES	NO	YES
Notice	Is mapped internally on the parameter SY.51.			

DSP402_ModesOfOperation

CAN-SDO-Index	**6060h**
SDO-Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Serves for the setting of the desired DSP402 operations mode.
Coding	**(-1) : Manufacturer-specific** 0 : reserved **1 : Profile Position Mode*1** **2 : Velocity Mode** 3 : Profile Velocity Mode (not possible here) 4 : Torque Profile Mode (not possible here) 5 : reserved **6 : Homing Mode*1** 7 : Interpolated Position Mode (not possible here)
Default setting	(-1) : Manufacturer-specific
Permitted PDO mapping	not mappable
Notice	At present no real difference exists between the modes (-1) and 2.

*1: These modes are permitted for the control types ud.02 = 4, 5, 6, 8, 9, 10 only.

DSP402_ModesOfOperationDisp

CAN-SDO-Index	**6061h**
SDO-Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_ONLY
Meaning	Indicates the current DSP402 operations mode.
Coding	s. DSP402_ModesOfOperation
Default setting	(-1) : Manufacturer-specific
Permitted PDO mapping	not mappable

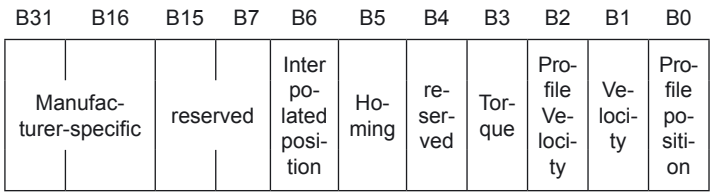
DSP402_AbortConn OptionCode

CAN-SDO-Index	**6007h**
SDO-Subindex	0
Object type	Single variable (var)
Data length	2 byte
Access	READ_WRITE
Meaning	Determines the behaviour after the connection to CAN was aborted. Abort of the connection to CAN is here synonymous with the following events: - Response of the Life Guarding timeout monitoring - BusOff condition of the CAN controller
Coding	**(-1)** : The behavior after an interruption on CAN is determined by the two parameters LifeGuardTout_Addr and LifeGuardTout_Data, see above **0** : No activity **1** : Change in PRE_OPERATIONAL, if actual state OPERATIONAL and explicitely release of the fieldbus watchdog (E.Bus) at the inverter control (if this watchdog is activated in the inverter). **2** : Change in PRE_OPERATIONAL, if actual state OPERATIONAL and command, Disable Voltage‘ via DSP402_Controlword. **3** : Change in PRE_OPERATIONAL, if actual state OPERATIONAL and command ‚Quick Stop‘ via DSP402_Controlword.
Default setting	(-1)
Permitted PDO mapping	not mappable
Notice	A value change is effective immediately and stored non-volatile.

DSP402_MotionProfileType

CAN-SDO-Index	**6086h**
SDO-Subindex	0
Object type	Single variable (var)
Data length	2 byte
Access	READ_WRITE
Meaning	Indicates the way of motion realization.
Coding	(-1) : Linear ramps with s-curves that can be activated 0 : Linear ramps (not adjustable here) 1 : sin2-ramps (not adjustable here) 2 : Jerk-free ramps (not adjustable here) 3 : Jerk-limited ramps (not adjustable here)
Default setting	(-1)
Permitted PDO mapping	not mappable
Notice	At present only one value can be selected.

DSP402_SuppDrive Modes

CAN-SDO-Index	**6502h**
SDO-Subindex	0
Object type	Single variable (var)
Data length	4 byte
Access	READ_ONLY
Meaning	Indicates bit-coded the supported modes of the CANopen interface connection .
Coding	 The manufacturer-specific bits are not used here.
Permitted PDO mapping	not mappable

12.5 Parameters of velocity mode

VL_TargetVelocity

CAN-SDO-Index	**6042h**			
SDO-Subindex	0			
Object type	Single variable (var)			
Data length	2 byte			
Access	READ_WRITE			
Meaning	Specifies the setpoint speed.			
Coding	1 rpm			
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	YES	YES	YES	YES
Notice	Is mapped internally on the parameter SY.52.			

VL_VelocityDemand

CAN-SDO-Index	**6043h**			
SDO-Subindex	0			
Object type	Single variable (var)			
Data length	2 byte			
Access	READ_ONLY			
Meaning	Specifies the speed value at the output of the ramp generator.			
Coding	1 rpm			
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	NO	YES	NO	YES
Notice	Is mapped internally on the parameter ru.02 .			

VL_ControlEffort

CAN-SDO-Index	**6044h**			
SDO-Subindex	0			
Object type	Single variable (var)			
Data length	2 byte			
Access	READ_ONLY			
Meaning	Indicates the actual speed value.			
Coding	1 rpm			
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	NO	YES	NO	YES
Notice	Is mapped internally on the parameter SY.53.			

VL_VelocityMin Amount

CAN-SDO-Index	**6046h**			
SDO-Subindex	1			
Object type	Single variable (var)			
Data length	4 byte			
Access	READ_WRITE			
Meaning	Indicates the amount of the lower limit value of the setpoint value.			
Coding	1 rpm			
Permitted PDO-mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	NO	NO	YES	YES
Notice	Is mapped internally on the parameter oP.06.			

VL_VelocityMax Amount

CAN-SDO-Index	**6046h**			
SDO-Subindex	2			
Object type	Single variable (var)			
Data length	4 byte			
Access	READ_WRITE			
Meaning	Indicates the amount of the upper limit value of the setpoint value.			
Coding	1 rpm			
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	NO	NO	YES	YES
Notice	Is mapped internally on the parameter oP.10 .			

VL_VelocityAccele-ration.Dspeed

CAN-SDO-Index	**6048h**			
SDO-Subindex	1			
Object type	Single variable (var)			
Data length	4 byte			
Access	READ_WRITE			
Meaning	Specifies together with the VL_VelocityAcceleration.Dtime the acceleration ramp.			
Coding	1 rpm			
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	NO	NO	YES	YES
Notice	Is mapped internally on the parameter oP.28.			

VL_VelocityAcceleration.Dtime

CAN-SDO-Index	**6048h**			
SDO-Subindex	2			
Object type	Single variable (var)			
Data length	2 byte			
Access	READ_WRITE			
Meaning	Specifies together with the VL_VelocityAcceleration. Dspeed the acceleration ramp.			
Coding	1 s4			
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	NO	NO	YES	YES
Notice	Is mapped internally on the parameter oP.28.			

VL_VelocityDeceleration.Dspeed

CAN-SDO-Index	**6049h**			
SDO-Subindex	1			
Object type	Single variable (var)			
Data length	4 byte			
Access	READ_WRITE			
Meaning	Specifies together with the VL_VelocityDeceleration.Dtime the deceleration ramp.			
Coding	1 rpm			
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	NO	NO	YES	YES
Notice	Is mapped internally on the parameter oP.30.			

VL_VelocityDeceleration.Dtime

CAN-SDO-Index	**6049h**			
SDO-Subindex	2			
Object type	Single variable (var)			
Data length	2 byte			
Access	READ_WRITE			
Meaning	Specifies together with the VL_VelocityDeceleration. Dspeed the deceleration ramp.			
Coding	1 s4			
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO	
	rx	tx	rx	tx
	NO	NO	YES	YES
Notice	Is mapped internally on the parameter oP.30.			

VL_VelocityQuick-Stop.Dspeed

CAN-SDO-Index	**604Ah**				
SDO-Subindex	1				
Object type	Single variable (var)				
Data length	4 byte				
Access	READ_WRITE				
Meaning	Specifies together with the VL_VelocityQuickStop.Dtime the quick-stop ramp.				
Coding	1rpm				
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO		
	rx	tx	rx	tx	
	NO	NO	YES	YES	
Notice	Is mapped internally on the parameter Pn.60.				

VL_VelocityQuick-Stop.Dtime

CAN-SDO-Index	**604Ah**				
SDO-Subindex	2				
Object type	Single variable (var)				
Data length	2 byte				
Access	READ_WRITE				
Meaning	Specifies together with the VL_VelocityQuickStop.Dspeed the quick-stop ramp.				
Coding	1 s4				
Permitted PDO mapping	High-Speed-PDO		Low-Speed-PDO		
	rx	tx	rx	tx	
	NO	NO	YES	YES	
Notice	Is mapped internally on the parameter Pn.60.				

VL_QuickStopOpti-onCode

CAN-SDO-Index	**605Ah**
SDO-Subindex	0
Object type	Single variable (var)
Data length	2 byte
Access	READ_WRITE
Meaning	Specifies the behaviour of the quick stop.
Coding	The DSP402-profile determines the value range for the manufacturer-specific mode within the range –32768....-1. For this reason the value is converted in the operator into the coding of the parameter PN.58 as follows: Value = Amount (VL_QuickStopOptionCode) - 1 (Pn.58) The significance of the individual modes is to be taken from the application instruction of the used inverter control. -1 → Pn.58 = 0 -2 → Pn.58 = 1 -3 → Pn.58 = 2 etc.
Permitted PDO mapping	not mappable
Notice	Is mapped internally on the parameter Pn.58.

VL_PoleNr

CAN-SDO-Index	**604Dh**
SDO-Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_ONLY
Meaning	Specifies the number of poles of the motor. This value is calculated from the parameters dr.01 and dr.05 and needed for all conversions of speed (rpm) into frequency (Hz).
Coding	1
Permitted PDO-mapping	not mappable
Notice	Is calculated internally from the parameters dr.01 and dr.05.

VL_Ramp_Calc Mode

CAN-SDO-Index	**5FDBh**
SDO-Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Determines the mode of calculation for the conversion of a KEB ramp time into a DSP402 Velocity-ramp.
Coding	0 : Both parts of the VL ramp (Dspeed, Dtime) are determined in such a way that the values become as small as possible, but the accuracy of the ramp time which can be converted remains received. 1 : Only the VL-Rampe.Dtime is calculated, VL-Rampe. Dspeed remains unchanged. 2 : The value of the KEB-ramp time is accepted as value for the VL-ramp.Dtime. VL ramp.Dspeed is set accordingly.
Permitted PDO-mapping	not mappable
Notice	A value change is effective immediately and stored non-volatile.

Save_VL_Ramps

CAN-SDO-Index	**5FDCh**
SDO-Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_WRITE
Meaning	Serves for the non-volatile storing of the velocity-mode ramps in the CAN operator.
Coding	B7 B6 B5 B4 B3 B2 B1 B0 Save VL-QST. Dtime | Save VL-DEC. Dtime | Save VL-ACC. Dtime | Save VL-QST. Dspeed | Save VL-DEC. Dspeed | Save VL-ACC. Dspeed With VL-ACC. Dspeed : Index=6048h,Subindex=1 With VL-DEC. Dspeed : Index=6049h,Subindex=1 With VL-QST. Dspeed : Index=604Ah,Subindex=1 With VL-ACC.Dtime : Index=6048h,Subindex=2 With VL-DEC.Dtime : Index=6049h,Subindex=2 With VL-QST.Dtime : Index=604Ah,Subindex=2
Permitted PDO-mapping	not mappable
Notice	During reading the value 0 is always returned.

13. Factors

The DSP402 profile defines many parameters with so-called user units. In order to realize these parameters conversion factors must be available, that perform the conversion into internal quantities. For this purpose the profile specifies an own group of parameters the so-called factor group. The KEB DSP402 realization supports no parameter of this group. But the following factors for the conversion of units are supported, that are used at different DSP402 parameters.

Every factor consists of an unsigned 32-bit nominator and an unsigned 32-bit denominator. Every factor is defined as structure with three members (as described below). The exact calculation formula is listed in the description to the concerned parameters:

- **Factor 0** : Conversion of user-specific path units into path units used by the frequency inverter.
- **Factor 1** : Conversion of user-specific speed units into speed units used by the frequency inverter.
- **Factor 2** : Conversion of user-specific acceleration units into acceleration units used by the frequency inverter.

FactorX.NrEntries

CAN-SDO-Index	**5FC0h + X**
SDO-Subindex	0
Object type	Single variable (var)
Data length	1 byte
Access	READ_ONLY
Meaning	Number of members in the structure factor X
Coding	1
Permitted PDO mapping	not mappable
Notice	A value change is effective immediately and stored non-volatile.

FactorX.Numerator

CAN-SDO-Index	**5FC0h + X**
SDO-Subindex	1
Object type	Single variable (var)
Data length	4 byte
Access	READ_WRITE
Meaning	Reading factor X
Coding	1
Default setting	1
Permitted PDO mapping	not mappable
Notice	A value change is effective immediately and stored non-volatile.

FactorX.Divisor

CAN-SDO-Index	**5FC0h + X**
SDO-Subindex	2
Object type	Single variable (var)
Data length	4 byte
Access	READ_WRITE
Meaning	Reading factor X
Coding	1
Default setting	1
Permitted PDO mapping	not mappable
Notice	A value change is effective immediately and stored non-volatile.

13.1 Advanced conversions

For some parameters the conversion by a factor consisting of numerator and denominator is not sufficient. These conversions include reference values of the FI control for the speed. This is necessary, for example, at the conversion of an acceleration (delta speed/delta time) into a ramp time. In addition to that the reference values are dependent on the F5 control type (refer to description of the parameter ud.02 in the application manual of the inverter). The following list of reference values is stored in the CAN Operator:

ud.02 value	Speed-Reference Value(VRef)	Standardization
0	100	Hz
1	200	Hz
2	400	Hz
3	reserved	reserved
4	1000	rpm
5	2000	rpm
6	4000	rpm
7	reserved	reserved
8	1000	rpm
9	2000	rpm
10	4000	rpm

13.2 Example for the definition of the factors

For the optimal definition of the factor values applies the following:

- Limit the nominator and denominator of the factors, if possible, to 16-bit-width
- The part of the calculation, that is the most intensive regarding the running time, is the division by factorX. divisor. Therefore, if possible, this value is to be = 1.

13.2.1 Factor 0: User-path units in increments

For this conversion the values of the parameters Ec.01 (encoder line number encoder 1) respectively Ec.11 (encoder line number encoder 2) and Ec.07 (multiple evaluation encoder 1) respectively Ec.17 (multiple evaluation encoder 2) must be known. Here it is based on encoder 1.

If the factor 0 is to be defined for a setting in µm (linear motion), the following applies:

$$F0 = \frac{Ec.01 * 2^{Ec.07}}{U_{driving\ pulley}} \quad \text{with } U_{driving\ pulley} = \text{circumference of the driving pulley in µm}$$

→ Factor0.Numerator = $Ec.01 * 2^{Ec.07}$

→ Factor0.Divisor = $U_{driving\ pulley}$ = circumference of the driving pulley in µm

If the factor 0 is to be defined for the setting of 0.01 angular degree (rotary motion), the following applies:

→ Factor0.Numerator = $Ec.01 * 2^{Ec.07}$

→ Factor0.Divisor = $U_{driving\ pulley}$ = circumference of the driving pulley in µm

$$F0 = \frac{Ec.01 * 2^{Ec.07}}{36000}$$

→ Factor0.Numerator = $Ec.01 * 2^{Ec.07}$

→ Factor0.Divisor = 36000

13.2.2 Factor 1: User-speed units in 0,125 rpm

If the factor1 is to be defined for the setting resolution 0.1 rpm, the following applies:

→ Factor1.Numerator = 8

→ Factor1.Divisor = 10

If the factor1 is to be defined for the setting 1 µm/s, the following applies:

→ Factor1.Numerator = 480

→ Factor1.Divisor = $U_{driving\ pulley}$ = circumference of the driving pulley in µm

If the factor1 is to be defined for the setting 0,01 angular degree/s, the following applies:

→ Factor1.Numerator = 8

→ Factor1.Divisor = 600

13.2.3 Factor 2: User-acceleration units in one KEB ramp time

The definition of factor 2 is a little more complicated. Please note, that at KEB the acceleration-/deceleration parameters are defined as ramp time. Thus the conversion is more extensive in this case. The factor is to be demonstrated here on the example of the parameter HM_Homing_Acc, that is mapped on the parameter PS.20 in the F5 control:

To simplify the matter Factor 2 can at first be written as element. Then the calculation formula is as follows:

$$PS.20 = \frac{Vref}{HM\_Homing\_Acc} * Factor\ 2$$

Changed according factor 2:

$$Factor\ 2 = \frac{PS.20 * HM\_Homing\_Acc}{Vref}$$

Assumption:

- ud.02 = 4: F5-M, maximumg speed = 4000 rpm , Vref = 1000 rpm
- The parameter HM_Homing_Acc is to be resolved in min^{-2}

The value PS.20 = 100 represents an acceleration of 1000 rpm per second. This corresponds to a value of 60000 min^{-2}. If one enters the value 100 for PS.20 and the value 60000 for HM_Homing_Acc in the above equation, the result for factor 2 is:

$$Factor\ 2 = \frac{100 * 60000}{1000} = 6000$$

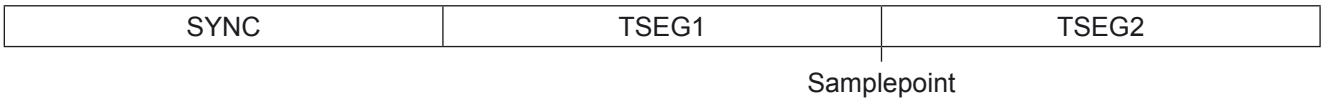
→ Factor2.Numerator = 6000

→ Factor2.Divisor = 1

14. Annex

14.1 CAN-Bit-Timing

Regarding the adjusted Bit-Timing the KEB CAN interface connection(s) adhere to the specifications of the CiA standard [2]:
The nominal Bit-Timing is as follows:



For all adjustable baud rates applies:

- t_q = Base time unit. All segments of the Bit-Timing result in a multiple of this time unit.
- SYNC = 0 → Only the edges from recessive to dominant are used for the synchronization.
- SJW = 0 → Synchronization step width = 1 * t_q
- TSEG2 = 1 → tSEG2 = 2 * t_q

Baud rate	Timequantum (t_q)	Samplepoint	TSEG1
10 Kbit/s	6,25 µs	14 * t_q = 87,5 µs	t_{SEG1} = 13 * t_q
20 Kbit/s	3,125 µs	14 * t_q = 43,75 µs	t_{SEG1} = 13 * t_q
25 Kbit/s	2,5 µs	14 * t_q = 35,0 µs	t_{SEG1} = 13 * t_q
50 Kbit/s	1,25 µs	14 * t_q = 17,5 µs	t_{SEG1} = 13 * t_q
100 Kbit/s	625 ns	14 * t_q = 8,75 µs	t_{SEG1} = 13 * t_q
125 Kbit/s	500 ns	14 * t_q = 7,0 µs	t_{SEG1} = 13 * t_q
250 Kbit/s	250 ns	14 * t_q = 3,5 µs	t_{SEG1} = 13 * t_q
500 Kbit/s	125 ns	13 * t_q = 1,625 µs	t_{SEG1} = 12 * t_q
800 Kbit/s	125 ns	7 * t_q = 1,25 µs	t_{SEG1} = 6 * t_q
1000 Kbit/s	125 ns	5 * t_q = 625 ns	t_{SEG1} = 4 * t_q

The grey highlighted transmission rates in the table are to be considered as particularly critical with regard to the line length. Moreover, the Bit-Timing for these bit rates deviates slightly from the ones recommended by [2].

14.1.1 Important warning notice

The KEB CAN interface connection has a potential-isolated CAN interface. The possible line length or the possible transmission rates are reduced by the additional delay elements (opto coupler) in the signal line. The possible line length or transmission rate depend on the delay times of all users in the CAN network. It is the responsibility of the customer to make an assessment concerning bit rate and possible line length. The necessary information for the KEB CAN interface connection are listed below:

Transmit-deceleration time of the CAN driver	: ≤ 80 ns.
Receive-deceleration time of the CAN driver	: ≤ 70 ns.
Transmit-deceleration time of used opto coupler	: ≤ 40 ns.
Receive-deceleration time of used opto coupler	: ≤ 40 ns.

Always select the smallest CAN transmission rate, that is needed for the processing of the process .

14.2 List of literature

[1] : Operating instructions frequency inverter control KEB COMBIVERT F5 with application manual.

[2] : Document to the agreement of the working committee Physical-Layer of CAN in Automation (CiA) user group: CiA/DS 102-1. publisher: CiA International Users and Manufacturers Group e.V., Am Weichselgarten 26, D-91058 Erlangen. Documents to the agreement of the working committee Higher-Layer-Protocols of CiA (publisher see above):

[3] : CiA/WG2/DS201 : CAN in the OSI Reference Model

[4] : CiA/WG2/DS202-1 : CMS Service Specification

[5] : CiA/WG2/DS202-2 : CMS Protocol Specification

[6] : CiA/WG2/DS202-3 : CMS Encoding Rules

[7] : CiA/WG2/DS203-1 : NMT Service Specification

[8] : CiA/WG2/DS203-2 : NMT Protocol Specification

[9] : CiA/WG2/DS204-1 : DBT Service Specification

[10] : CiA/WG2/DS204-2 : DBT Protocol Specification

[11] : CiA/WG2/DS207 : Application Layer Naming Conventions

[12] : CiA/DS301 V. 4.01 : Application Layer and Communication Profile from 01.06.2000

[13] : CiA/DSP402 V.2.0 : Device Profile Drives and Motion Control

14.3 Summary of the operator parameters according to CANopen

Index	Name	Object type	Subindex	Data Length in Byte	Access
1000h	device type	VAR	0	4	ro
1001h	error register	VAR	0	1	ro
1002h	Manufacturer Status Register	VAR	0	4	ro
1003h	Predefined ErrorField	ARRAY	1 up to max. 5	4	rw
1006h	ComCycle	VAR	0	4	rw
1008h	Manufacturer Device Name	VAR	0	4	ro
1018h	Identify Object	RECORD			
1400h	1st receive PDO Parameter	RECORD			
1400h	Number of supported entries	VAR	0	1	ro
1400h	COB-ID	VAR	1	4	rw
1400h	transmission type	VAR	2	1	rw
1401h	2nd receive PDO Parameter	RECORD			
1401h	Number of supported entries	VAR	0	1	ro
1401h	COB-ID	VAR	1	4	rw
1401h	transmission type	VAR	2	1	rw
1600h	1st receive PDO Mapping	RECORD			
1600h	Number of mapped objects	VAR	0	1	rw
1600h	nth object to be mapped	VAR	1 up to max. 4	4	rw
1601h	2nd receive PDO Mapping	RECORD			
1601h	Number of mapped objects	VAR	0	1	rw
1601h	nth object to be mapped	VAR	1 up to max. 4	4	rw
1800h	1st transmit PDO Parameter	RECORD			
1800h	Number of supported entries	VAR	0	1	ro
1800h	COB-ID	VAR	1	4	rw
1800h	transmission type	VAR	2	1	rw
1800h	Inhibit time	VAR	3	2	rw
1801h	2nd transmit PDO Parameter	RECORD			
1801h	Number of supported entries	VAR	0	1	ro
1801h	COB-ID	VAR	1	4	rw
1801h	transmission type	VAR	2	1	rw
1801h	Inhibit time	VAR	3	2	rw
100Ah	Manufacturer Software Version	VAR	0	4	ro
100Ch	Guard Time	VAR	0	2	rw
100Dh	Life Time Factor	VAR	0	1	rw
1A00h	1st transmit PDO Mapping	RECORD			
1A00h	Number of mapped objects	VAR	0	1	rw

further on next side

Index	Name	Object type	Subindex	Data Length in Byte	Access
1A00h	nth object to be mapped	VAR	1 up to max. 4	4	rw
1A01h	2nd transmit PDO Mapping	RECORD			
1A01h	Number of mapped objects	VAR	0	1	rw
1A01h	nth objects to be mapped	VAR	1 up to max. 4	4	rw
5FDAh	Watchdog Activation	VAR	0	1	rw
5FDBh	VL_Ramp_CalcMode	VAR	0	1	rw
5FDCh	Save_VL_Ramps	VAR	0	1	rw
5FDDh	EmergencyCycle	VAR	0	2	rw
5FDEh	LifeGuardTout.Data	VAR	0	4	rw
5FDFh	LifeGuardTout.Addr	VAR	0	4	rw
5FE0h	HS_SyncToutDelay	VAR	0	2	rw
5FE2h	PD_Stored	VAR	0	1	rw
5FE3h	OP_NodeId	VAR	0	1	rw
5FE4h	PDOUT_WrMode	VAR	0	1	rw
5FE5h	HS_PDO_Index	VAR	0	1	rw
5FE6h	PDIN1_Cycle_Time	VAR	0	2	rw
5FE7h	PDIN2_Cycle_Time	VAR	0	2	rw
5FF9h	Watchdog Inhibit	VAR	0	1	rw
5FECh	CAN_Baud2	VAR	0	1	rw
5FFEh	SAVE_CAN_Baud	VAR	0	1	rw
5FFFh	CAN_Baud	VAR	0	1	rw

14.4 Compact summary of CAN communication

Fixed identifier allocation:

SDO(rx) Identifier	=	1536 + Node_Id	:	SDO request to KEB FI
SDO(tx) Identifier	=	1408 + Node_Id	:	SDO acknowledgement from KEB FI
PDO1(rx) Identifier	=	512 + Node_Id	:	Process data to KEB FI
PDO1(tx) Identifier	=	384 + Node_Id	:	Process data from KEB FI
PDO2(rx) Identifier	=	768 + Node_Id	:	Process data to KEB FI
PDO2(tx) Identifier	=	640 + Node_Id	:	Process data from KEB FI
Node Guarding Identifier	=	1792 + Node_Id		
Emergency Identifier	=	128 + Node_Id	:	Emergency message from KEB FI

The most important NMT commands (telegrams) to identifier = 0

Start_Remote_Node

01h	Node-Id
B0	B1

Enter_Pre_Operational_State

80h	Node-Id
B0	B1

Reset_Node

81h	Node-Id
B0	B1

The most important values of the node state:

PRE_ OPERATIONAL	=	7Fh	:	Communication active except for the PDO's
OPERATIONAL	=	05h	:	Complete communication active

Examples for SDO communication:

Reading of parameter reference setting (oP.03) in set 4 → Index = 2303h, subindex = 10h

SDO(rx) telegram (to KEB FI)

40h	03h	23h	10h	XXh	XXh	XXh	XXh
B0	B1	B2	B3	B4	B5	B6	B7

SDO(tx) telegram (from KEB FI)

43h	03h	23h	10h	E8h	03h	00h	00h
B0	B1	B2	B3	B4	B5	B6	B7

In this example the read value is = 1000 (03E8h)

Write value = 5 to parameter reference source (oP.00) in set 0 → Index = 2300h, subindex = 01h

SDO(rx) telegram (to KEB FI)

22h	00h	23h	01h	05h	00h	00h	00h
B0	B1	B2	B3	B4	B5	B6	B7

SDO(tx) telegram (from KEB FI)

60h	00h	23h	01h	05h	00h	00h	00h
B0	B1	B2	B3	B4	B5	B6	B7

Example for the setting of new process data with the PDO1(rx) telegram:

The standard process data assignment is assumed here.

The parameter	control word	(SY.50) shall receive	the value	=	1,
The parameter	setpoint speed	(SY.52)	the value	=	1500 (05DCh).

PDO1(rx) telegram (to KEB FI)

01h	00h	DCh	05h

Memo

Karl E. Brinkmann GmbH
Försterweg 36-38 • D-32683 Barntrup
fon: +49 5263 401-0 • fax: +49 5263 401-116
net: www.keb.de • mail: info@keb.de

KEB worldwide…

KEB Antriebstechnik Austria GmbH
Ritzstraße 8 • **A**-4614 Marchtrenk
fon: +43 7243 53586-0 • fax: +43 7243 53586-21
net: www.keb.at • mail: info@keb.at

KEB Antriebstechnik
Herenveld 2 • **B**-9500 Geraadsbergen
fon: +32 5443 7860 • fax: +32 5443 7898
mail: vb.belgien@keb.de

KEB Power Transmission Technology (Shanghai) Co.,Ltd.
No. 435 QianPu Road, Songjiang East Industrial Zone,
CHN-201611 Shanghai, P.R. China
fon: +86 21 37746688 • fax: +86 21 37746600
net: www.keb.cn • mail: info@keb.cn

KEB Antriebstechnik Austria GmbH
Organizační složka
K. Weise 1675/5 • **CZ**-370 04 České Budějovice
fon: +420 387 699 111 • fax: +420 387 699 119
net: www.keb.cz • mail: info.keb@seznam.cz

KEB Antriebstechnik GmbH & Co. KG
Wildbacher Str. 5 • **D**–08289 Schneeberg
fon: +49 3772 67-0 • fax: +49 3772 67-281
mail: info@keb-combidrive.de

KEB España
C/ Mitjer, Nave 8 - Pol. Ind. LA MASIA
E-08798 Sant Cugat Sesgarrigues (Barcelona)
fon: +34 93 897 0268 • fax: +34 93 899 2035
mail: vb.espana@keb.de

Société Française KEB
Z.I. de la Croix St. Nicolas • 14, rue Gustave Eiffel
F-94510 LA QUEUE EN BRIE
fon: +33 1 49620101 • fax: +33 1 45767495
net: www.keb.fr • mail: info@keb.fr

KEB (UK) Ltd.
6 Chieftain Buisiness Park, Morris Close
Park Farm, Wellingborough **GB**-Northants, NN8 6 XF
fon: +44 1933 402220 • fax: +44 1933 400724
net: www.keb-uk.co.uk • mail: info@keb-uk.co.uk

KEB Italia S.r.l.
Via Newton, 2 • **I**-20019 Settimo Milanese (Milano)
fon: +39 02 33535311 • fax: +39 02 33500790
net: www.keb.it • mail: kebitalia@keb.it

KEB Japan Ltd.
15–16, 2–Chome, Takanawa Minato-ku
J–Tokyo 108-0074
fon: +81 33 445-8515 • fax: +81 33 445-8215
mail: info@keb.jp

KEB Korea Seoul
Room 1709, 415 Missy 2000
725 Su Seo Dong, Gang Nam Gu
ROK-135-757 Seoul/South Korea
fon: +82 2 6253 6771 • fax: +82 2 6253 6770
mail: vb.korea@keb.de

KEB RUS Ltd.
Krasnokazarmeny proezd 1,
RUS-111050 Moscow / Russia
fon: +7 445 695 3912 • fax: +7 495 645 3913
mail: info@keb.ru

KEB Sverige
Box 265 (Bergavägen 19)
S-43093 Hälsö
fon: +46 31 961520 • fax: +46 31 961124
mail: vb.schweden@keb.de

KEB America, Inc.
5100 Valley Industrial Blvd. South
USA-Shakopee, MN 55379
fon: +1 952 224-1400 • fax: +1 952 224-1499
net: www.kebamerica.com • mail: info@kebamerica.com

More and newest addresses at http://www.keb.de

© KEB	
Mat.No.	CCF50E0-K002
Rev.	1B
Date	01/2009