CAN-CBM-SIO
CAN-CBM-SIO4

CAN - RS-232,
RS-422, RS-485
or TTY Interface

Manual of the Module-Specific Software

N O T E

The information in this document has been carefully checked and is believed to be entirely reliable. **esd** makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. **esd** reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

esd assumes no responsibility for the use of any circuitry other than circuitry which is part of a product of **esd** gmbh.

esd does not convey to the purchaser of the product described herein any license under the patent rights of **esd** gmbh nor the rights of others.

esd electronic system design gmbh
Vahrenwalder Str. 205
D-30165 Hannover
Germany

Tel: +49-511-372-980
Fax: +49-511-372-981-98
Email: info@esd-electronics.com
Internet: http://www.esd-electronics.com

Manual file:	I:\TEXTE\DOKU\MANUALS\CAN\CBM\SIO-331\CSIO-10S.EN6
Date of setting copy:	14.07.1999

Software version:	sio4_V1.0aE0

Changes in the chapters

The changes in the user's manual listed below affect changes in the firmware as well as changes in the description of the facts only.

Manual Rev.	Chapter	Changes versus previous version
-	-	First revision
	-	-

Technical details are subject to change without notice.

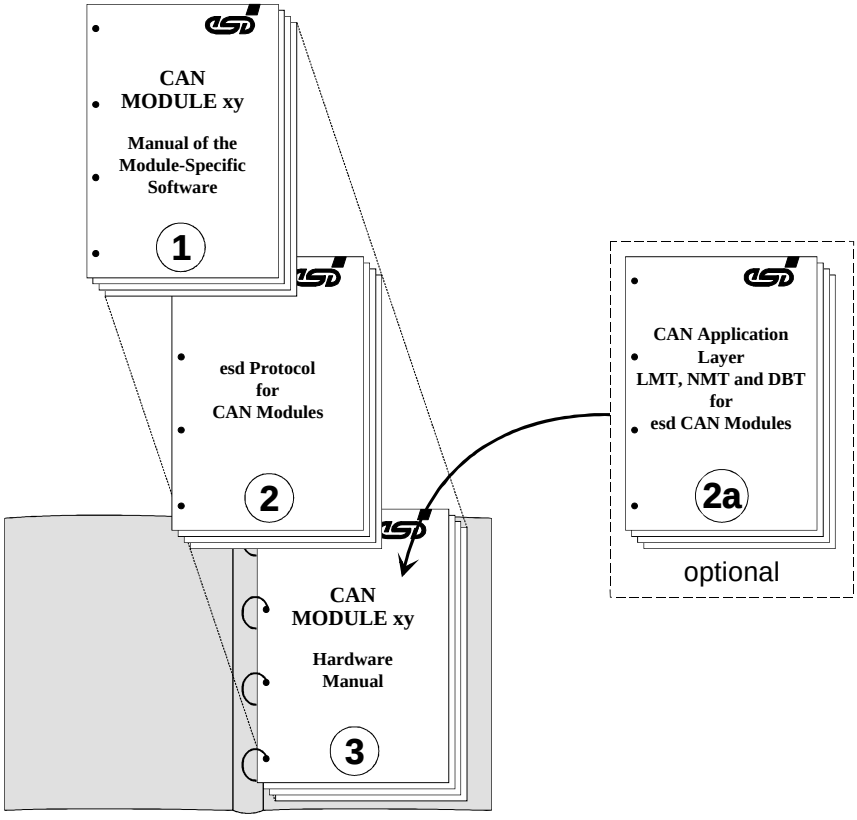
Content

Page

1. Overview

1.1 Which is Where?

The description of esd-CAN modules has been divided into three manuals, which are delivered together in one ring binder.



The first manual deals with software properties and parameters which are module-specific. This manual can therefore be used regardless of the CAN protocol you chose:

CAN-CBM-SIO CAN-CBM-SIO4
Serial Interfaces
Manual of the Module-Specific Software

This manual, for example, looks into the functions of the type-specific firmware, the identifier assignment and the assignment of user parameters.

In this manual the esd-CAN modules CAN-CBM-SIO4 and CAN-CBM-SIO will be described. Both manuals are generally the same, in contrast to the CAN-CBM-SIO4 module, however, the CAN-CBM-SIO module only has got one serial interface, which is at channel 1. Due to this, it only has got two CAN identifiers or two COB-IDs.
In the following the CAN-CBM-SIO4 module will be generally described. Differences to the CAN-CBM-SIO module will be described at the adequate points.

The second manual contains general software descriptions which are valid for all esd-CAN modules operated by the same protocol.

Two different protocols are available for the modules: The esd-CAN protocol and the CMS protocol. The protocols are independent from each other and are used alternatively. Depending on the implemented protocol, therefore, one of the following two manuals is valid for the module:

The esd-CAN protocol is described in the manual:

esd Protocol for CAN Modules

The protocol allows the user to set the esd-CAN modules by means of an initialisation identifier ($700). By means of this protocol identifiers can be assigned to the modules, user parameters can be set and watchdog functions can be activated.

Alternatively, the modules can be controlled via the CMS protocol. If this protocol has been implemented, you will have to consult the manual

CAN Application Layer LMT, NMT and DBT in esd Modules

for the CMS option. This manual explains the CMS services of
the Layer Management (LMT),
the Network Management (NMT) and
the Identifier Distributor (DBT)
in esd-CAN modules.

The third manual contains the hardware description of the module. It explains general as well as module-specific characteristics of the hardware. Here you can find subjects such as installation notes and connector assignments.

CAN-CBM-SIO CAN-CBM-SIO4
CAN - RS-232, RS-422, RS-485 or TTY Interface
Hardware Manual

1.2 Default Settings

The default settings of the manual are active, when one or more of the following conditions apply:

- A default RESET had been triggered on the module via the esd-CAN protocol.
- The data of the I²C-EEPROM are not OK (e.g. EEPROM is not equipped).
- The position of the coding switches after a RESET or power-on had been set to ‘00' and had then be changed to another value.

Individual parameters can be changed without influencing the default setting of other parameters. Changes in parameters are only retained after a RESET, if they had been stored in the EEPROM.

Default values when operating the module with the esd-CAN protocol	
INIT-Id.	in all operating modes $700
Rx-identifier, Tx-identifier	RxId1, TxId1 -< serial channel 1 RxId2, TxId2 -< serial channel 2 RxId3, TxId3 -< serial channel 3 RxId4, TxId4 -< serial channel 4 RxId5, TxId5 -< serial channel 5 The default values of the identifiers correspond to the settings via the coding switches. Please refer to the hardware manual for a detailed description of the settings. (*)
Module No.	= setting of the coding switches
CAN-bit rate	= 125 kbit/s

(*) Please note that the CAN-CBM-SIO4 module covers five Tx- and five Rx-identifiers. Therefore, the identifiers of the following modules have to be selected with an offset of at least +10, because otherwise the identifiers would clash!

Table 1.2.1: Default settings of the module when operated with the esd protocol

Attention: The terminal interface (at DSUB9) on the CAN-CBM-SIO4 module has been assigned with Rx-identifier RxId**5** and Tx-identifier TxId**5**. On the CAN-CBM-SIO module with only one serial interface, however, the terminal interface has been assigned with Rx-identifier RxId**1** and Tx-identifier TxId**1**.

Default values when operating the module with the CAL protocol	
Manufacturer name	ASCII 'esd_han'
Product name	has not been defined yet
Module-ID	= setting of coding switches
Module name	has not been defined yet
CAN-bit rate	125 KBIT/s
After a default RESET a *Configuration Download* to the module via the NMT protocol is absolutely necessary!	

Table 1.2.2: Default settings of the module when operated with CAL

Default values of user parameters (regardless of protocol used)		
First Tx-activate delay	10.000 msec	
CAN-Tx mode (all channels)	$1411, i.e.	'MinChar', 'MaxChar' = 1, 'Inhibit-Time' = 20 ms
Serial mode (all channels)	$2273, ie.	CTS* active (terminal interface: no CTS), 9600 baud, 2 stop bit (terminal interface: 1 stop bit) no parity, 8 bit/character

Table 1.2.3: Default settings of parameters of the module

Explanations of the terms for the user parameters in table 1.2.3:

First Tx-activate delay... Delay after a RESET before the module starts transmitting messages to the CAN or the serial interfaces.

CAN-Tx mode... 'MinChar' and 'MaxChar' determine the minimum and maximum number of data bytes which are to be transmitted within the CAN frame on the CAN-bus. ‘Inhibit time’ shows the delay of the CAN controller between the last successful transmission on the CAN-bus and the start of the following transmissions.

Serial mode... By means of serial mode the serial interfaces are set.

2. Description of the Data Transfer

The serial data are buffered between the CAN-bus and the serial interfaces in both data directions and by means of a 256 bytes sized toroidal-core store for each channel. The data being received first are also transmitted again first.

If the toroidal-core store is full and further data is received, it will be lost. Therefore, it is important to match the transmission rates of CAN-bus and serial interfaces with each other.

2.1 Serial Interfaces Transmit Data

The amount of CAN-bus data to be stored in the toroidal-core store per interval, has to be smaller than the amount of data transmitted by the serial interfaces.

The amount of received data per interval depends on the amount of data bytes transmitted, the frequency of transmissions, the bit rate of the CAN-bus and the assigned bus enabling of the CAN-bus.

If data is lost during operation, the intervals between transmission are to be prolonged or/and the amount of transmitted bytes has to be decreased.

The data is transmitted to the module by the CAN-bus via the Rx-identifiers or via COBs 1 to 4. The user is free to select the amount of bytes to be transmitted.

esd Prot. RxId	CAL COB	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
RxId 1	COB 1	Data for serial interface channel 1							
RxId 2	COB 2	Data for serial interface channel 2							
RxId 3	COB 3	Data for serial interface channel 3							
RxId 4	COB 4	Data for serial interface channel 4							
RxId 5	COB 5	Data for serial interface channel 5							

Table 2.1.1: Receiving the data to be transmitted via Rx-identifiers or COB 1 to 5 (CAN-CBM-SIO4)

Attention:
The terminal interface (at DSUB9) on the CAN-CBM-SIO4 module has been assigned with Rx-identifier RxId**5** and Tx-identifier TxId**5**. On the CAN-CBM-SIO module with only one serial interface, however, the terminal interface has been assigned with Rx-identifier RxId**1** and Tx-identifier TxId**1**.

Data Transfer

When operating with the esd protocol, the data output on the serial interfaces can be stopped by the supervisor command 'Suspend Module'. All four channels are stopped simultaneously. By means of the command 'Continue' the channels are started again simultaneously.

If the module is in 'Suspended' status, data being received from the CAN-bus and the serial interfaces will be stored in the toroidal-core store.

The data transfer from CAN-bus to serial interfaces principally has the following chronological course:

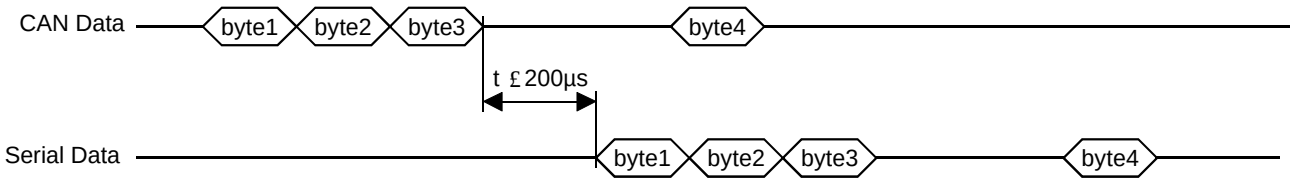


Fig. 2.1.1: Data transfer CAN-bus -> serial interface

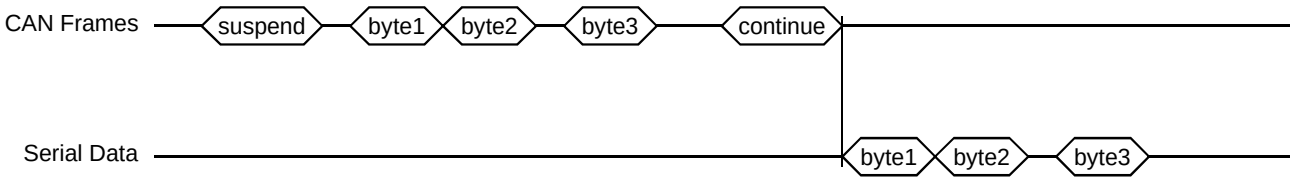


Fig. 2.1.2: Data transfer CAN-bus -> serial interface controlled by means of 'Suspend' and 'Continue'

2.2 Serial Interfaces Receive Data

The amount of serial interface data which is stored in the toroidal-core store per interval has to be smaller than the amount of data transmitted from the CAN-bus.

The CAN-bus limits the data flow via the bus demand (priority), the CAN-bus bit rate, the frequency of transmissions and the amount of transmitted bytes.

By means of parameters 'Inhibit-Time', 'MaxChar' and 'MinChar' the module offers the possibility to influence the last two factors. These parameters have been combined in the user parameter 'CAN-Tx-Mode'.

Via 'Inhibit-Time' the delay between two transmissions on the CAN-bus is determined.

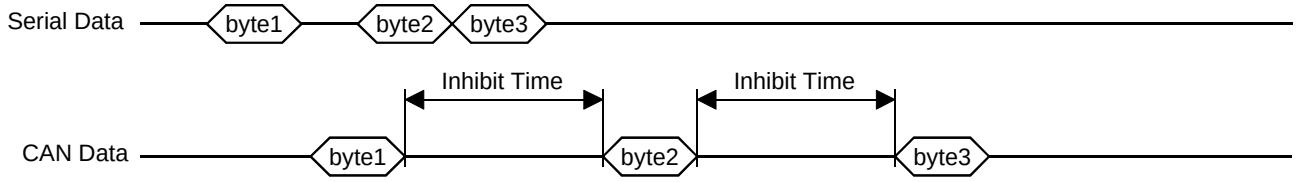


Fig. 2.2.1: Function of 'Inhibit-Time'

'MinChar' and 'MaxChar' determine the minimum and maximum number of data bytes which are to be transmitted on the CAN-bus within a CAN frame.

The data is transmitted by the module via Tx-identifiers or COBs 5 to 8 on the CAN-bus.

esd Prot.	CAL	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
TxId	COB								
TxId 1	COB 6	Data of serial interface channel 1							
TxId 2	COB 7	Data of serial interface channel 2							
TxId 3	COB 8	Data of serial interface channel 3							
TxId 4	COB 9	Data of serial interface channel 4							
TxId 5	COB 10	Data of serial interface channel 5							

Table 2.2.1: Transmission of data received on the serial interfaces via Tx-identifiers or COBs (CAN-CBM-SIO4)

The CAN-CBM-SIO module has only one Tx-identifier TxId1 or the COB 1.

3. User Parameters of the CAN-CBM-SIO and CAN-CBM-SIO4 Modules

By means of the user parameters the parameters for the serial interfaces and the parameters ‘Inhibit Time’, ‘MaxChar’ and ‘MinChar’ (CAN-Tx mode) for controlling the CAN-bus transmission rate are specified on the module.

If the module is operated with the esd protocol, the user parameters will be specified by the command 'Set User Parameters' ($86) on bytes 5 and 6 of the INIT-Id ($700).
All user parameters always have to be transmitted as 16-bit value with byte 5 as MSB!

If the CMS protocol has been implemented, the user parameters will be set via a configuration down-load (NMT).

The following table gives an overview of the user parameters of the CAN-CBM-SIO4 module. Only parameters $00...$02 are required for the CAN-CBM-SIO module.

User parameter No.	Parameter	Value range	Default settings
$00	First Tx-activate delay	$0000...$FFFF (0...65535 msec)	10.000 msec
$01	CAN-Tx mode channel 1	$0011...$FF88	$1411
$02	Serial mode channel 1	$0000...$88FF	$2273
$03	reserved	-	-
$04...$08	not assigned	-	-
$09	CAN-Tx mode channel 2	$0011...$FF88	$1411
$0A	Serial mode channel 2	$0000...$88FF	$2273
$0B	reserved	-	-
$0C...$10	not assigned	-	-
$11	CAN-Tx mode channel 3	$0011...$FF88	$1411
$12	Serial mode channel 3	$0000...$88FF	$2273
$13	reserved	-	-
$14...$18	not assigned	-	-
$19	CAN-Tx mode channel 4	$0011...$FF88	$1411
$1A	Serial mode channel 4	$0000...$88FF	$2273
$1B	reserved	-	-
$14...$20	not assigned	-	-
$21	CAN-Tx mode channel 5	$0011...$FF88	$1411
$22	Serial mode channel 5	$0000...$88FF	$2273
$23	reserved	-	-

Table 3.1.1: User parameters of the CAN-CBM-SIO/CBM-SIO4 module

3.1 First Tx-activate Delay (Parameter 0)

Parameter 0 specifies the delay before the module starts transmitting data to the CAN-bus and the serial interfaces after a RESET.

This delay is to secure that all modules operate stable on the CAN-bus before the module starts transmitting.

User parameter No. (= sub-command No.)	Parameter	Value range	Default setting
$00	First Tx-activate delay	$0000...$FFFF (0...65535 msec)	10.000 msec

Table 3.1.2: User parameter 0

3.2 CAN-Tx-Mode (Parameters 1, 9, 11, 19, 21)

By means of these parameters the 'Inhibit-Time' and parameters 'MaxChar' and 'MinChar' are specified.

Via 'Inhibit-Time' the delay between two transmissions on the CAN-bus is determined.

'MinChar' and 'MaxChar' determine the maximum and minimum amount of data bytes which are to be transmitted on the CAN-bus within a CAN frame. Starting from software-rev. '17.7e' the function os parameter ‘MinChar’ has been extended. A truncation status can now also be specified here.

User parameter No. (sub-command No.)	Parameter	Value range	Default setting
$01	CAN-Tx-mode chan. 1	$0011..$FF88	$1411
$09	CAN-Tx-mode chan. 2		
$11	CAN-Tx-mode chan. 3		
$19	CAN-Tx-mode chan. 4		
$21	CAN-Tx-mode chan. 5		

Table 3.2.1: User parameters CAN-Tx-mode

The two bytes of parameter CAN-Tx-mode are structured as follows:

	CAN-Tx-mode		
	Byte 5 of INIT-Id $700	Byte 6 of INIT-Id $700	
Parameter	Inhibit-Time	MaxChar	MinChar
Value range	$00...$FF	$1...$8	$1...$8

Table 3.2.2: Structure of parameter CAN-Tx-mode

Inhibit-Time..... The parameter Inhibit-Time specifies the time the CAN-controller waits after the last successful transmission of CAN-bus data, before it starts a new transmission. The entry is made in [ms].
The default setting is $14 (= 20 ms).

MinChar, MaxChar... In these two parameters the number of bytes from which a transmission to the CAN-bus is to be started, and the maximum amount of data bytes to be transmitted in a CAN frame are specified. The default setting is $11, i.e. each received byte is transmitted individually in one frame.

MinChar is only determined as described above, if values between $1...$8 are specified.
Values between $9 and $F change the functionality of the parameter: In this case the local software evaluates the entries of flags 'Carriage Return' ($0D) or 'Line Feed' ($0A) as transmission statuses. If one or both characters will be recognized, the data having been received by the serial interface will be transmitted to the CAN-bus. If eight bytes have been received before the flags have been received, the transmission will automatically start.
For the transmission you can also choose whether the flags are to be transmitted together with the data on the CAN-bus or not. The following table shows the various options:

MinChar [HEX]	Trans. status	Flag transmission to CAN	Comments
1...8	-	-	At least 1 to 8 bytes are always transmitted. Parameter MaxChar is also evaluated.
9	< Cr >	with < Cr >	Data is transmitted, if < Cr > had been received. < Cr > is also transmitted.
A	< Lf >	with < Lf >	As above, but with < Lf >.
B	< Cr >	without < Cr >	As above, but < Cr > is not transmitted.
C	< Lf >	without < Lf >	As above, but without < Lf >.
D	< Cr >	without < Cr > and without <Lf >	< Cr > and < Lf > can be at the end of the message. The data is transmitted after < Cr > had been received. Neither < Cr > nor < Lf > are transmitted with the data.
E	< Lf >	without < Cr > and without < Lf >	As above, but with < Lf >.
F	-	-	reserved

Table 3.2.3: Selection of transmission status via parameter MinChar

Examples:

1. For MinChar value $B has been selected. Via the serial interface the data 'abcd<Cr>' are received. The CAN-bus would transmit the data ‘abcd’ after having received <Cr>.

2. For MinChar value $E has been selected. Via the serial interface the data ‘abcd <Cr> <Lf>’ are received. The CAN-bus would transmit the data ‘abcd’ after having received <Lf>.

3.3 Serial Mode (Parameters 2, A, 12, 1A, 22)

The 'Serial Mode' user parameters set the bit rate, the stop bits, the number of bits/character and the CTS* locking and determine the parity evaluation of the serial channels.

User parameter No. (sub-command No.)	Parameter	Value range	Default setting
$02	Serial mode channel 1	$0000..$88FF	$2273
$0A	Serial mode channel 2		
$12	Serial mode channel 3		
$1A	Serial mode channel 4		
$22	Serial mode channel 5		

Table 3.3.1: User parameter Serial Mode

The two bytes of the parameter Serial Mode are structured as follows:

	Serial Mode		
	Byte 5 of INIT-Id $700		Byte 6 of INIT-Id $700
Parameter	Rx-Baudrate	Tx-Baudrate	Mode
Value range	$0...$8	$0...$8	$00...$7F

Table 3.3.2: Structure of parameter Serial Mode

Rx-bit rate,
Tx-bit rate... 4 bits each determine the bit rate with which data is transmitted (Tx) or received (Rx) on the serial interfaces. The default setting for Tx- and Rx-bit rate is 9600 KBIT/s. The physically attainable bit rate is limited by the hardware to a maximum of 38.4 kbit/s, when at the same time using all four channels.

Parameter Rx-(Tx) bit rate [HEX]	Bit rate [Bit/s]
0	38400
1	19200
2	**9600**
3	4800
4	2400
5	1200
6	600
7	300
8	150
9	7200
A	14400
B	28800
C	(57600)
D	(115200)
E	(2304000)
F	(76800)

Table 3.3.3: Setting the bit rate of the serial interfaces

Mode..... The bits of parameter Mode are assigned with the following functions:

Bit	7	6	5	4	3	2	1	0
Assign.	RTS-Mode	CTS*-enable	Stop-Bit	Parity-Mode		Parity-Type	Bits per Character	
Default	0	1	1	1	0	0	1	1

Table 3.3.4: Assignment of parameter Mode

Explanations of bits of parameter Mode

RTS-Mode..... Via this bit the RTS-modem mode for RS485-interfaces can be selected.

RTS-mode	Evaluation
0	RTS on Rx (default setting)
1	RTS-modem (RS485)

Table 3.3.5: Evaluation of RTS-mode bits

CTS* enable... Via this bit the CTS*-function of the serial controllers is enabled.
If the bit is '0', the CTS* signal will not be evaluated and the controller transmits the available data at once (provided the module is not in ‘Suspend’ status).
The evaluation of bits can be taken from the following table.

CTS* enable bit	CTS* -evaluation
0	CTS* -input is ignored (always at terminal interface)
1	CTS* -input is evaluated: hardware handshake (default setting for channel 1...4)

Table 3.3.6: Evaluation of 'CTS* enable' bits

The RS232 driver used activate the CTS* signal automatically, if the CTS* line of the serial interface is not connected.

At the terminal interface (DSUB9) the CTS input is not evaluated. The CTS* enable bit is insignificant for this interface, therefore.

Stop Bit..... Here the number of stop bits of the serial interface is determined:

Stop bit	Number of stop bits
0	1 stop bit (always at terminal interface)
1	2 stop bits (default setting for channel 1...4)

Table 3.3.7: Number of stop bits

The terminal interface (DSUB9) always operates with only one stop bit. The setting ‘1', therefore, is insignificant for this interface.

Parity Mode... By means of these two bits the evaluation of the parity bits is determined:

Parity mode		Evaluation
Bit 4	Bit 3	
0	0	parity evaluation when receiving data and transmitting the parity bit
0	1	reserved
1	0	no parity evaluation, no parity transmission (default setting)
1	1	reserved

Table 3.3.8: Parity evaluation

Parity Type... The polarity of the parity bit is determined by the parameter bit ‘Parity Type’:

Parity Type	Polarity
0	'even' (default setting)
1	'odd'

Table 3.3.9: Setting the polarity

Bits per Character..... Via these two bits the number of bits/character is selected:

Bits per Character		Number of transmitted bits
Bit 1	Bit 0	
0	0	5 (not at terminal interface)
0	1	6 (not at terminal interface)
1	0	7
1	1	8 (default setting)

Table 3.3.10: Number of bits/character

For the terminal interface only the following combinations of ‘Bits per Character’ and ‘Parity Mode’ are permissible:

7 data bits +	1 parity bit	
8 data bits	no parity	(default setting)
8 data bits +	1 parity bit	

4. Examples

In this chapter the operation and initialization of a module which is operated with the ***esd-CAN-Protocol*** will be explained by means of some examples.

4.1 Operation with Default Parameters

4.1.1 Basic Conditions, Objective

A device is to be connected to channel 1 of the CAN-CBM-SIO4 which corresponds in parameters to the default setting of the CAN-CBM-SIO4:

- 9600 Baud
- 8 Bit/Character
- 2 Stop-Bits
- no Parity

The module has not been initialized yet.

The desired Tx-identifier for transmitting data to the serial interface is to be $15E. The Rx-identifier is to be $15F.

4.1.2 Procedure

4.1.2.1 Set Identifiers

The identifier is determined, for example, by the coding switches. It corresponds to the tenfold of the value set at the coding switches plus the identifier offset (note: The selection of any value is therefore not possible via the coding switches. If you wish to do so, the setting has to be made by means of the esd-CAN protocol).

With the specification for the identifier, given above, the coding switches have to be set to a value corresponding to $15E/$A.

$15E / $A = $23

At coding switch ‘HIGH’ the value ‘2’, and at coding switch ‘LOW’ the value ‘3’ is now set.

The offset results from the desired serial channel: TxId-offset = 0, RxId-offset=1

The position and function of the coding switches is described in the hardware manual of the module.

Examples

4.1.2.2 Transmitting the Data to the Serial Interface

Via Rx-identifier RxId1 data is transmitted to the module. The number of data bytes transmitted can be between 0 and 8. In this example the following 5 bytes are to be transmitted:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
$11	$22	$33	$44	$55

On the CAN-bus the bytes are transmitted on the Rx-identifier as follows:

RxId1	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
$15F	$11	$22	$33	$44	$55	-	-	-

Table 4.1.2: Transmission of data to be transmitted via Rx-identifier RxId1

Bytes 6...8 are not required.

4.1.2.3 Receiving Data from the Serial Interface

Via Tx-identifier TxId1 the data received by the serial interface is sent from the module to the CAN-bus. The module is operated with the default parameters and therefore each byte received is immediately transmitted. The delay between initiating the individual transmissions by the CAN-controller is 20 ms.

In this example the device connected at channel 1 is to transmit a block of 8 bytes with the following contents:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
$FF	$EE	$DD	$CC	$BB	$AA	$99	$88

The module now transmits the individual bytes of this block on the Tx-identifier with delays of about 20 ms. If the CAN-bus is occupied with messages of a higher priority, the delay between the transmissions can be longer.

TxId1	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
$15E	$FF	-	-	-	-	-	-	-

:
t $ 20ms
:

TxId1	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
$15E	$EE	-	-	-	-	-	-	-

:
t $ 20ms
:

TxId1	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
$15E	$DD	-	-	-	-	-	-	-

:
...
:
t $ 20ms
:

TxId1	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
$15E	$88	-	-	-	-	-	-	-

Table 4.1.3: Transmitting received serial data to the CAN-bus

Bytes 2...8 of the individual Tx-transfers are not transmitted, because value ‘1' is set for ‘MaxChar’ in default setting.

4.2 Changing the Bit Rate

The bit rates of channel 1 are to be increased from 9600 baud (default setting) to 19200 baud for receive and transmission data.

The module No. of the CAN-CBM-SIO4 corresponds to the default setting of the coding switch setting and is therefore $23.

The bit rate is changed in the cells 'Rx-Baudrate' and 'Tx-Baudrate' of user parameter 'Serial-Mode'. These two cells are assigned to the first byte of the user parameter (at transmission = byte 5). For channel 1 this parameter is selected by means of sub-command $02.

The assignment of bit rate 19200 results in the values ‘$1' for each nibble of byte 5 of the user parameter.

Byte 6 of the parameter (Serial-Mode) should not be changed, has to be transmitted as well, however, as have all user parameters! Therefore, the default value $73 is entered here.

CAN-Id	Byte 1 (command)	Byte 2 (sub-command)	Byte 3 (always $00)	Byte 4 (module No.)	Byte 5 (bit rate)	Byte 6 (serial mode)
$700 =INIT-Id	$86	$02	$00	$23	$11	$73

Table 4.2.1: Changing the bit rates of serial channel 1

Byte 7 and 8 are not required for this command.

The changed bit rate is always active immediately after the user parameter has been received.