CAN - CSC595/2

CAN - S5 PLC Interface Module

Software Manual

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: +49-511-633-650
E-mail: info@esd-electronics.com
Internet: http://www.esd-electronics.com

Manual file:	I:\TEXTE\DOKU\MANUALS\CAN\CSC595.2\ENGLISCH\CS592-2S.EN6	
Date of print:	27.01.99	

Software described:	CANopen	CSC515 Kernel
Revision:	V0.2l	V0.2g

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.

Firmware version	Chapter	Changes versus Rev. 1.0 of the manual
-	2.	Chapter 'Configuration Program 5x5cfg' inserted.
	-	-

Contents

Page

1. Overview . . . 3
1.1 On this Manual . . . 3
1.2 Software Operation . . . 3
1.3 Additionally Implemented Functions . . . 5
1.4 Rules for Setting CAN-PLC-Interface Parameters . . . 6

2. Configuration Program 5x5cfg . . . 7
2.1 Overview . . . 7
2.2 Configuring . . . 7
2.3 Selecting the Hardware to be Configured . . . 8
2.4 Online / Offline Configuration . . . 9
2.5 Configuring the CSC5x5 . . . 10
2.5.1 Selecting the Number Format . . . 11
2.5.2 Parameter Setting . . . 12
2.5.2.1 Slot Number (only CSC595/2) . . . 12
2.5.2.2 Module Number . . . 12
2.5.2.3 Wakeup Time . . . 12
2.5.2.4 CAN-Baudrate . . . 13
2.5.2.5 Option Flags . . . 15
2.5.3 User-Parameter-Entries . . . 17
2.5.3.1 Adding a User-Parameter Entry . . . 17
2.5.3.2 Inserting a User-Parameter Entry . . . 17
2.5.3.3 Editing a User-Parameter Entry . . . 18
2.5.3.4 Deleting a User-Parameter Entry . . . 18
2.5.3.5 Deleting All User-Parameter Entries . . . 18
2.5.3.6 Requesting Information About Wrong User Parameters . . . 18
2.6 The User-Parameter Window . . . 19
2.7 Loading the Configuration from the CSC5x5 . . . 23
2.8 Transmitting the Configuration to the CSC5x5 . . . 23
2.9 Store the Configuration File in PC . . . 23
2.10 Load Configuration File from PC . . . 23
2.11 Printing a Configuration File . . . 24
2.12 Example . . . 25

3. Configuration via Terminal . . . 33
3.1 Terminal Setup . . . 33
3.2 Procedure . . . 33
3.3 Overview of Commands Implemented . . . 34
3.4 Syntax . . . 34
3.5 Command Descriptions . . . 35
3.5.1 Resetting the Module to Default Parameters . . . 35
3.5.2 Set and Display Bit Rate . . . 36
3.5.3 Setting the PLC-Slot Number of the Module . . . 39
3.5.4 Setting General Flags . . . 40

Contents

Page

3.5.5 Set and Display Parameters 41
3.5.6 Define Module No. 45
3.5.7 Set Wake-up Time after Start 46
3.5.8 Save Configuration 47
3.5.9 Reset Module 48
3.6 Example 49

4. Communication via the Virtual CAN Buffer 51
4.1 Internal Virtual CAN Buffers 51
4.2 Communication Window 55
4.3 Assignment of Bytes of the Communication Window 55
4.4 Command and Sub-Command in the Communication Window 56
4.5 Status Bits 60
4.6 Examples 61
4.6.1 Configuration Example 61
4.6.2 Transmitting Data to the CAN bus 62
4.6.3 Reading Data from the CAN bus 64

5. Sequential Data Exchange - Rx-FIFO 67
5.1 Rx-FIFO 67
5.1.1 Overview 67
5.1.2 Operation 67
5.2 Tx-FIFO 69

6. Default Settings/Parameters after a RESET 71
6.1 Performance after RESET 71
6.2 Default Settings when Module is Shipped 72
6.3 Re-establishing the Default Parameters (Default Setting) 73

Appendix 75
A.1 Loading CSC595/2-Firmware Updates 75
A.1.1 Connecting the CSC595/2-Module to the PC 75
A.1.2 Procedure 75

1. Overview

1.1 On this Manual

This manual describes the local firmware of the CSC595/2 module. The firmware has been designed for the purpose of running the module in a CANopen network. Doing this, the module can adapt either slave as well as master functionalities.

Furthermore the PC program '5x5cfg' will be described, which offers a comfortable user surface for configuring the CSC595/2.

After a short introduction into the way the module operates, the configuration via the PC program 5x5cfg will be described, followed by the description of a configuration via a terminal instead of a PC. The commands and parameters for both methods are generally the same. **The configuration via a PC, however, is much more comfortable and therefore preferable!**

1.2 Software Operation

By means of the communication processor CSC595/2 SIEMENS-PLC of models S5-95U, S5-100U or ET-100 and CAN-I/O modules or other CAN participants can directly be linked.

The CSC595/2 module assigns PLC input/output addresses to CAN identifiers. It simulates one or more PLC-input/output units. Data which has been received via the CAN bus (RxIds) is made available to the PLC as input data and PLC output data is transmitted to the CAN bus if a change occurs (TxIds). The data is transmitted unfiltered between PLC and CAN bus. Only one configuration is required, which is stored into an internal I²C-EEPROM. The data is then automatically exchanged in all further operations. The configuration can be made for instance via the PC program 5x5cfg, which will be described in detail in the following chapter from page 7.

The parameters set are not active at once, because they are only evaluated when module **and** PLC are booted. This is therefore an 'Offline Configuration' at whose end the parameters set have to be stored into the internal I²C-EEPROM via the command 'Save Configuration'. The parameters are then active after the following booting of the system.

The module guarantees a complete transparency of process data to the PLC programmer. No further function or data components are required so that PLC programs can be run in the usual way.

After having been configurated, the firmware is running in the background, 'invisible' to the PLC user. The CAN-I/O devices have now been assigned with PLC addresses under which they can be addressed by the PLC like PLC-peripheral units. The PLC can be programmed as usual. In order to minimize the CAN-bus load, data will only be transmitted if changes occur.

Overview

The following figure exemplary represents a configuration table with five entries. The arrows are used to clarify the data direction and the relation between PLC address and identifier.

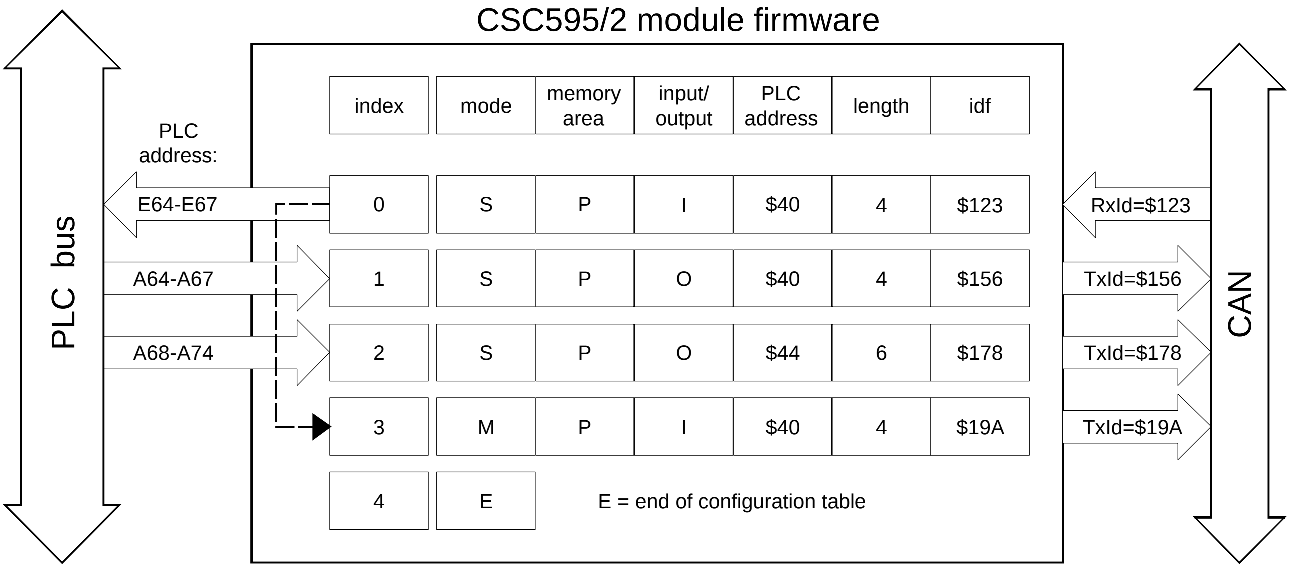


Fig. 1.1: Link of PLC and CAN data by means of the configuration table

The first three entries are ‘simulation links’: The CAN units represent themselves to the PLC CPU as PLC-I/O units. The fourth entry is a ‘monitor link’ to monitor the data which is transmitted to a digital peripheral unit.

The PLC addresses of the simulation must not interfere with other PLC addresses. This must also be considered when entering the data length.

1.3 Additionally Implemented Functions

Apart from simulating PLC-I/O units the CSC595/2 offers two further means of communication between PLC and CAN:

Virtual CAN-Buffer and Communication Window

Two bytes of data per cycle can be stored into a memory field of 256 bytes. The data can then be transmitted via defined CAN-identifiers. Rx data can be received and stored into this memory field. The data is read out again by two bytes per PLC cycle.

This function can be useful when reading or setting time inputs and outputs which are regardless of time, such as analog inputs and outputs. Furthermore, the communication window is required for handling the Rx FIFO.

The virtual CAN buffer and the communication window will be described in detail from page 51.

Sequential Data Exchange via Rx-FIFO and Tx-FIFO

The Rx-FIFO has been designed to enable received CAN-Rx data being handled sequentially. In contrast to the reception of Rx frames via the object buffer (ie. at simulation or in the virtual CAN buffer) old data is not overwritten with new data of the CAN ID each time.

The Rx-FIFO can buffer 1024 CAN-Rx telegrams.

A FIFO is useful in the following applications, for instance:

- sporadic reception of various CAN IDs, when an action is to be triggered because data has been received
- reception of fragmented data, such as data of a serial interface which sequentially transmits the data to be received on the same CAN identifier

The Tx-FIFO can buffer 255 CAN-Tx telegrams, which are then transmitted sequentially.

The Rx- and the Tx-FIFO will be described in detail from page 67.

1.4 Rules for Setting CAN-PLC-Interface Parameters

1	Each entry can have a **data length** of 1-8 bytes. The CAN telegrams (TxIds) transmitted by the CSC595/2 always have the length which has been specified. If telegrams received (RxIds) have more bytes than specified, the excess bytes will be ignored. If the telegrams have less bytes than specified, only the available data is taken from the telegram while the missing bytes remain unchanged.

2	The user-parameter entries can be assigned to any PLC address, provided that the PLC is able to support these addresses. The unit simulates for the PLC the whole area of addresses set. If the ‘virtual CAN buffer’ and/or the ‘Rx FIFO’ are being used, the first four PLC addresses will be assigned by the communication window! In this case those addresses should not be used for user-parameter entries!

3	If complete slots are not assigned up to the **maximum defined address**, these slots will be simulated as empty slots (length: 1 nibble).

4	The maximum **shift register length** of the CSC595/2 module is 192 nibbles (96 bytes).

2. Configuration Program 5x5cfg

2.1 Overview

5x5cfg is a menu-controlled program for PC, suitable for configuring CAN-S5 PLC interface modules CSC595, CSC595/2 and CSC515.
It can be operated under Windows 3.11, Windows NT 4.x and Windows 95.

This chapter will describe the individual points of the menu of 5x5cfg in detail. The descriptions will only be general, because the program can be used or various modules, as mentioned above. Restrictions and exceptional features for a module will be mentioned at appropriate passages in the text. An example will be given at the end of this chapter.

Principally the module can be directly configured via commands of the serial interface. The configuration via this tool, however, is much easier and more clearly. **Therefore we strongly advise you to configure the module by means of 5x5cfg!**

2.2 Configuring

The following steps have to be followed to configure the module:

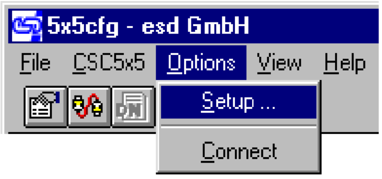
Command		Description
1	*Setup*	selection of a serial interface at PC (COM1, COM2,..)
2	*Connect*	when you want to configure ‚online‘
3	*Data Mapping*	opens dialog box
4	*Hex/Dec*	selection of data format for parameter definition
5	*Slot No.*	number of PLC slot selected
6	*Module No.*	is only required for CANopen
7	*Wakeup Timer*	set Wakeup timer
8	*Baudrate*	set CAN-bit rate
9	*Option Flags*	see corresponding chapter starting on page 15
10	*Add/New*	enter user parameters (one entry per identifier); here PLC addresses and CAN IDs are assigned; last entry: end mark
11	*Save to...*	configuration can be stored in the PC for filing
12	*Connect*	when you configured ‚offline‘

2.3 Selecting the Hardware to be Configured

Configuration Program 5x5cfg

In order to be able to configure the CSC5x5 you have to select the hardware to be configured, the CAN protocol and the serial interface to which the CSC5x5 is connected first. For this you have to proceed as follows:

1. Select ***Setup*** from menu ***Options*** (see the following figure) or click the Setup button in the tool bar.



Setup-
Button

Fig. 2.1: Select *Setup*

2. Now the dialog box shown in the following figure appears (example CSC595/2). In order to configure the CSC5x5 you have to enter the settings shown. The selection of the ***Serial Port*** depends on the interface available at your PC. If the serial interface COM1 is connecting a mouse to the PC, you should select COM2.
   The configuration program automatically sets the required parameters of the serial interface in your PC. Apart from the port selection (COM1, COM2...) no further settings are required.

   If the hardware to be configured is a **CSC515** with 2 CAN networks, you also have to set the number of CAN networks to 2.

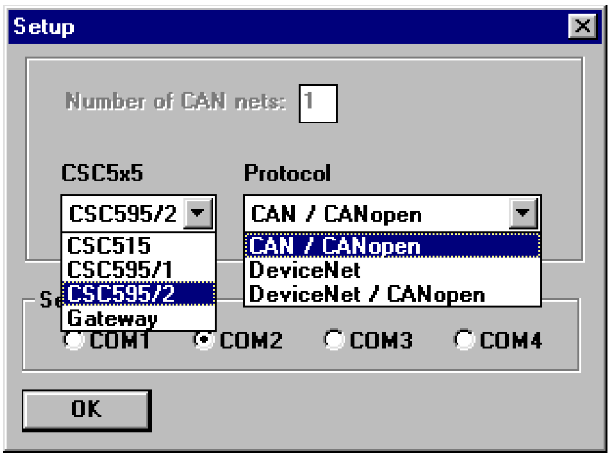


Fig. 2.2: Settings of menu *Setup*

3. By clicking the ***OK*** button or by striking the [Enter] key the selection of the hardware to be configured is completed.

These settings will be stored into the INI-file and will immediately be available when calling the program the following time.

2.4 Online / Offline Configuration

The CSC5x5 can either be configured in online or in offline operation. If you choose the online configuration, make sure that the PC and the CSC5x5 module are connected via the serial interface *during* the configuration. If you configure offline, you only have to establish the connection to the CSC5x5 for instance *after* entering all configuration parameters in the 5x5cfg program.

Online Configuration

For online configuration you have to select ***Connect*** from menu ***Options*** or click the Connect/Disconnect button in the tool bar.

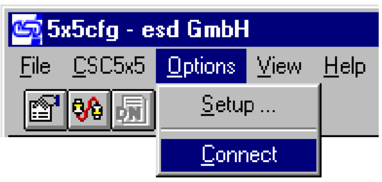
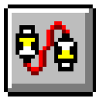


Fig. 2.3: Select *Connect*



Connect-
Button

When the connection to the CSC5x5 could be established, the message 'ONLINE' appears in the status line down on the right and the CSC5x5 can be configured as will be described in the following chapter.

Offline Configuration

In offline operation you have to configure the CSC5x5 as will be described in the following chapter. In order to transmit the configuration to the CSC5x5 after the offline configuration you have to execute the command ***Save to...*** from the ***CSC5x5*** menu. The CSC5x5 menu will be described in more detail in one of the following passages.

2.5 Configuring the CSC5x5

In order to configure the CSC5x5 you have to select ***Data Mapping*** from the menu ***CSC5x5*** or click the Data-Mapping button in the tool bar.

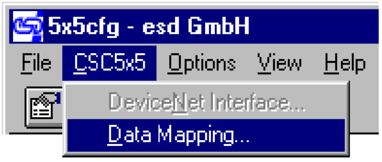
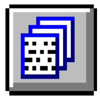


Fig. 2.4: Select *Data Mapping*



Data Mapping-Button

The following dialog box will open:

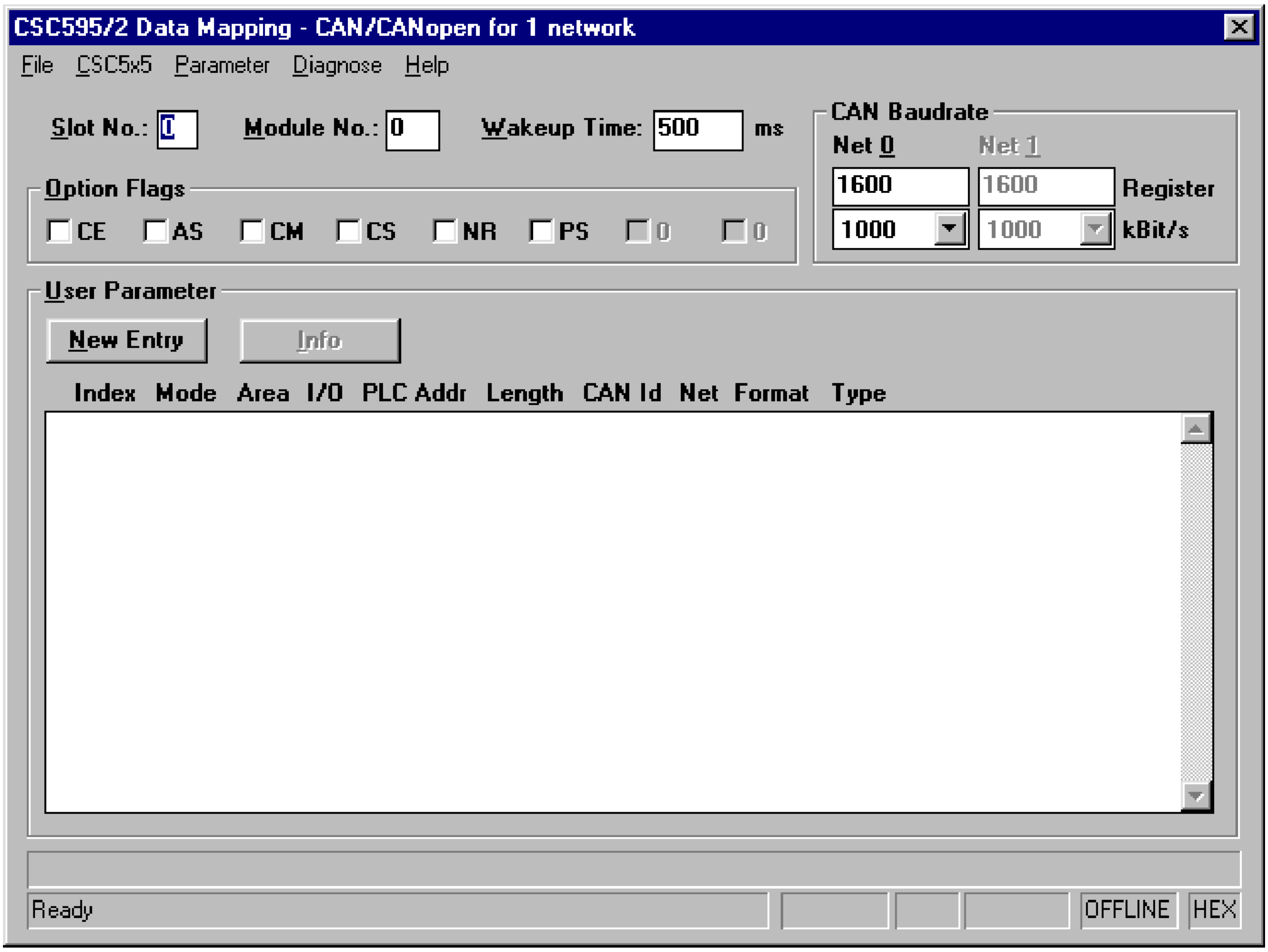


Fig. 2.5: Data-Mapping dialog box

2.5.1 Selecting the Number Format

1. In order to set the number format, decimal or hexadecimal, you have to select ***Hex / Dec*** in the ***Parameter*** menu.

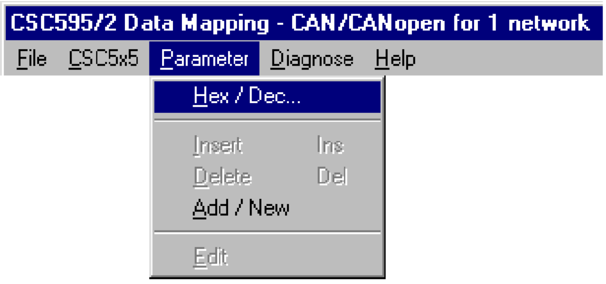


Fig. 2.6: Select *Hex/Dec*

2. You have to select the according number format in the dialog box which will open then. The selected number format will be shown in the status line.

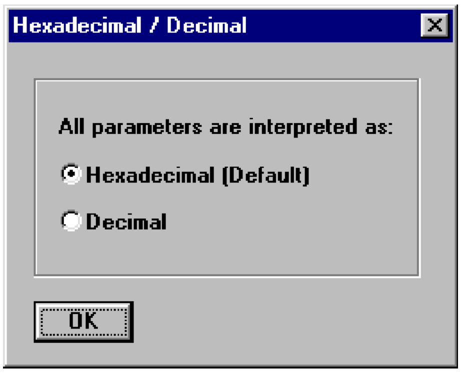


Fig. 2.7: Selecting the data format

When leaving the dialog box all parameters entered in the dialog box ***CSC5x5/Data Mapping*** will be automatically adapted in accordance with the data format selected. Below on the right in the status line the selected data format HEX or DEC will be shown.
Exceptions are the ***Wakeup Time***, which is only shown decimally, and the ***CAN-Baudrate***, whose assigned register contents are always shown hexadecimally.

The data format you have selected is also stored into the `INI` file and therefore does not have to be set again each time.

2.5.2 Parameter Setting

2.5.2.1 Slot Number (only CSC595/2)

The slot number can be entered either decimally or hexadecimally, depending on the data format set. The slot number corresponds to the physical slot position in the S5 and is required by the local firmware for calculating the simulated PLC addresses.

Permissible value range: `DEC 0...30 ($00...$1E)`

2.5.2.2 Module Number

The ***Module Number*** can be entered either decimally or hexadecimally, depending on the data format set.

Parameter	Value range	Descriptions
Module Number	0 ($00)	The module No. corresponds to the number set at the coding switch.
	1...255 ($01...$FF)	The module No. corresponds to the number defined here.

Table 2.1: Function of parameter *Module Number*

2.5.2.3 Wakeup Time

By means of this parameter a delay in milliseconds is defined which determines the period the module waits after a RESET or Power-On before transmitting data to the CAN bus again.

When the default setting of the parameters is active, the Wakeup time is 500 ms.

The ***Wakeup Time*** can <u>only</u> be entered decimally.

Parameter	Value range [ms]	Descriptions
Wakeup Time	32767	Delay in milliseconds. Value range decimal: 0 ms ... 32767 ms

Table 2.2: Function of parameter *Wakeup Time*

2.5.2.4 CAN-Baudrate

The ***CAN-Baudrate*** can be set in two ways:

1. Selecting the CAN-bit rate in the selection box

1. Click the selection box with the left mouse button. A list of available standard CAN-bit rates will appear.

2. Select CAN-bit rate. The bit-timing register according to the CAN-bit rate will automatically be shown in the editing field.

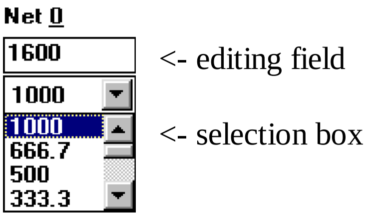


Fig. 2.8: Selecting the CAN-bit rate

2. Direct entry of the bit-timing-register value

1. Mark the bit-timing-register value in the editing field.

2. Enter new four-digit value. The bit-timing-register value is entered in format BTR0 / BTR1 (CAN-controller register name).

   The following table shows the register contents for various bit rates. The typical attainable line lengths are based on experimental values from experience. The minimum attainable lengths have been determined from specifications of component manufacturers for worst case delays.

Bit rate [kbit/s]	Value for editing field= bit-timing register of C167C [HEX]	typical values of attainable line length l_{max} [m]	minimum attainable line length l_{min} [m]
1000	1600	37	20
666	1B00	80	65
500	2F00	130	110
333	1B01	180	160
250	2F01	270	250
166	2F02	420	400
125	1C04	570	550
100	2F04	710	700
66	1B09	1000	980
50	2F09	1400	1400
33	2F0E	2000	2000
20	2F18	3600	3600
12.5	2F27	5400	5400
10	2F31	7300	7300
6	7F7F	10000	10000

The specifications in the table are based on limit values of the bit timing of the CAN protocol, the delays of the local esd-CAN interface and the delays of the cable. The cable delay is assumed to be ca. 5.5 ns/m. Further influences, such as unsuitable line terminations, the resistivity, the cable geometry or external disturbances during transmission have not been considered!

Table 2.2: Assignment of bit rate to the registers of the controller

3. Strike [Enter] key. When the bit rate is a standard-CAN-bit rate supported by the CSC5x5, this will be shown in the selection box. If there is no standard-CAN-bit rate matching the value specified, the entry ‘Special’ will be shown in the selection box.

If the hardware to be configured is a **CSC515** with two CAN networks, you have to specify the CAN-bit rate for network 0 as well as for network 1.

2.5.2.5 Option Flags

You CAN activate or deactivate the ***Option Flags*** either by clicking the individual flags by mouse or by striking the [Enter] key.

2.5.2.5.1 Meaning of Flags

Option flag	Explanations
CE *AS*	These two flags are evaluated together regarding the start features. For activating the communication window only the status of the CE flag is significant. *CE* *AS* Function -- 0 x Module starts transmissions automatically after Power-On or RESET. 1 1 Ditto & communication window is activated. 1 0 Module must be started by PLC. And communication window is activated.
CM	CAN-master functionality When this flag is set to ‚1', the module CAN automatically start the general CAN transfer by a start telegram to all CAN-slave modules.
CS	CAN-slave functionality When this flag is set to ‚1', the module does not start the general CAN transfer before it has been enabled by the PLC and has received a start telegram from the CAN bus.
NR	In default (*NR*=0), remote frames are transmitted on the input identifiers apart from the output data when the module starts. This CAN be avoided by *NR*=1.
PS	PLC start: The CAN transfer only starts after a special start command (incl. bit rate) by the PLC. This option is only active, when the communication window is also active (*CE* set).

Table 2.3: Meaning of option flags

2.5.2.5.2 Special Features of Some Option Flags

***CE*-Flag**
When the *CE* flag is:

activated — the entry of the communication window (PAE / PAA) will be shown in the user-parameter list.

deactivated — the entry of the communication window (PAE / PAA) will be deleted from the user-parameter list, and if the PS flag is activated, it will also be deactivated.

***CM*-Flag / *CS*-Flag**
Only the following status are permissible:

CM-Flag	*CS*-Flag	Function
0	0	None
0	1	CANopen-slave functionality
1	0	CANopen-master functionality
1	1	Not permissible

***PS*-Flag**
When the *PS*-flag is:

activated — the *CE*-flag will also be activated and the entry of the communication window (PAE / PAA) will be shown in the user-parameter list.

deactivated — the *CE*-flag will remain active.

2.5.3 User-Parameter-Entries

2.5.3.1 Adding a User-Parameter Entry

1. A user-parameter entry is added by selecting ***Add / New*** from the menu ***Parameter*** or by clicking the ***New Entry*** button. The ***Parameter*** window will appear (see page 19). When the list already contains an end entry (Mode = 'E'), or when the maximum number of entries has been reached, no further user parameter CAN be entered.

2. Edit parameter in '***Add-Parameter'*** window.

3. By striking the [Enter] key or by clicking the ***OK*** button in the ***Parameter*** window the new entry is added to the user-parameter list and the list is set up again.

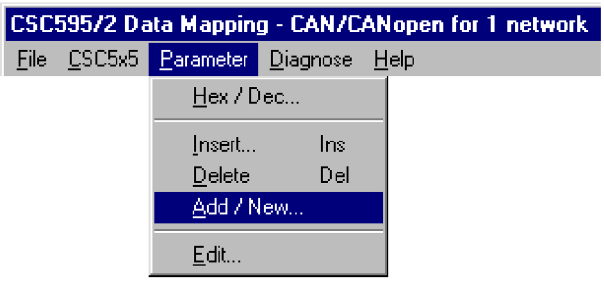


Fig. 2.9: Calling *Add/New*

2.5.3.2 Inserting a User-Parameter Entry

1. Mark the user-parameter entry **before** which you want to insert the new entry.

2. Select ***Insert*** from the menu ***Parameter*** or strike the [Ins] key. The ***Parameter*** window will appear (see page 19).

3. Edit parameter in **'*Insert-Parameter*'** window.

4. By striking the [Enter] key or by clicking the ***OK*** button in the ***Parameter*** window the new entry is inserted into the list and the list is set up again.

2.5.3.3 Editing a User-Parameter Entry

1. Mark the user-parameter entry you want to edit.

2. Select ***Edit*** from the ***Parameter*** menu, strike [Enter] key or double-click the entry with left mouse button. The ***Parameter*** window will appear.

3. Edit parameter in **'*Edit-Parameter*'** window (see page 19).

4. By striking the [Enter] key or by clicking the ***OK*** button in the ***Parameter*** window the entry with the edited parameters is shown in the list.

2.5.3.4 Deleting a User-Parameter Entry

1. Mark user-parameter entry.

2. Select ***Delete*** from ***Parameter*** menu or strike [Del] key. The entry will be deleted and the list will be set up again.

The user-parameter entry you delete is **irretrievably** deleted.

2.5.3.5 Deleting All User-Parameter Entries

1. Select ***Clear*** in ***CSC5x5*** menu.

2. You will be asked to store your changes into a file.

3. All user-parameter entries will be deleted.

2.5.3.6 Requesting Information About Wrong User Parameters

If an error occurs caused on the editing of a user-parameter entry (such as assigning a CAN identifier twice), this will be shown by the larger sign '>'. You CAN get information about the error by following the steps below.

1. Mark the user-parameter entry for which you want to have information about an error.

2. Click ***Info*** button. A window with the following error message will appear:

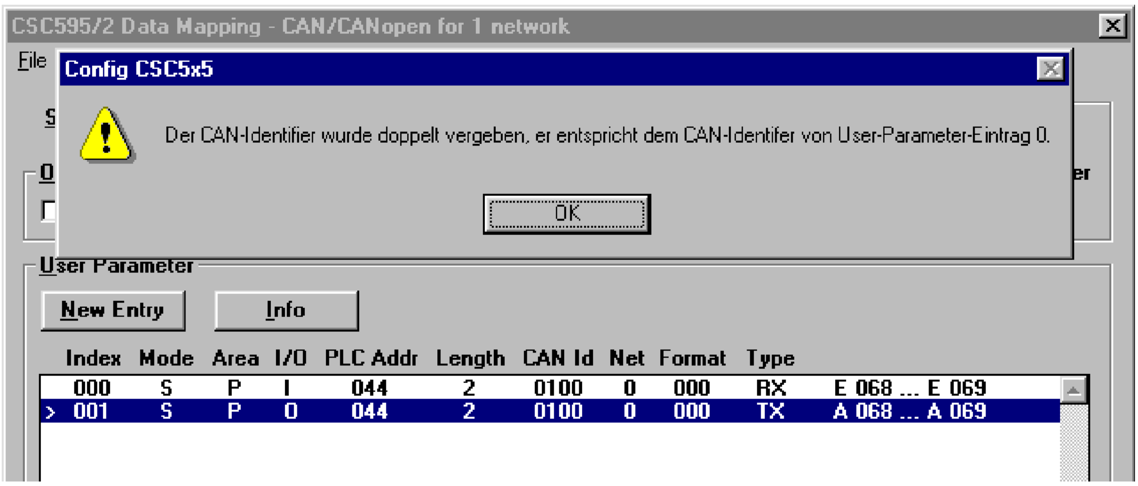


Fig. 2.10: Example: Error message because of an identifier assigned twice

3. Read error message and click ***OK*** button or strike [Enter] key.

4. Remove error by editing the user-parameter entry.

2.6 The User-Parameter Window

Via the ***User-Parameter*** window the user-parameter entries CAN be edited.

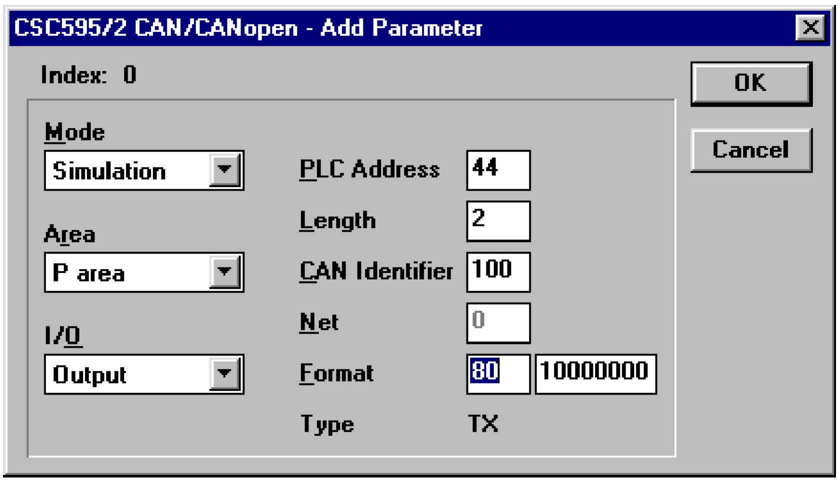


Fig. 2.11: *User-Parameter* window

Mode

In the selection box ***Mode*** you CAN choose between the parameters ***Simulation*** (standard operation), ***Monitoring*** (transmit outputs to PLC also to CAN) and ***End*** (end of table, no further parameters). When you select ***End***, all selection boxes and editing fields apart from the ***Mode*** selection box will be disabled. You cannot handle these fields anymore.

Area

Selecting the PLC-memory area which has the address selected via ***PLC Address***.
The CSC595/2 only has the P-area, the CSC515 only the P- and Q-area and the IPC-flags.

I/O

In the selection box ***I/O*** the data direction of the PLC port selected via the ***PLC Address*** is determined (as seen from the PLC). You CAN choose between the parameters ***Input*** and ***Output***.

PLC Address

Absolute PLC-peripheral address.
The parameter ***PLC Address*** CAN either be entered decimally or hexadecimally, depending on the data format.

Value range: `0...31    ($00...$1F)`
`64...127  ($40...$7F)`

Attention: When you use the ‚virtual CAN buffer' and/or the 'Rx-FIFO', the first four addresses will be used by the communication window! Do not use them otherwise! The communication window is activated by means of the CE-flag. The memory area used by the window is shown in the Data-Mapping window (see example on page 27).
The 'virtual CAN buffer' and the 'Rx-FIFO' will be described in detail in special chapters.

Length

Number of data bytes of a frame transmitted on the CAN bus.
The parameter ***Length*** CAN either be entered decimally or hexadecimally, depending on the data format set.

Value range: `0...8`

CAN Identifier

CAN identifier which is to be linked to the PLC address.
When you have selected a PLC-input port for the address, ie. *I/O=**Input***, the ***CAN Identifier*** for the CSC5x5 module is a receive identifier (RxId).
When you have selected a PLC-output port for the address, ie. *I/O=**Output***, the ***CAN Identifier*** for the CSC5x5 module is a transmit identifier (TxId).

The parameter ***CAN Identifier*** CAN either be entered decimally or hexadecimally, depending on the data format set.

Value range:	0...2045	($0...$7FD)
	2046	($7FE)
	2047	($7FF)

The identifier 2046 ($7FE) is used for the special case of Rx-FIFO reading (see according chapter).
The identifier 2047 ($7FF) CAN be used as a dummy identifier in programming, because it does not execute any actions on the CAN bus.

Net

The parameter ***Net*** CAN only be changed on a **CSC515** with two CAN networks. In this case you have to enter a 0 for CAN network 1 and a 1 for CAN network 2. Otherwise the value is always 0.

Format

Setting the byte assignment of CAN and PLC data regarding Intel and Motorola format.
The parameter ***Format*** CAN either be entered decimally or hexadecimally, depending on the data format set. In addition you CAN also enter it in binary form.
The parameter ***Format*** will be described in detail at the end of this chapter.

Type

The parameter ***Type*** depends on the setting of other parameters and is only used for information.

Further Explanations About the Parameters

Parameter *Format*: Transforming the Data Format

The parameter ***Format*** is used to transform the data format.
Background: Messages which are longer than 1 byte are normally transmitted in Intel format (low byte first) on the CANbus, while the Siemens PLC is operating in Motorola format (high byte first). These formats CAN be transformed into each other by the format byte.

Starting with bit 7 of the format byte you CAN decide whether the following byte is also to be transformed (swapped) or not. When you enter a ‚1' for a byte, the following bytes will be swapped including the following ‚0' defined. This CAN be shown best by means of an example.

Example:
A CAN telegram has a date in Intel format in the first word, then two bytes which are not to be swapped and a longword in Intel format again in the last four bytes.
Binary you have the following picture for the format byte:

Bit No.	7	6	5	4	3	2	1	0
Bit of *Format*	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**
Action	begin swap	end swap	remain	remain	begin swap	swap	swap	end swap

Data bytes	1	2	3	4	5	6	7	8
CAN frame	2 bytes Intel format		byte 3	byte 4	4 bytes Intel format			
PLC data	2 bytes Motorola format		byte 3	byte 4	4 bytes Motorola format			

From this the format byte results in $8E. When you want to swap all eight bytes, you would have to define the value $FE for the format byte.

The LSB is generally insignificant, because the telegram and therefore the formatting have been completed. It should always be set to 0.

When the parameter ***Format*** is not set, the default setting will be used. In default setting all bits are set to ‚0', ie. that no data formats are swapped.

2.7 Loading the Configuration from the CSC5x5

When you want to load the configuration which is currently stored in the CSC5x5, you have to select ***Load from...*** from the ***CSC5x5*** menu. All data stored in the CSC5x5 will be read and will appear in the configuration window.

The configuration data loaded via ***Load from...*** overwrite the data which has already been set in the configuration window. Therefore you will be asked to store your changes into a CFG-file.

2.8 Transmitting the Configuration to the CSC5x5

When all parameters are correct, *and only then*, you CAN transmit the configuration also to the CSC5x5.
For this you have to select ***Save to...*** from the ***CSC5x5*** menu. If you have configured the CSC5x5 offline so far, a connection to the CSC5x5 will be established first. When the connection has successfully been established, all parameters will be transmitted to the CSC5x5. At the end of the download the Save-Configuration (CS) and Reset (RS) commands will automatically be executed and the configuration will be stored in the CSC5x5 so that the configuration parameters will be available after the next Power-On of the PLC.

2.9 Store the Configuration File in PC

1. Select ***Save*** from the ***File*** menu. A window will open from which you CAN select a file.

2. Select CFG-file (change path if necessary).

3. Click ***OK*** button or strike [Enter] key.

2.10 Load Configuration File from PC

1. Select ***Open*** from the ***File*** menu. A window will open from which you CAN select a file.

2. Select CFG-file (change path if necessary).

3. Click ***OK*** button or strike [Enter] key. All data stored in the file will be shown in the configuration window.

2.11 Printing a Configuration File

1. Select ***Print*** from the ***File*** menu. A window will open from which you CAN select a file.

2. Select CFG-file (change path if necessary).

3. Click ***OK*** button or strike [Enter] key. The Windows-standard window for setting the printer options will open.

4. Click ***OK*** button or strike [Enter] key. The print-out will be created.

2.12 Example

In this chapter we will use an example to show how the CSC5x5 is configured. The following frame conditions and requirements will be assumed:

In a CANopen network a DO module (digital output) is to be driven. The frame conditions are as follows:

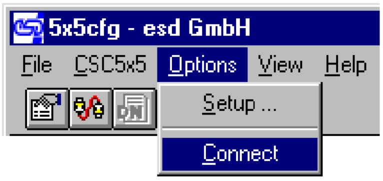
Bit rate of the CANopen network . . . . 125 kbit/s
Slot number of the CSC595/2 in the PLC 0
Module No. 5
CAN-ID $205
Autostart after . . . . 1 s

The CSC595/2 module is in slot 0. The bit rate does not correspond to the default setting and has to be changed therefore. The communication window is to be activated for future applications.
The device is to be switched on completely, ie. the module is to start the CAN network automatically.

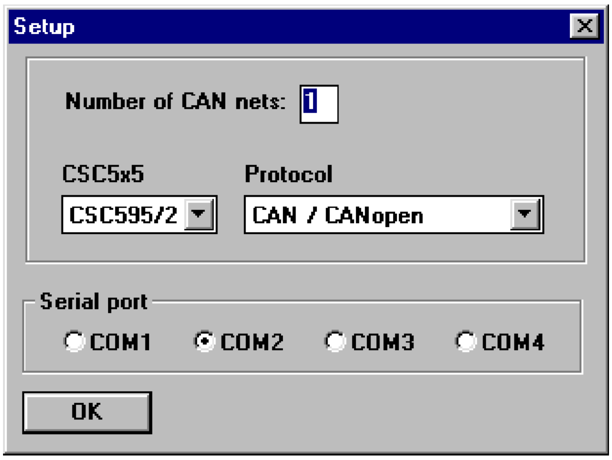
The module is to be configured in offline mode.

1. Call `5x5cfg.exe` program.

2. Select ***Setup*** in ***Options*** menu:

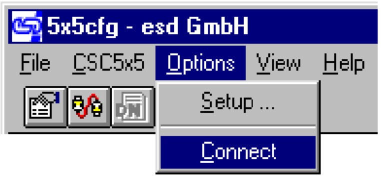


3. Carry out settings in the Setup dialog box:

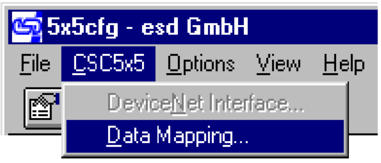


Configuration Program 5x5cfg

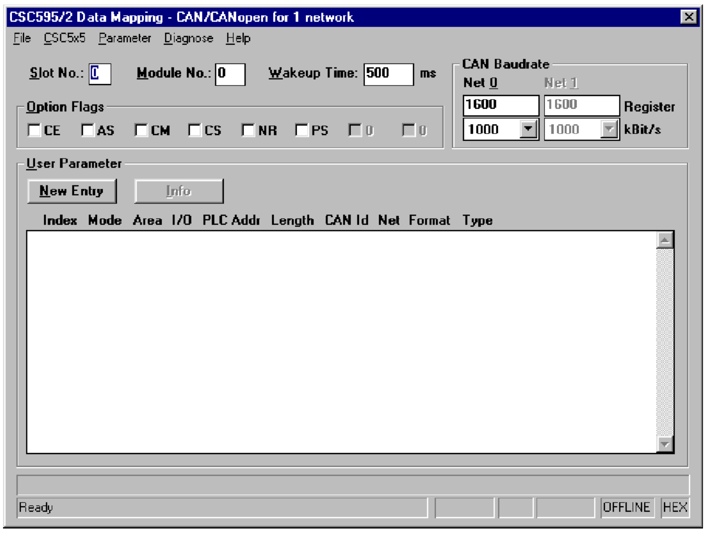
4. In order to configure the CSC5x5 in online operation, you would have to select ***Connect*** in the ***Options*** menu. As mentioned above, however, the CSC5x5 is to be configured in offline operation:



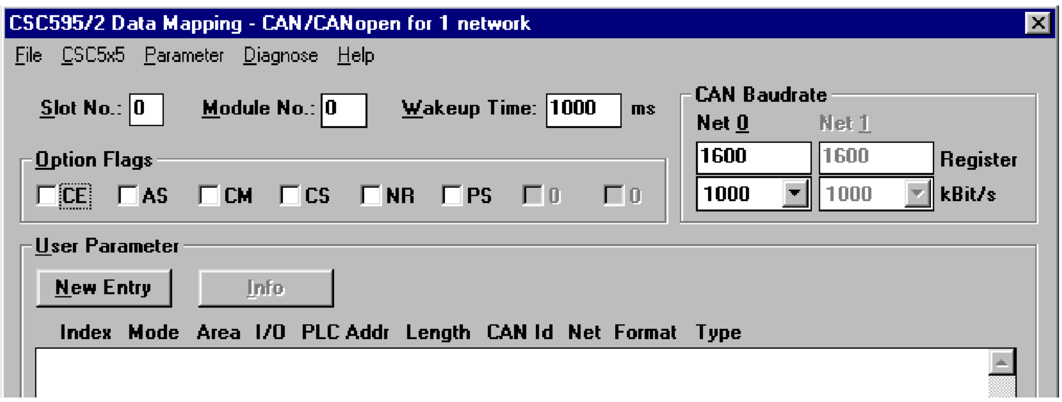
5. Select ***Data Mapping*** in the ***CSC5x5*** menu:



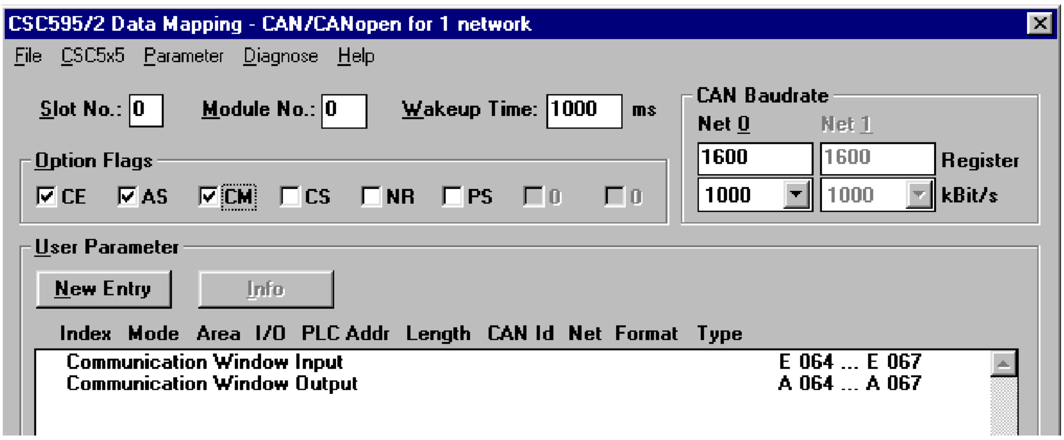
The following window opens:



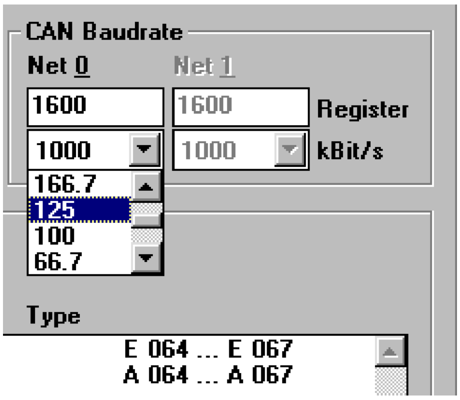
6. Set parameters ***Slot No***, ***Module No*** and ***Wakeup Time*** as shown below:

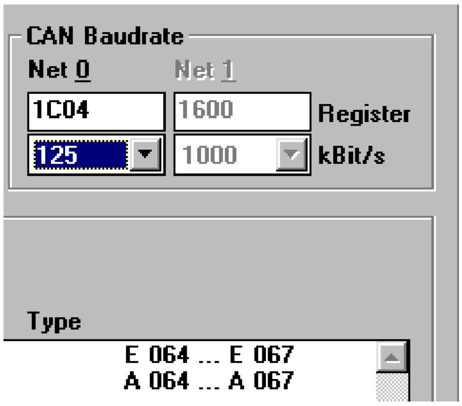


7. Activate the option flags *CE*, *AS* and *CM*:

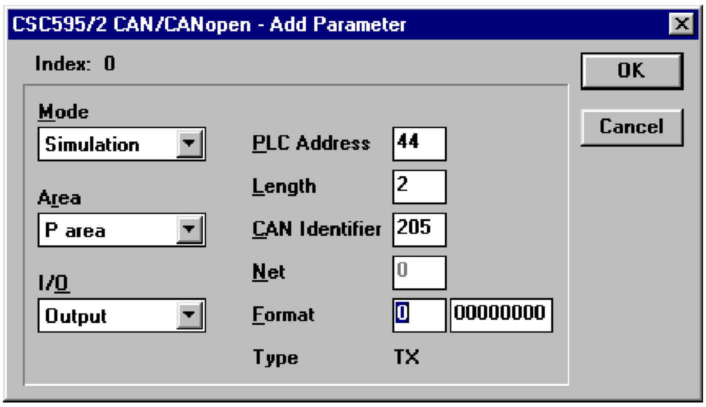


8. Click the ***CAN-Baudrate***-selection box with the left mouse button. A list field with four entries will open. Select 125 kbit/s. The bit-timing-register value in the editing field above the selection box will automatically be set to $1C04 when releasing the mouse button.



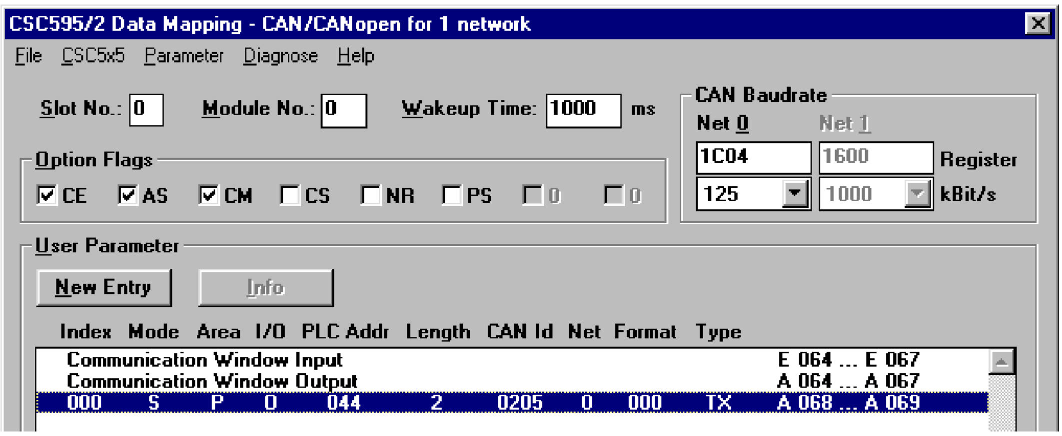


9. Select ***Add / New*** in the ***Parameter*** menu or click ***New Entry***. The ***Add-Parameter***-dialog box will open. Set the parameters as shown and close the dialog with ***OK***:

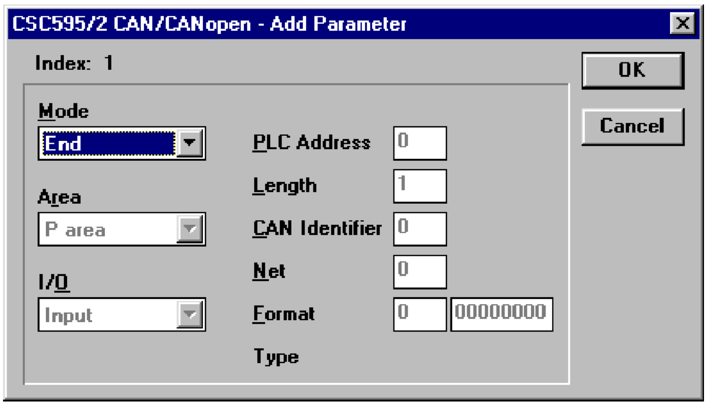


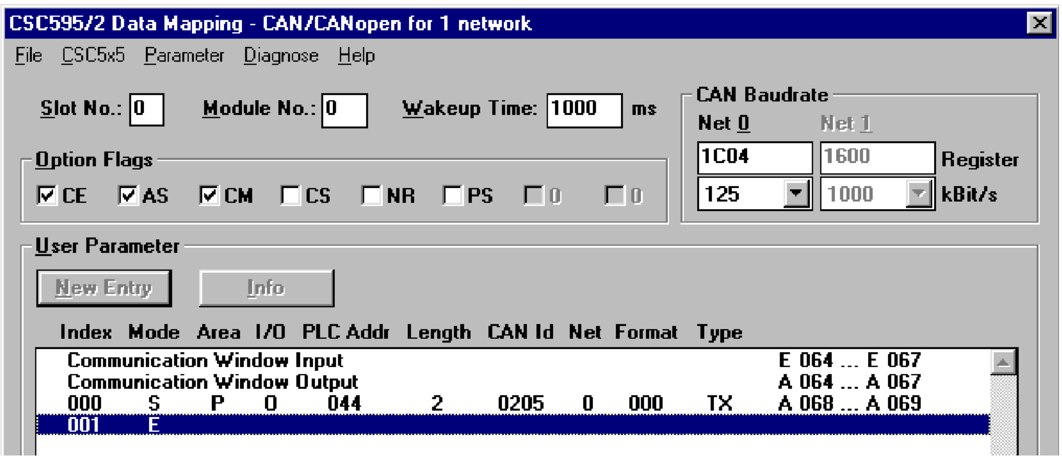
Configuration Program 5x5cfg

The parameter entry will be shown in the user-parameter list as shown below:

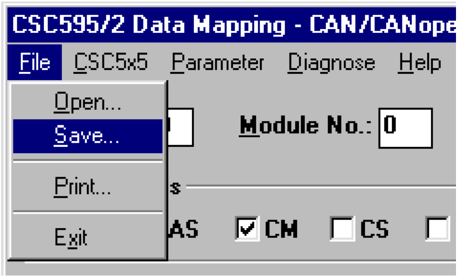


10. Insert end mark:

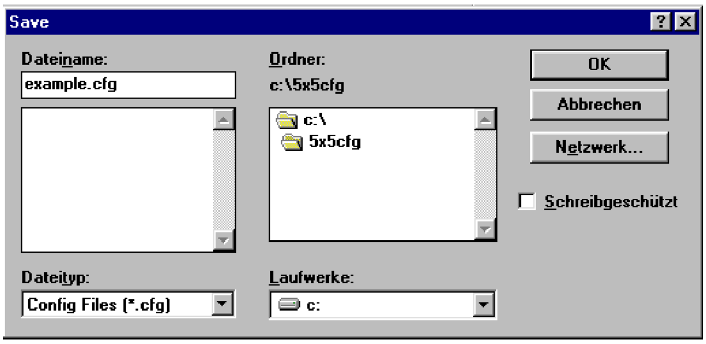




11. Save configuration in a file. Select ***Save*** in the ***File*** menu.

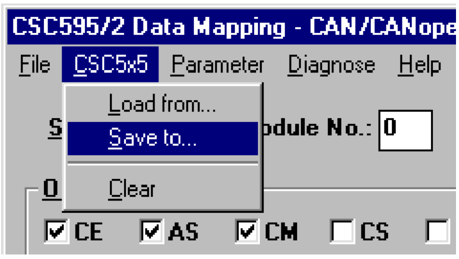


The following dialog box will open:

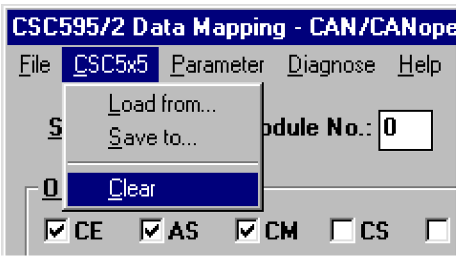


Enter file name 'example.cfg' and close the dialog box by ***OK***. The path specification and the file name will be shown in the status line.

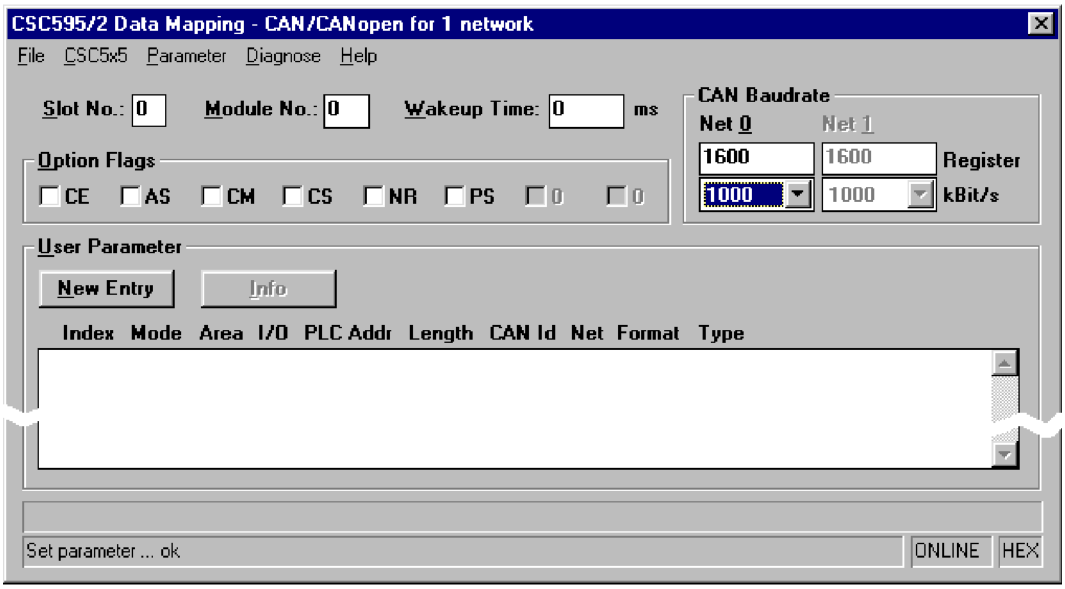
12. Transmit configuration to CSC5x5. Select ***Save to...*** in the ***CSC5x5*** menu:



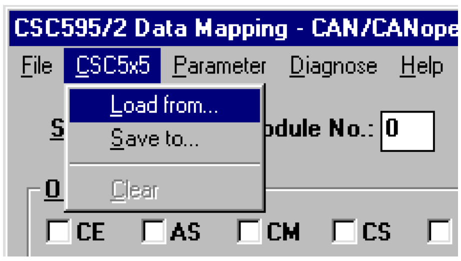
13. In order to control whether the parameters have really been stored in the CSC5x5, you have to delete all parameters first. To delete the user parameters you have to select ***Clear*** in the ***CSC5x5*** menu:



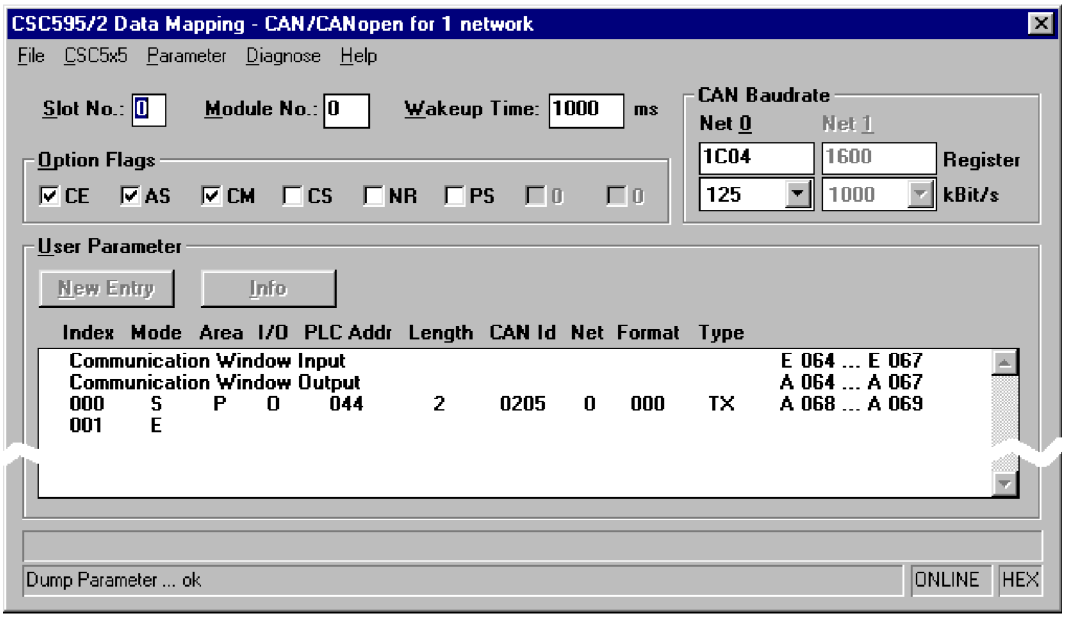
All other parameters have to be deleted manually:



14. Then you CAN load the configuration stored in the CSC5x5 by selecting ***Load from...*** in the ***CSC5x5*** menu.



Because this is an offline configuration, a connection to the CSC5x5 will be established first. This will only be successful, when you have selected the correct serial interface in the ***Setup***-dialog box at the beginning of this example. When the connection could be established, the message 'ONLINE' will be shown in the status line down below on the right.


CSC595/2 Data Mapping - CAN/CANopen for 1 network
File CSC5x5 Parameter Diagnose Help
Slot No.: 0
Module No.: 0
Wakeup Time: 1000 ms
CAN Baudrate
Net 0
Net 1
1C04
1600
Register
125
1000
kBit/s
Option Flags
CE AS CM CS NR PS 0 0
User Parameter
New Entry
Info
Index Mode Area I/O PLC Addr Length CAN Id Net Format Type
Communication Window Input E 064 ... E 067
Communication Window Output A 064 ... A 067
000 S P O 044 2 0205 0 000 TX A 068 ... A 069
001 E
Dump Parameter ... ok
ONLINE
HEX

3. Configuration via Terminal

> You should only directly configure via the commands of the terminal interface, if no PC is available. The configuration is much more comfortable via PC and the configuration program 5x5cfg.

3.1 Terminal Setup

The wiring of the terminal via the 9-pole DSUB connector is described in the hardware manual of the CSC595/2. If a PC is used instead a terminal, the ‘Hyper Terminal Function’ is recommended. It can be started e.g. under Windows NT by selecting the windows ‘Start’/’Programs’/ ’Accessories’/ ’Hyper Terminal’.

The terminal settings has to be as follows:

19200 bit/s
8 bit data
1 stop bit
no protocol (or Xon/Xoff, if ‘no protocol’ is not selectable)

3.2 Procedure

Please observe the following order when configuring the module:

Command		Notes
1	CD (CAN default)	selecting the default parameters always guarantees defined conditions to start with
2	SB (set bit rate)	if you desire another than 1 Mbit/s
3	SN (slot No.)	Number of the PLC slot selected (required for the internal address calculation)
4	FL (set flags)	see according chapter on page 40
5	UP (user parameters) : :	select PLC - CAN link, command has to be called once for each CAN identifier desired
6	MN (module No.)	is only required for CANopen
7	WU (wake-up timer)	if the start delay is to unequal 500 ms
8	CS (save config)	storing all settings into the I^2C-EEPROM
9	RS (reset module)	new parameters will only be taken over after re-booting
10	end of configuration	PLC now regards CAN-I/Os as PLC-peripheral devices

Procedure

3.3 Overview of Commands Implemented

The following table gives an overview of all commands which have been implemented so far:

Syntax	Function
`CD`	CAN default
`SB`	set bit rate
`DB`	display bit rate
`SN`	slot No.
`FL`	flags
`UP`	user parameters
`MN`	module number
`WU`	wake-up time
`CS`	save config
`RS`	reset module

Table 3.1: Configuration commands

If a '`?`' is entered after a command, an overview of parameters to be entered will be put out onto the screen (if parameters are required for this command).

3.4 Syntax

In the following descriptions some parts of the text have been highlighted. Generally they have the following meaning:

`font change`	File names, entries and outputs onto the screen and names of keys will be marked by another type of font.
8	Characterization of blanks in entries.
`variable`	Variable input lengths will be shown in italics.
`[ ]`	Optional entries will be given in square brackets.

Commands and parameters can either be entered in small or capital letters via the terminal. The same applies for hexadecimal numbers.

3.5 Command Descriptions

3.5.1 Resetting the Module to Default Parameters

CAN Default - CD

This command resets all module parameters to the default values.

The default parameters are only active after a following ‘Save Configuration’ command (CS) and a standard RESET (RS)!

No parameters are transmitted.

Command call:

CD

3.5.2 Set and Display Bit Rate

Set Bit Rate - `SB` and Display Bit Rate - `DB`

By means of these parameters the bit rate of the CAN controller is set. The new bit rate is only activated when the parameter has been stored onto the local I²C-EEPROM via the command `SAVE_CONFIG` and a RESET has been triggered.

Command call:

`SB8` ***`bitindex`***

or:

`SB8` ***`regvalue`***

The following table shows the index and register contents assigned to the index for various bit rates. Alternatively the register contents (*`regvalue`*) can be directly specified. The typically attainable line lengths base on values from experience. The minimum attainable line lengths have been determined from the specifications of component manufacturers for the ‘worst case’ delay times.

bitindex	Bit rate [kbit/s]	*regvalue* Bit-timing register (C167C)	typical values of attainable line lengths l_{max} [m]	minimum attainable line length l_{min} [m]
0	1000	1600	37	20
1	666	1B00	80	65
2	500	2F00	130	110
3	333	1B01	180	160
4	250	2F01	270	250
5	166	2F02	420	400
6	125	1C04	570	550
7	100	2F04	710	700
8	66	1B09	1000	980
9	50	2F09	1400	1400
A	33	2F0E	2000	2000
B	20	2F18	3600	3600
C	12.5	2F27	5400	5400
D	10	2F31	7300	7300
E	6	7F7F	10000	10000

The specifications in the table base on the limit values of the bit timing of the CAN protocol, the delays of the local esd-CAN interface and the delays of the cable. The delay of the cable is assumed at about 5.5 ns/m. Further influences, such as caused by unsuitable line terminations, the resistivity, the cable geometry or external disturbances during transmission have not been considered!

Table 3.2: Assigning the bit rate to the controller registers

Example:

Setting the bit rate to 500 kbit/s.

```
SB82     >RETURN<
```

or alternatively

```
SB82F00 >RETURN<
```

SB DB **Set Bit Rate / Display Bit Rate**

When requesting the bit rate the previously programmed index or register contents is returned:

Command call:

DB

Output:

bitindex

or:

regvalue

Example:
Requesting the bit rate of a CSC595/2.

Calling the command

`DB >RETURN<`

results at a current bit rate of 500 kbit/s e.g. in the output of

`’2’`

when the bit rate has previously been transmitted as *bitindex*, or in the output of

`’2F00’`

when the bit rate has previously been transmitted as register contents *regvalue*.

3.5.3 Setting the PLC-Slot Number of the Module

Set Slot No. - SN

Via this command the number of the slot has to be entered into which the CSC595/2 module has been plugged on the PLC. The firmware needs this specification for determining the data position.

Command call:

SN8 *slot_no*

slot_no = $00...$1E

If the command is called without parameter specification, the currently set slot number will be returned.

FL

Set Flags

3.5.4 Setting General Flags

Set Flags - FL

By means of this command various flags for configuring the module can be set. In default setting all flags are set to ‘0'.

Command call:

```
FL8 CE8 AS8 CM8 CS8 NR8 PS
```

<table>
<tr><th>Parameter flag</th><th>Descriptions</th></tr>
<tr><td>CE</td><td rowspan="2">These two flags are evaluated together:

CE AS Function
----
0 x module starts independently transmitting after power on or RESET
1 1 dto.
1 0 module must necessarily be started by the PLC</td></tr>
<tr><td>AS</td></tr>
<tr><td>CM</td><td>CAN-master functionality
If this flag is set to ‘1', the module can independently start the general CAN transfer by a start telegram to all CAN-slave modules.</td></tr>
<tr><td>CS</td><td>CAN-slave functionality
If this flag is set to ‘1', the module only starts the general CAN transfer after it has been enabled by the PLC and has received a start telegram by the CAN bus.</td></tr>
<tr><td>NR</td><td>In default (NR=0) remote frames are transmitted on the input identifiers in addition to the output data when the module starts. This can be avoided by NR=1.</td></tr>
<tr><td>PS</td><td>PLC start: The CAN transfer only starts after a special start command (including bit rate) by the PLC. This option is only active when the communication window is also activated (CE set).</td></tr>
</table>

Table 3.3: Parameters of command ‘Set Flags’

3.5.5 Set and Display Parameters

User Parameters - UP

The command 'User Parameters' replaces the previously used commands 'Set Parameters' and 'Display Parameters'. If the command 'User Parameters' is called with specified parameters, it corresponds to the previous command 'Set Parameters'. If it is called without parameter specifications, the user parameters will be shown.
The commands 'Set Parameters' and 'Display Parameters' can still be used. They will not be described here, however, because only the command 'User Parameters' should be used in future.

Via the command 'User Parameters' a table of entries is generated which assigns PLC addresses and CAN identifiers. In each line of this table a CAN identifier and a number of data bytes is assigned to a PLC address.

Command call for setting the parameters:

```
UP8 index8 mode[8] P[8] io8 spsaddr8 length8 idf8 [net8 form]
```

The end of the table has to be determined by the entry 'E' for the variable '*mode*'. All entries following this entry will be ignored. The penultimate parameter '*net*' must always be set to '0', because this board only has one CAN network.

When entering the command 'User Parameters' without parameter specification the programmed parameters will be shown. The parameters are shown like when being entered. In addition the text 'rx' or 'tx' is put out at the end of each line to show the identifier type. All lines of the assignment table will be put out onto the screen (also the parameters which only have to entered optionally).

Command call:

```
UP
```

Output (per line):

```
'index8 mode8 area8 io8 spsaddr8 length8 idf8 net8 form8 typ'
```

After the call the following 32 entries marked by '*index*' are shown.
The description of the parameters and their value ranges can be taken from the following pages.

UP **User Parameters**

Parameter	Value range	Explanations
index	$00...$7E	Sequential numerical designations of the parameter groups in the assignment table.
mode	S, M, E	Selection of operating mode by entering a letter: S - simulation (standard operating mode) M - monitoring (transmit outputs to PLC also to CAN) E - end of table, no further parameters
P	P	Selecting the PLC-memory area in which the address selected via *spsaddr* is. For the CSC595/2 only the P-area can be selected, therefore you always have to enter a 'P' here.
io	I, O	Determination of the data direction of the PLC ports selected via *spsaddr* (as seen from the PLC view): I - input O - output
spsaddr	$00...$1F, $40...$7F	Absolute PLC-peripheral address (hexadecimal) **Attention:** If the 'virtual CAN buffer' and/or the 'Rx-FIFO' are used, these first four addresses will be used by the communication window! They are then not be used otherwise!
length	1...8	Number of bytes transmitted on the CAN bus.
idf	$0...$7FD, $7FE, $7FF	CAN-identifier which is to be linked to this address. When you have chosen a PLC-input port, ie. *io*=I for the address, *idf* is a receive identifier (RxId) for the CSC595/2. When you have chosen a PLC-output port, ie. *io*=O for this address, *idf* is a transmit identifier (TxId) for the CSC595/2. The identifier $7FE is used for the special case of reading the Rx-FIFO (see page 67). The identifier $7FF can be used as a 'dummy identifier' for programming, because it does not execute any actions on the CAN bus.
typ	rx, tx	This parameter is only shown at parameter acknowledge. rx - CAN-receive identifier (RxId) tx - CAN-transmit identifier (TxId)
net	0	This parameter has always to be set to '0', because the CSC595/2 has only one CAN channel.
form	$00...$FE	Setting the byte assignment between CAN and PLC data regarding Intel and Motorola formats.

Table 3.4: Parameters of command 'User Parameters'

Further Explanations on the Parameters

Parameter *form* : Realising the data format

Parameter '*form*' defined via the UP-command, is used to realize the data format.
Background: Messages which are longer then 1 byte are normally transmitted in Intel format (low byte first) in the CAN network, while the Siemens PLC operates in Motorola format (high byte first). These formats each be converted into the other by means of the format byte.

Starting with bit 7 of the format byte, you can decide whether the following byte is to be converted, ie. Swapped, or not. If a '1' is entered for a byte, the following bytes are swapped until and including the next '0' is defined. This operation can be explained best by means of an example.

Example:
A CAN telegram has a date in Intel format in the first word, then 2 bytes which are not to be swapped and a longword in Intel format again in the last four bytes.
The binary representation for the format byte would be the following:

Bit No.	7	6	5	4	3	2	1	0
Bit of *form*	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**
Action	begin swap	end swap	unchanged	unchanged	begin swap	swap	swap	end swap

Data bytes	1	2	3	4	5	6	7	8
CAN frame	2 byte Intel format		byte 3	byte 4	4 byte Intel format			
PLC data	2 byte Motorola format		byte 3	byte 4	4 byte Motorola format			

From this the format byte for $8E results. When for example all 8 bytes are to be swapped, the value $FE would have to be defined for the format byte.

The LSB has generally no significance, because the telegram and therefore the formatting have been concluded. It should always be set to '0'.
If parameter '*form*' is not set, the default setting will be used. In default setting all bits are set to '0', ie. no data formats will be swapped.

UP

User Parameters

Example for Command UP (Table with two Entries)

```
+----------------------  command syntax for ‘Set Parameters’
|   +------------------  index       (line number in table = 0)
|   | +----------------  mode        (operating mode = simulation)
|   | | +--------------  P           (memory area = P-area [always!])
|   | | | +------------  io          (data direction = input)
|   | | | | +----------  spsaddr     (PLC address = $10 (DEC 16))
|   | | | | |  +-------  length      (length = 4 data bytes)
|   | | | | |  | +-----  idf         (CAN identifier = $123)
|   | | | | |  | |   +-  net         (CAN network = 0 [always!])
|   | | | | |  | |   | +- form       (byte order = Intel format)
|   | | | | |  | |   | |
|   | | | | |  | |   | |
|   | | | | |  | |   | |
UP 808S8P8I840848123808 00  >RETURN<
UP8 18E                     >RETURN<
   | |
   | +----------------  mode         (E = end of valid entries)
   +------------------  index        (Line number in table = 1)
```

The parameters acknowledging with UP

will cause the following message on the monitor:
‘008 SPI8 408 048 01238 08 008 rx’

By means of this entry the CSC595/2 copies the data of CAN identifier $123 into the PLC-peripheral addresses 64 to at most 67 and makes them available to the PLC. The data can be read by the PLC by means of the commands ‘L EW 64’ and ‘L EW 66’, provided that the module has been placed into the first PLC slot.

3.5.6 Define Module No.

Define Module-No. - `MN`

By means of this command you can select whether the module No. set at the coding switch is to be used for this module, otherwise a module No. is programmed.

When the module is shipped and after a default RESET (e.g. via the command CD) the number set at the coding switch is valid.

Command call:

```
MN8 module-no
```

Parameter	Value range	Explanations
module-no	$00	The module No. corresponds to the number set at the coding switch.
	$01...$FF	The module No. corresponds to the number which is defined here.

Table 3.5: Function of parameter *module-no*

WU

Wake-up Time

3.5.7 Set Wake-up Time after Start

Set Wake-up Time - `WU`

By means of this parameter a delay in milliseconds is defined which determines the time a module waits after a RESET or power-on before transmitting data to the CAN bus.

When the default setting of the parameters is active, the wake-up time is 500 ms.

Command call:

```
WU8 time
```

Parameter	Value range	Explanations
time	$0...$7FFF	Delay in milliseconds. Value range decimally: 0 ms ... 32767 ms

Table 3.6: Function of parameter *time*

3.5.8 Save Configuration

Save Config - CS

The command SAVE_CONFIG saves all parameters including the bit-rate setting into the local I²C-EEPROM. By doing this, the parameters are retained after a RESET.

Command call:

```
CS
```

After successfully saving the configuration an ‘ok’-prompt appears on the screen. If it does not appear the CS-command has to be repeated until the ‘ok’-prompt appears.

RS	**Reset Module**

3.5.9 Reset Module

RESET Module No. - RS

By means of this command a RESET is triggered in the module and all local units are reset. The parameters which have been saved by ‘CS - Save Configuration’ are retained. No parameters are defined.

Command call:

```
RS
```

3.6 Example

A DO-module (Digital Output) is to be controlled in a CANopen network. The frame conditions are as follows:

Bit rate of the CANopen network . . . . 125 kbit/s
Slot number of the CSC595/2 in the PLC 0
Module No 5
CAN-ID $205
Autostart after . . . . 1 s

The CSC595/2 module is in slot 0. The bit rate does not correspond to the default setting and has to be changed therefore. The communication window is to be activated for future applications.
The device is to be switched on completely, ie. the module is to start automatically the CAN network.

Command		Comment
CD		activate original status
SB	6	set bit rate to 125 kbit/s
FL	1 1 1 0 0 0	set flags: communication window on, autostart, master
SN	0	slot 0
UP	0 SPO 44 2 205	identifier results in CANopen at $205 with module No. 5
UP	1 E	end of user-parameter entries
WU	3E8	autostart after 1 s
CS		save configuration
RS		after Reset new parameters become active

4. Communication via the Virtual CAN Buffer

4.1 Internal Virtual CAN Buffers

In addition to the CAN buffers which can be defined via the command UP (User Parameters), the module has 32 more internal CAN buffers. Data and commands are read and written into this memory area by means of the first four bytes of the PLC slot occupied.
These four bytes will be called ‘communication window’ in the following.

By means of the communication window and the virtual buffer some configuration commands can be transmitted to the CSC595/2 by the PLC (instead of by the RS-232 interface). Furthermore 2 bytes of data can be transmitted on a CAN-Tx identifier and 2 bytes can be rad on an Rx identifier per PLC cycle. By this the access possibilities on the CAN bus are multiplied, considering the much restricted PLC peripheral area.

The disadvantage of this means of access is that fewer data can be transmitted between PLC and CAN bus. **Therefore these additional CAN buffers are only to be used for rare configuration and service tasks (such as during the initialisation of a network) and for transmission of data which is regardless of time (such as of analog inputs and outputs).**

The buffers are only assigned to any CAN identifier during operation, either as Rx or Tx buffers, temporarily (for a telegram) or statically. After a Reset generally no identifier is assigned, all assignments made during operation can be cancelled again.

You cannot assign the CAN identifiers which already have been linked to PLC addresses via the UP-command.

The internal CAN buffer is divided into two blocks (pages) of 256 bytes: A control block and a data block in which 8 bytes are reserved for each CAN-ID for the data to be transmitted or to be received.

Communication via the Virtual CAN Buffer

These blocks can be accesses word by word by means of the communication window.

Structure of the control buffer (control page):

		$0		$2		$4		$6	
Buffer 0	$000	length	buff_cmmd	ID.H CAN ID.L		reserved	reserved	reserved	reserved
Buffer 1	$008								
Buffer 2	$010								
:	:								
Buffer 31	$0F8								

Structure of the data buffer (data page):

		$0		$2		$4		$6	
Buffer 0	$100	data 0	data 1	data 2	data 3	data 4	data 5	data 6	data 7
Buffer 1	$108								
Buffer 2	$110								
:	:								
Buffer 31	$1F8								

By writing into the command bytes (together with the according length) the individual actions can be triggered (combinable):

Command `buff_cmmd`	Function
`$80`	Transmit CAN frame with according length. The CAN_ID and the data must have been entered before. The transmission is carried out also if the module is still in stop status.
`$40`	Transmit RTR frame on the CAN-ID. If the data are to be transmitted into this CAN buffer, it has to be declared as an Rx buffer for the according CAN-ID before (see communication window).
`$20`	Declare this CAN buffer as a Tx buffer. Doing this, the data are automatically transmitted following the reception of RTR frames (only if 82527 is equipped). This command can be combined with the transmission of the CAN frame (`buff_cmmd $80`).

All other bits in the `buff_cmmd`-byte are insignificant for the time being. The bits set are deleted again after the action has been executed.

Bit 1 can only be used by the user to control the reception of a CAN frame, because when a corresponding frame is received, the command byte is overwritten. All further bits are reserved for future applications.

Bit No. (7...0)	Set/ evaluate	Status	Meaning
7	set	0	command: do not transmit a CAN frame
		1	command: transmit CAN frame
6	set	0	command: do not transmit RTR frame on CAN-ID
		1	command: transmit RTR frame on CAN-ID
5	set	0	command: no Tx-buffer function for this buffer
		1	command: declare this buffer as Tx buffer (only with optional 82527 controller)
4	-	-	reserved for firmware, do not set or evaluate
3	-	-	
2	-	-	
1	set and evaluate	0	CAN frame received, bit has been reset by firmware
		1	since setting to ‘1' no CAN frames have been received on this ID
0	-	-	reserved for firmware, do not set or evaluate

Table 4.1: Function of the bits of the buffer-command byte `buff_cmmd`

4.2 Communication Window

The communication window is required to write data and commands into the virtual CAN buffer or to read then out of the buffer. The window is only available after it has been activated via the ‘flag command (FL) by means of the terminal.

The communication window is defined by the first four bytes of the first simulated slot (= slot address of the module).

Attention:	These bytes are not to be used for other PLC-CAN links (via command UP), if the communication window is used!

Note: If the commands SP and DP from older software versions are used instead of the current command UP, you have to consider the shift of the following PLC addresses by the four bytes of the communication window.
When using UP *no* addresses are shifted, because the addressing is absolute.

4.3 Assignment of Bytes of the Communication Window

The bytes are used as follows, starting from the basis address **‘n’** of the CSC595/2:

	PLC address			
	n	n+1	n+2	n+3
Output (AB)	command win_cmmd	sub-command sub_cmmd	parameter 0 or output data byte 0	parameter 1 or output data byte 1
Input (EB)	status bits	not used	input data byte 0	input data byte 1

Table 4.2: Assignment of the communication window

Input data is CAN-Rx data, ie. data from CAN to PLC.
Output data is CAN-Tx data, ie. data from PLC to CAN.

4.4 Command and Sub-Command in the Communication Window

At the moment the following commands have been implemented:

Command win_cmmd	Sub-command sub_cmmd	Description
00	x	**Null command**
01	00	Null command
	01	**Start - Module** The module starts transmitting CAN-data telegrams. When the CAN-master flag (CM) has been set, a start telegram is transmitted to all slave modules first (ID=0, length=2, data=$0100). When the data transfer starts data (RTR frames) is requested first on all Rx identifiers.
	02	**Stop - Module** The module stops transmitting CAN-data telegrams. When the CAN-master flag (CM) has been set, a stop telegram is transmitted to all slave modules (ID=0, length=2, data=$0000).
	03	**Save Configuration** This command is the same as the ‘CS’-command of the RS-232 interface.
	04	**Set baud rate** This command is the same as the ‘SB’-command of the RS-232 interface. The baud rate is defined as a parameter in bytes 2 and 3 of the slot.

<table>
<tr><th>Command
win_cmmd</th><th>Sub-
command
sub_cmmd</th><th colspan="2">Description</th></tr>
<tr><td rowspan="6">01</td><td>05</td><td>Installing an Id in the Rx FIFO
(Parameter = ID)</td><td rowspan="4">In contrast to the reception of Rx frames via win_cmmd 4, here the old data is not stored in the object buffer and not every time overwritten with the new data of this CAN ID, but stored in a FIFO. Up to 1024 CAN telegrams can be stored in the FIFO.</td></tr>
<tr><td>06</td><td>Removing an Id from the Rx FIFO (Parameter = ID)</td></tr>
<tr><td>07</td><td>Reading message out of Rx FIFO
After reading, the value '-3' (DEC) is returned as message 'Rx FIFO is empty' in the cell command.</td></tr>
<tr><td>08</td><td>Deleting the Rx FIFO
All data in the Rx FIFO are deleted.</td></tr>
<tr><td>09</td><td colspan="2">Reset Module (parameter = 0xC0DE)</td></tr>
<tr><td>10</td><td colspan="2">Start/Restart CAN with Baudrate x (parameter = baudrate)
The parameter values for the different bit rates are shown at page 37
The index or the register value may be used.</td></tr>
</table>

<table>
<tr><th>Command
win_cmmd</th><th>Sub-command
sub_cmmd</th><th>Description</th></tr>
<tr><td rowspan="3">02</td><td>xxxx.x001
(binary)</td><td>Set up Rx-buffer
By means of this command the CAN buffer (binary coded in xxxx.x) for the CAN-ID specified in the parameter word is set up as a Rx-buffer, ie. frames being received on this CAN-ID are then written into this buffer. The number of bytes which has been received is stored into the buffer (0...8). The CAN-ID from the parameter word is automatically deposited in the control block.

xxxx.x - buffer number: binary 0...11111 (DEC 0...31)</td></tr>
<tr><td>xxxx.x010
(binary)</td><td>Cancel Rx-buffer
By means of this command the CAN buffer(binary coded in xxxx.x) is cancelled again by the CAN-ID deposited in the control block. There is no need to write into the parameter word. Therefore it is possible to re-register a CAN buffer onto another CAN-ID in connection with the previous command.

xxxx.x - buffer number: binary 0...11111 (DEC 0...31)</td></tr>
<tr><td>xxxx.x011
(binary)</td><td>Re-register Rx-buffer
The CAN buffer specified is re-registered from the previous CAN-ID, deposited in the control block, to another CAN-ID, specified in the parameter word.

xxxx.x - buffer number: binary 0...11111 (DEC 0...31)</td></tr>
<tr><td>03</td><td>xxxx.xyyd
(binary)</td><td>Displaying a data word from the virtual CAN buffer
The data word of the PLC (EW n+2) shows the requested word of the data buffer. The display is automatically updated after the reception of a CAN telegram. The parameter word is insignificant, the bits in the sub-command have the following meaning:

xxxx.x - buffer number
binary 0...11111 (DEC 0...31)
yy - data offset 0, 2, 4 or 6
d - 0: control buffer
1: data buffer

<table><tr><th>yy</th><th>Offset</th></tr><tr><td>00
01
10
11</td><td>0
2
4
6</td></tr></table></td></tr>
</table>

Command `win_cmmd`	Sub-command `sub_cmmd`	Description
04	`xxxx.xyyd` (binary)	**Write into a data word or parameter of the CAN buffer** The data contained in the parameter word is written into the virtual CAN buffer. The meaning of the bits of the sub-command is the same as that of command 3. When the command byte in a CAN buffer (`xxxx.x000`) is being written into with this access, the meaning of the command byte (see the previous chapter 'Internal Virtual CAN Buffers') is triggered. `xxxx.x` - buffer number binary 0...11111 (DEC 0...31) `yy` - data offset 0, 2, 4 or 6 `d` - 0: control buffer 1: data buffer yy / Offset: 00 / 0 01 / 2 10 / 4 11 / 6
05	z	**Read user parameter** By means of this command the user parameter z is shown in the input data word (EW n+2). An eventually previously started display of a data word from the virtual CAN buffer then ends.
06	z	**Write user parameter** By means of this command the user parameter z is overwritten by the parameter word. The user parameter becomes only valid after saving the configuration and the following RESET.
07	*length*	**Transmit a CAN message via Tx-buffer** With this command the data of the virtual CAN buffer can be send. (`win_cmmd`=ID, `sub_cmmd` = length)

Table 4.3: Commands in the communication window

4.5 Status Bits

The status bits in the first byte of the communication window are representing the module and CAN bus status and are continuously updated. The bits and their meaning are the following:

Bit	Meaning
7	**TOGGLE** This bit is always inverted after a command (also a null command) has been handled, then the user can define the following command. A command is only recognized and handled after at least one bit of the four output bytes in the communication window has been changed.
6	**CAN bus Off**
5	**CAN Warn**
4	**STOP** The module is not yet or not anymore in ‘operational’ status. No telegrams from the normal entries (user parameters) are transmitted. Transmissions from the virtual CAN buffers, however, are possible. Generally all valid telegrams are received and transmitted to the PLC.
3	**Rx-FIFO not empty** Data has been received in the FIFO and can be read out.
2	**Rx-FIFO overrun** The FIFO cannot receive data from the CAN bus anymore, because it is full, ie. data is faster written into the FIFO than read out. The new data is lost as long as the old data is not being read out, making ‘place’ for new data.
1	not used
0	**SUCCESS** When this bit is set, the last command has been executed successfully. Otherwise an error has occurred. This bit is set or deleted synchronously to the toggle bit.

Table 4.4: Status bits

4.6 Examples

The following frame conditions have been assumed for the three examples below:

PLC-basis address of CSC595/2-module: $40 (DEC64)

4.6.1 Configuration Example

The CSC595/2-module has been configured as a CAN master by the FL-command, without any autostart function, however. It is now to be activated by means of the start command via the PLC.

Entry	Comment
L KH 0101	load Hex constant, win_cmmd = 1, sub_cmmd = 1
T AW 64	transfer value to output address 64

4.6.2 Transmitting Data to the CAN bus

Four data bytes containing '$1234.5678' are to be transmitted via CAN Tx-identifier '$012A'. Ion order to execute this transfer four PLC cycles are required (for the first transmission), because only two bytes command and two bytes data can be written per each cycle.

Entry	Comment
1. cycle: Writing the first data word into the data buffer:	
`L KH 0401`	load Hex constant, `win_cmmd` = 4, `sub_cmmd` = 1
`T AW 64`	transfer value to output address 64
	The sub-command is structured as follows: Sub-command = xxxx.xyyd with xxxx.x = 0000.0 for data buffer No. 0 yy = 00 because offset within buffer is to be 0 d = 1 for selecting the data buffer From this the sub-command results 0000.0001 = $01
`L KH 1234`	1. data word = $1234
`T AW 66`	transfer value to output address 66
2. Cycle: Writing the second data word into the data buffer:	
`L KH 0403`	`win_cmmd` = 4, `sub_cmmd` = 3
`T AW 64`	
	The sub-command is structured as follows: Sub-command = xxxx.xyyd with xxxx.x = 0000.0 for data buffer No. 0 yy = 01 because offset within the buffer is to be 2 d = 1 for selecting the data buffer From this the sub-command results 0000.0011 = $03
`L KH 5678`	2. data word = $5678
`T AW 66`	

3. Cycle: Writing the Tx-identifier into the control buffer:

```
L KH 0402          win_cmmd = 4, sub_cmmd = 2
T AW 64

L KH 012A          Identifier = $012A
T AW 66
```

4. Cycle: Starting transmission by defining length and transmission command:

```
L KH 0400          win_cmmd = 4, sub_cmmd = 0
T AW 64
                   The sub-command is structured as follows:
                   Sub-command = xxxx.xyyd with
                   xxxx.x =  0000.0 for control buffer No. 0
                   yy =      00 because offset within buffer is to be 0
                   d =       0 for selecting the control buffer
                   From this the sub-command results 0000.0000 = $00

L KH 0480          length = 4, buff_cmmd = $80
T AW 66
```

4.6.3 Reading Data from the CAN bus

The data received on the CAN identifier ‘$012B’ are to be stored and read in the data buffer starting from buffer offset $8. In order to execute this transfer you need six PLC cycles, because from transmitting the first ‘read command’ of the PLC to the CSC595/2-module until the first read data is available three cycles pass in the PLC. By putting commands together *only* six instead of 12 cycles are required. The following diagram shows the problem:

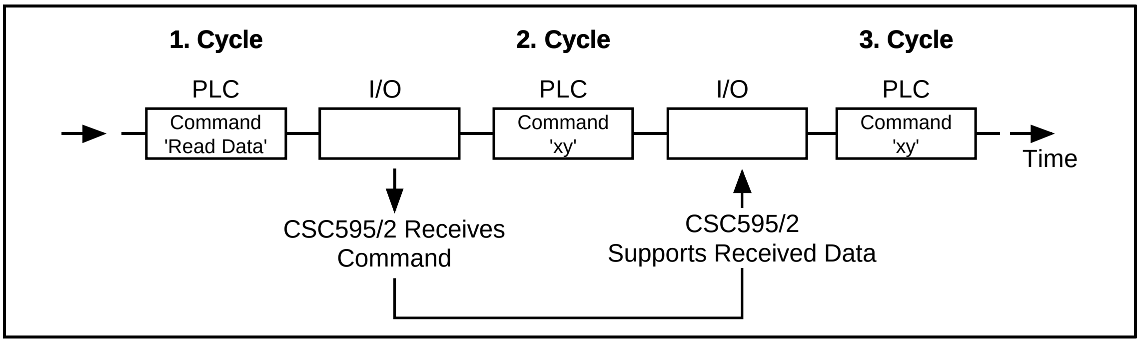


Fig. 4.1: Asynchronous reading of CAN data

Entry	Comment
Initialisation cycle:	set up Rx-buffer for data of the identifier
L KH 0209	load Hex constants, win_cmmd = 2, sub_cmmd = 9
T AW 64	transfer value to output address 64
	The sub-command is structured as follows: Sub-command = xxxx.x001 with xxxx.x = 0000.1 for data buffer No. 1 (from $8) From this the sub-command results 0000.1001 = $09
L KH 012B	define identifier $012B in parameter word
T AW 66	transfer value to output address 66

1. Cycle: Call command for displaying the first data word from the data buffer

L KH 0309 — load Hex constant, win_cmmd = 3, sub_cmmd = 9
T AW 64

The sub-command is structured as follows:
Sub-command = xxxx.xyyd with
xxxx.x = 0000.1 for data buffer No. 1 (from $8)
yy = 00 for first data word to be read
d = 1 for selecting the data buffer
From this the sub-command results 0000.1001 = $09

2. Cycle: Call command for displaying the second data word from the data buffer

L KH 030B — load Hex constant, win_cmmd = 3, sub_cmmd = B
T AW 64

The sub-command is structured as follows:
Sub-command = xxxx.xyyd with
xxxx.x = 0000.1 for data buffer No. 1 (from $8)
yy = 01 for second data word to be read
d = 1 for selecting the data buffer
From this the sub-command results 0000.1011 = $0B

3. Cycle: Call command for displaying the third data word from the data buffer and display first data word

L KH 030D — load Hex constant, win_cmmd = 3, sub_cmmd = D
T AW 64

The sub-command is structured as follows:
Sub-command = xxxx.xyyd with
xxxx.x = 0000.1 for data buffer No. 1 (from $8)
yy = 10 for third data word to be read
d = 1 for selecting the data buffer
From this the sub-command results 0000.1101 = $0D

L EW 66 — display the **first** data word from the data buffer

4. Cycle: Call command for displaying the fourth data word from the data buffer and display the second data word

```
L KH 030F       load Hex constant, win_cmmd = 3, sub_cmmd = F
T AW 64
                The sub-command is structured as follows:
                Sub-command = xxxx.xyyd with
                xxxx.x =  0000.1 for data buffer No. 1 (from $8)
                yy =      11 for the fourth data word to be read
                d =       1 for selecting the data buffer
                From this the sub-command results 0000.1111 = $0F

L EW 66         display the second data word from the data buffer
```

5. Cycle: Display third data word

```
L EW 66         display the third data word from the data buffer
```

6. Cycle: Display fourth data word

```
L EW 66         display the fourth data word from the data buffer
```

5. Sequential Data Exchange - Rx-FIFO

5.1 Rx-FIFO

5.1.1 Overview

The Rx-FIFO has been designed to provide a sequential handling of CAN-Rx data received. In contrast to the reception of Rx-frames via object buffers the old data is not overwritten by the new data of the CAN-ID each time.

The FIFO has been realized via software in a RAM memory. Up to 1023 CAN telegrams by any identifier can be stored in the chronological order of their reception.

The FIFO can be useful in the following applications, for instance:
- sporadic reception of various CAN-Ids, if an action is to be triggered because of the reception of data
- reception of fragmented data, such as data of a serial interface which sequentially transmits the data to be transmitted on a constant CAN identifier

The reception of new data can be seen by evaluating the bit ‘Rx-FIFO not empty’ in module status in the communication window. The module-status byte can be read as a digital input, for instance, and the status of the ‘Rx-FIFO not empty’ bit can be linked to an acknowledge condition.

The data transfer via the Rx-FIFO is a little bit faster than via the virtual CAN buffer.

5.1.2 Operation

When the data of one or more Rx-identifiers are to be stored into the Rx-FIFO, it is useful to operate as follows:

1. Defining a PLC address from which the data is to be read.
   In order to read data from the FIFO a PLC address is required with read access. This access is not intended to establish a ‘real’ link with a CAN identifier, therefore you can use the identifier $7FE. It is useful to select the number of bytes as data length which corresponds to the largest expected number of bytes to be received. If you want to make sure that you will get all data you therefore have to enter a length of eight bytes.

2. Defining the Rx-identifier/s which/whose data are to be collected in the Rx-FIFO.
   You can choose any number of Rx-identifiers to store their data in the Rx-FIFO. The identifiers are selected via the communication window by means of the command ‘Add ID to Rx-FIFO’.

FI FO	**Rx-FIFO**

3. Program loop:

3.1. Evaluating the status flag ‘Rx-FIFO not empty’
This bit signalizes that data can be read out of the FIFO.

3.2 Call command for reading the FIFO.
Via the communication window the command ‘Get next message from Rx-FIFO’ is called.

3.3 Wait two PLC cycles until data is available.

3.4 Read Rx-data

3.4.1 Read data length and identifiers via the communication window.
The number of data bytes and identifiers of the current readable message of the FIFO is read via the communication window. If the basis address is for instance $40 (DEC64), the ‘Rx-FIFO not empty’ flag, the length and the identifier would be read as followed:

EB 64	read the status bytes
EB 65	read the number of data bytes received (length)
EW 66	read the Rx-identifier

3.4.2 Read data on the PLC address defined under 1.
Here the number of data bytes is returned which has been defined under 1.. Only the number of bytes which has been read under 3.4.1 is valid currently!

Attention:	Commands will only be recognized, if at least one bit in the communication window is changed. In order to ensure this it is recommendable to toggle an unused bit (e.g.: in the parameter word) with every command call.

5.2 Tx-FIFO

The Tx-FIFO has been realized in a RAM memory area via software. It can buffer up to 255 CAN frames.

The data of the Tx-FIFO are transmitted sequentially, ie. when for instance two sequential frames with the same identifier are written into the FIFO, the first frame will not be overwritten by the second one, even if the first frame could not be transmitted until now. Therefore no Tx-frame will be lost, even if no CAN warning occurred.

Another advantage of the Tx-FIFO function is the fact that the transmission procedure is faster than via the command (`buff_cmmd`) $80 of the virtual CAN buffer.

6. Default Settings/Parameters after a RESET

6.1 Performance after RESET

The CSC595/2-module operates in standard by the micro controller C167C. Optionally, a second controller, the 82527 can be equipped (required for receiving RTRs).

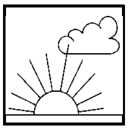
After a RESET of the complete module, the C167C starts running, reads the I^2C-EEPROM, evaluates it and starts the configuration in the SRAM. Then remote requests are transmitted on all CAN-Rx identifiers after the network has started, in order to update the input data.

The C167C then transmits ‘0' on all defined CAN-Tx identifiers first to reset all output units.

The C167C can also evaluate the PLC enabling signal (not in default status). If the enabling signal is deactivated, the reset mechanism of the output units, mentioned above, will be activated.

The enabling signal of the PLC bus is normally deactivated in the following cases:

1. Switch movement at PLC: "*RUN*" -> "*STOP*"
2. Bus error on the peripheral bus
3. Switching off the PLC



Start after RESET

6.2 Default Settings when Module is Shipped

The following table shows the setting of variable parameters at the moment the module is shipped. The following chapter will describe how this default setting can be re-established after the parameters have been changed.

CAN identifiers	none defined
CAN-bit rate	1 Mbit/s
PLC simulation	CSC595/2 simulates an analog I/O-module with 2 bytes length
Slot No.	0
Wake-up time	500 msec
Module No.	= setting of coding switch
Flags	only CAN-layer-2 operation

Table 4.1: Default setting of the module

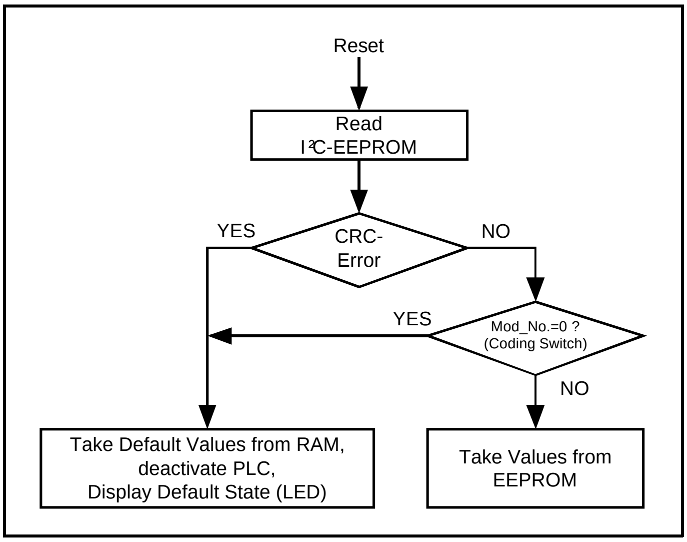
6.3 Re-establishing the Default Parameters (Default Setting)

The default setting of parameter as been described in the previous chapter can be re-established in three different ways. In order to activate the default parameter they have to be stored into the I²C-EEPROM by means of the Save-Config-command (CS) after having been selected.

1. Activating via Local Software

When the controller detects that the data stored in the I²C-EEPROM is wrong (e.g. wrong checksum (note: There is no check as regards contents.)), it selects the default parameters. This status is shown by a flashing LED (500 ms green, 100 ms red).
This is en error status and the module is therefore inactive at the PLC interface.

The following figure represents the connection between CRC-check of the I²C-EEPROM and the coding-switch setting:



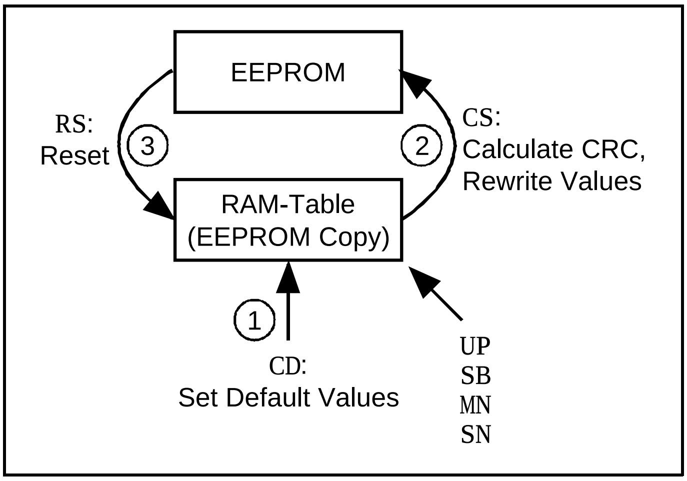
2. Activation by the User

2.1 Via Coding Switch

- Switch off module
- Set coding switch to ‘0'
- Switch module on again
- Status as under 1.

2.2 Via a Terminal at the RS-232 Interface

- call the command ‘CAN default’ (CD) via the terminal interface



As mentioned above, regardless of the way the default parameters have been selected they have to be stored in the I²C-EEPROM by means of the command CS. Then a RESET has to be triggered to activate the stored parameters.

Appendix

A.1 Loading CSC595/2-Firmware Updates

The program code of the CSC595/2-module is stored into a Flash EPROM. When the module is shipped the Flash EPROM has already been loaded with the current firmware-program code.

Updates of the software can be installed by the users. The update is shipped in S-Record format together with an installation program on a 3.5" disk. For the installation you need a PC with at least one serial interface (MS-DOS or DOS window).

A.1.1 Connecting the CSC595/2-Module to the PC

The CSC595/2-module has a serial interface (RS-232) to which the PC is connected and via which the program is booted into the module. A detailed description on connecting a PC to the CSC595/2-module can be taken from the hardware manual.

The notes on connecting the CSC595/2-module to the PC, as given in the hardware manual of the module, have to be followed to ensure a faultless operation and prevent damage at the devices!

A.1.2 Procedure

I. Preparation

1. Create a directory for the data of the update disk on your PC (e.g. `\CSC5952\`).
2. Copy <u>all</u> files of the disk into this directory.
3. If you have not installed the hardware of the CSC595/2-module so far, start now (see hardware manual of the module, chapter ‘Installation’). Please remember the interface configuration (baudrate 19200 baud etc.).
4. Insert coding pin for bootstrap loader (as far as possible). The pin will only be correctly interpreted after a RESET (power down) of the PLC.
5. Switch PLC on again (i.e. set switch to ‘stop status’ to prevent actions being executed), switch PC on again.

II. Program Call and Commands

1. Change into the created CSC595/2 directory of the PC.

2. If the baudrate of the CSC595/2-interface is correctly configured, you can now call the interactive update program ‘update.exe’.
   Depending on the serial port of the PC you have selected one of the following programs has to be called:
   CSC595/2 at COM1 -> >update81< (or only update.exe)
   CSC595/2 at COM2 -> >update82<

3. The update program is running automatically and uses, e.g., the flashing status of the C67C-status LED to lead the operation. When a confirmation or negation of the user is expected, it has to be entered via the keys >Y< for ‘yes’ or >N< for ‘no’ (note: The program evaluates all keys apart from ‘N’ as confirmation (‘yes’).).

4. After it has been successfully programmed, the update program is terminated by the screen output ‘ok’.

5. Starting the CSC595/2-module with the new software:
   Draw the coding pin and trigger a RESET by switching the PLC on and off.

End of update.

Attention:
An update may contain an extension of the data stored in the I^2C-EEPROM. By this the checksum calculation changes and the module wakes up with a default RESET. In this case the module has to be configured again.