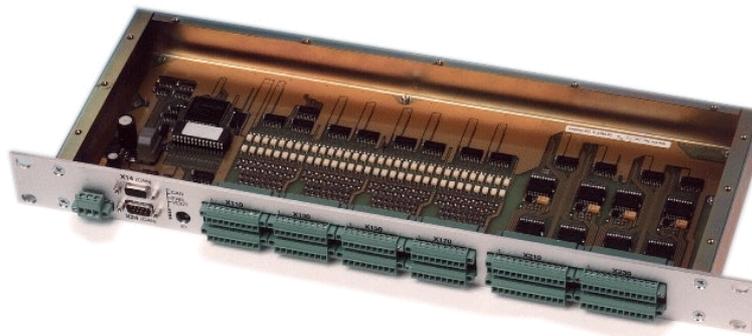


CAN-Control-I/O



Hardware Manual

to product: C.2071.xx

Manual File:	I:\texte\Doku\MANUALS\CAN\CAN-Control\IO\can-control-io_13h.en9
Date of Print:	31.03.2005

Described PCB Version:	CAN-Control-I/O Rev. 1.0
-------------------------------	--------------------------

Changes in the Chapters

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

Manual rev.	Chapter	Changes with respect to previous revision
1.1	-	Connector names have been changed according to front panel text.
1.2	-	Pin numbers of plug-in connectors added.
1.3	1.2.4	Load value of digital outputs corrected.

Further technical changes are subject to change without notice.

NOTE

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. 207

30165 Hannover

Germany

Phone: +49-511-372 98-0

Fax: +49-511-372 98-68

E-mail: info@esd-electronics.com

Internet: www.esd-electronics.com

USA / Canada:

esd electronics Inc.

12 Elm Street

Hatfield, MA 01038-0048

USA

Phone: +1-800-732-8006

Fax: +1-800-732-8093

E-mail: us-sales@esd-electronics.com

Internet: www.esd-electronics.us

Content

Page

1. Overview	3
1.1 Module Description	3
1.2 Summary of the Technical Data	4
1.2.1 General Technical Data	4
1.2.2 CAN- and μ Controller Circuits	5
1.2.3 Digital Inputs	6
1.2.4 Digital Outputs	7
1.2.5 Digital Output Driver Supply Voltage Monitor Circuit	7
1.2.6 LED Display	8
1.3 Order Information	9
2. Description of the Units	11
2.1 Rotary Coding Switch and Jumper	11
2.1.1 Circuit Location	11
2.1.2 Function of the Coding Switch	12
2.1.3 Function of Jumper J220	12
2.2 Digital Outputs	14
2.3 Digital Inputs	15
2.4 CAN Interface	16
3. Appendix	17
3.1 Connector Pin Assignments	17
3.1.1 CAN Connector X24 (9-pole DSUB Male)	17
3.1.2 CAN Connector X14 (9-pole DSUB Female)	18
3.1.3 I/O Connectors X100...X170 (MCD 1,5/9-G1F-3,81)	19
3.1.3.1 Inputs I0..I15 at X100 and X110	19
3.1.3.2 Inputs I16..I31 at X120 and X130	19
3.1.3.3 Inputs I32..I47 at X140 and X150	20
3.1.3.4 Inputs I48..I63 at X160 and X170	20
3.1.4 I/O Connectors X200...X230 (MCD 1,5/12-G1F-3,81)	21
3.1.4.1 Outputs Q0...Q15 at X200 and X210	21
3.1.4.2 Outputs Q16...Q31 at X220 and X230	21
3.1.5 Supply Voltage Connector X10 (MSTB 2,5/3-GF-5,08)	22
3.2 Front Panel	23

This page is intentionally left blank.



1. Overview

1.1 Module Description

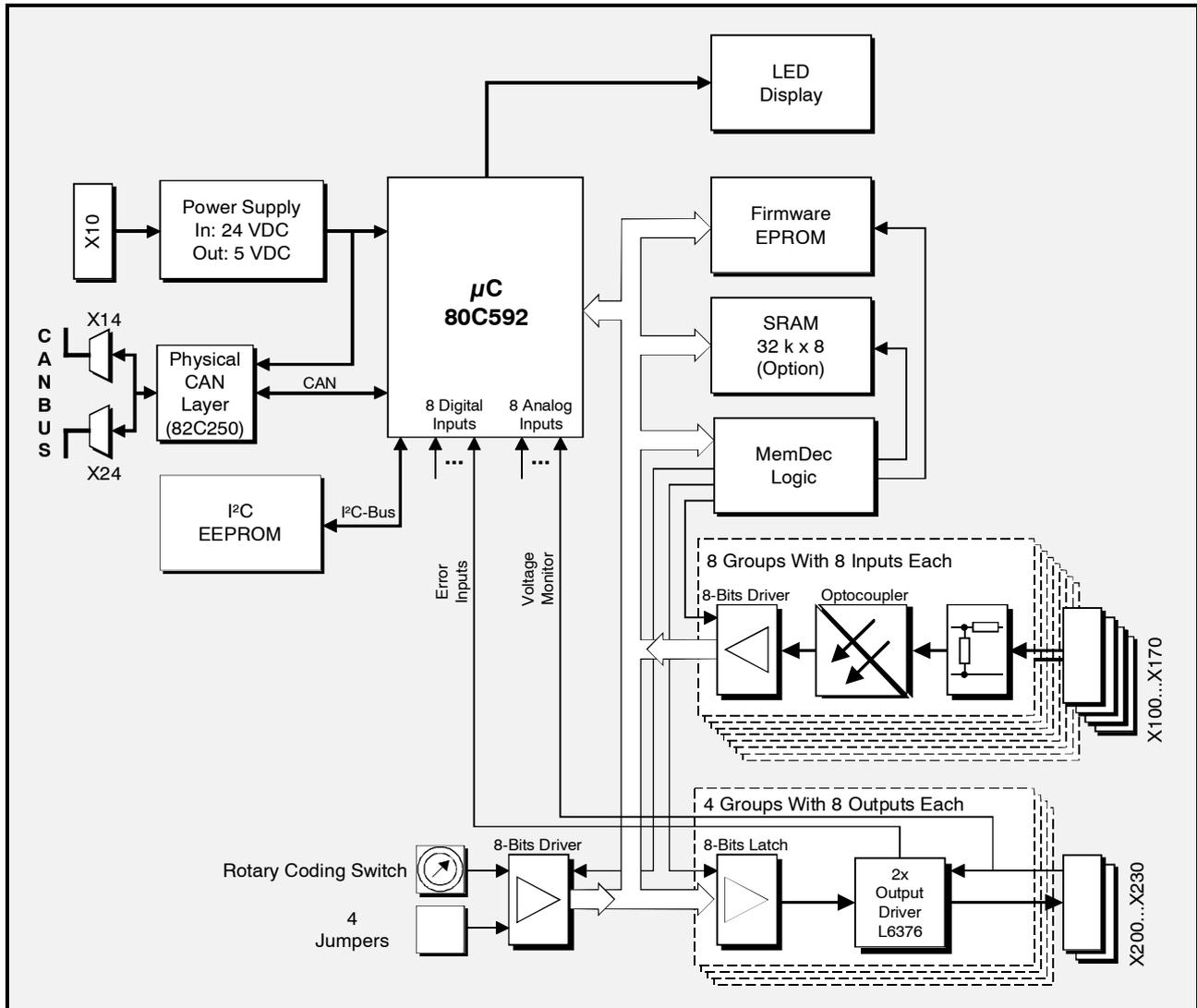


Fig. 1.1.1: Block diagram of the CAN-Control-I/O module



1.2 Summary of the Technical Data

1.2.1 General Technical Data

Temperature range	permissible ambient temperature: 0...50 °C
Humidity	max. 90%, not condensing
Operating voltage	nominal voltage 18 V...30 VDC power consumption (typical, at 20 °C): approx. 1.5 W, electronical reverse battery protection safety fuse 200 mA/T
Connectors	X10 (MDSTB/2,5/3-GF-5,08) - supply voltage X24 (DSUB9/Male) - CAN connection X14 (DSUB9/Female) - CAN connection X100, X110 (MCD 1,5/9-G1F-3,81) - digital Inputs I0...I15 X120, X130 (MCD 1,5/9-G1F-3,81) - digital Inputs I16...I31 X140, X150 (MCD 1,5/9-G1F-3,81) - digital Inputs I32...I47 X160, X170 (MCD 1,5/9-G1F-3,81) - digital Inputs I48...I63 X200, X210 (MCD 1,5/12-G1F-3,81) - digital Outputs Q0...Q15 X220, X230 (MCD 1,5/12-G1F-3,81) - digital Outputs Q16...Q31
Case	dimensions: 19" / 1 HE / max. 250 mm all connectors and switches are front accessible
Weight	approx. 550 g (PCB without case)

Table 1.2.1: General data of the module



1.2.2 CAN- and μ Controller Circuits

CAN Interface	physical layer according to ISO 11898, without electrical isolation of CAN signals
Transmission rate	selectable via jumpers or programmable from 10 kbit/s to 1 Mbit/s
CAN Identifier	11-bit identifier, default value can be set via rotary coding switch, programming via CAN is also possible
μ Controller	8xC592, 16 MHz
EPROM	firmware storage
EEPROM	for CAN parameter and user parameter storage
SRAM	option: memory capacity 32 kbytes

Table 1.2.2: Technical data of the CAN- and μ controller circuits



1.2.3 Digital Inputs

Number	64 in groups with 8 inputs each
Input circuit	refer page 15
Inputs voltages	inputs voltage range: max. permissible: $-30\text{ V} \leq U_{\text{IN}} \leq 33\text{ V}$ switching threshold input = '0': $U_{\text{IN}} < 9\text{ V}$ switching threshold input = '1': $U_{\text{IN}} > 15\text{ V}$
Input current	input = '1': $I_{\text{ON}} = \text{approx. } 3.5\text{ mA}$ (typical, at 18 V) $I_{\text{ON}} = \text{approx. } 10\text{ mA}$ (typical, at 30 V)
Circuit protection	electrical isolation by optocouplers, each 8 inputs use one common GND
Filter	option: RC with $\tau_{\text{on}[24\text{V}]} \approx 500\text{ ns}$

Table 1.2.3: Input specification



1.2.4 Digital Outputs

Output Circuit Specification

Number	32 (4 groups with 8 outputs each)		
Outputs circuit	refer to page 14		
Driver	L6376 4 outputs/circuit		
Supply voltage	maximum voltage rating: $U_{VCC} = 9.5 \text{ VDC} \dots 35 \text{ VDC}$		
Load	nominal current/channel (24 V):	0.5 A	(50 °C)
	shut-down current/channel (24 V):	0.65 A ... 1.2 A	(50 °C)
Protection circuit	short circuit protection, overload (thermal)		
Status messages	message in case of error (low supply voltage, over current, over temperature)		
Timing	$U_{VCC} = 24\text{V}$, $R_L = 47 \Omega$: rising edge: 3 V ... 7 V/ μs falling edge: 4 V ... 10 V/ μs		

Table 1.2.4: Output specification

1.2.5 Digital Output Driver Supply Voltage Monitor Circuit

Number	8 output supply voltages of the digital output drivers are monitored		
Converter	internal A/D converter of the CAN micro controller		
Resolution	converter's resolution: 10 Bit		
Precision	$\pm 50 \text{ mV}$		
Measuring range	unipolar:	0... $\approx +50 \text{ V}$	

Table 1.2.5: Specification of the analog inputs



1.2.6 LED Display

Display function	Marking	Colour
CAN status LED 1	1	red
CAN status LED 2	2	green
Digital out error	3	red
VCC-OK	4	green

Table 1.2.6: LED Display



1.3 Order Information

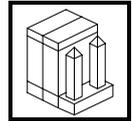
Type	Properties	Order No.
CAN-Control-I/O	64 digital inputs 24 V, 32 digital outputs 24V/0.5A	C.2071.01
CAN-Control-I/O-ME	User manual in English ^{1*)} (this manual)	C.2071.21
CAN-Control-I/O-ENG	Engineering Manual in English ^{2*)} Contents: Circuit diagrams, PCB top overlay drawing, data sheets of significant components	C.2071.25

1*) If module and manual are ordered together, the manual is free of charge.

2*) This manual is liable for costs, please contact our support.

Table 1.3.1: Order information

This page is intentionally left blank.



2. Description of the Units

2.1 Rotary Coding Switch and Jumper

2.1.1 Circuit Location

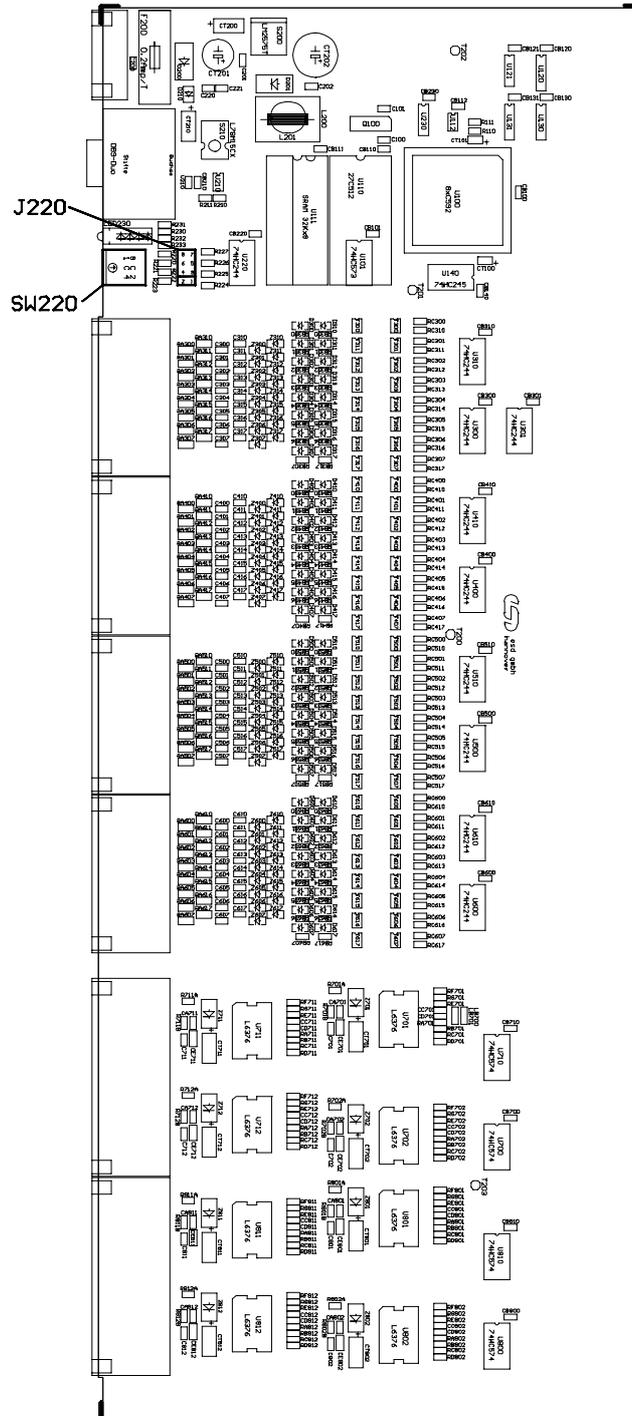
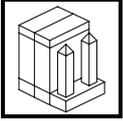


Fig. 2.1.1: Location of the coding switch and the jumper



Jumper Settings

2.1.2 Function of the Coding Switch

Via the coding switch the default value of the CAN identifier bits Id4 to Id7 (counting 1 to 11) is set.

The coding switch is factory set to '0'.

A detailed description of the identifier settings can be read in the software manual of this module.

2.1.3 Function of Jumper J220

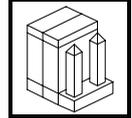
Via the jumper J220 the default bit rate of the CAN is set. The bit rate can be programmed by using the esd-CAN protocol (refer to manual 'esd-CAN Protocol').

Jumper	Bit	Function
J220	8 7 ☒ ☒	→ D3
	☒ ☒	→ D2
	☒ ☒	→ D1
	2 1 ☒ ☒	→ D0
		Bit rate setting

Bit Rate (D0-D3):

A set jumper sets the according bit to value '1'.

The specified typical line lengths base on experimental values from experience. The minimum reachable line lengths results from the 'worst case' delay times of the used components.



The default setting of the module as factory-set is shown in **bold** print.

Jumper bits				8xC592 register		Bitrate [kbit/s]	Typical values of the reachable line length l_{\max} [m]	Minimum reachable line length l_{\min} [m]
D3	D2	D1	D0	BTR0 [HEX]	BTR1 [HEX]			
0	0	0	0	00	14	1000	37	20
0	0	0	1	00	18	666.6	80	65
0	0	1	0	00	1C	500	130	110
0	0	1	1	01	18	333.3	180	160
0	1	0	0	01	1C	250	270	250
0	1	0	1	02	1C	166	420	400
0	1	1	0	03	1C	125	570	550
0	1	1	1	04	1C	100	710	700
1	0	0	0	45	2F	66.6	1000	980
1	0	0	1	09	1C	50	1400	1400
1	0	1	0	4B	2F	33.3	2000	2000
1	0	1	1	18	1C	20	3600	3600
1	1	0	0	5F	2F	12.5	5400	5400
1	1	0	1	31	1C	10	7300	7300
1	1	1	0	00	16	800	59	42

The specifications in the table base on the limit values of the bit timing of the CAN protocol, the delay time of the local CAN interface and the delay times of the cable. The delay time of the cable is assumed with ca. 5.5 ns/m. Further influences, caused, for instance, by missing terminal resistors, the specific resistor, the cable geometry or outer disturbances, have not been taken into consideration in the transmission!

Table 2.1.1: Setting the bit rate via J220



Digital Outputs

2.2 Digital Outputs

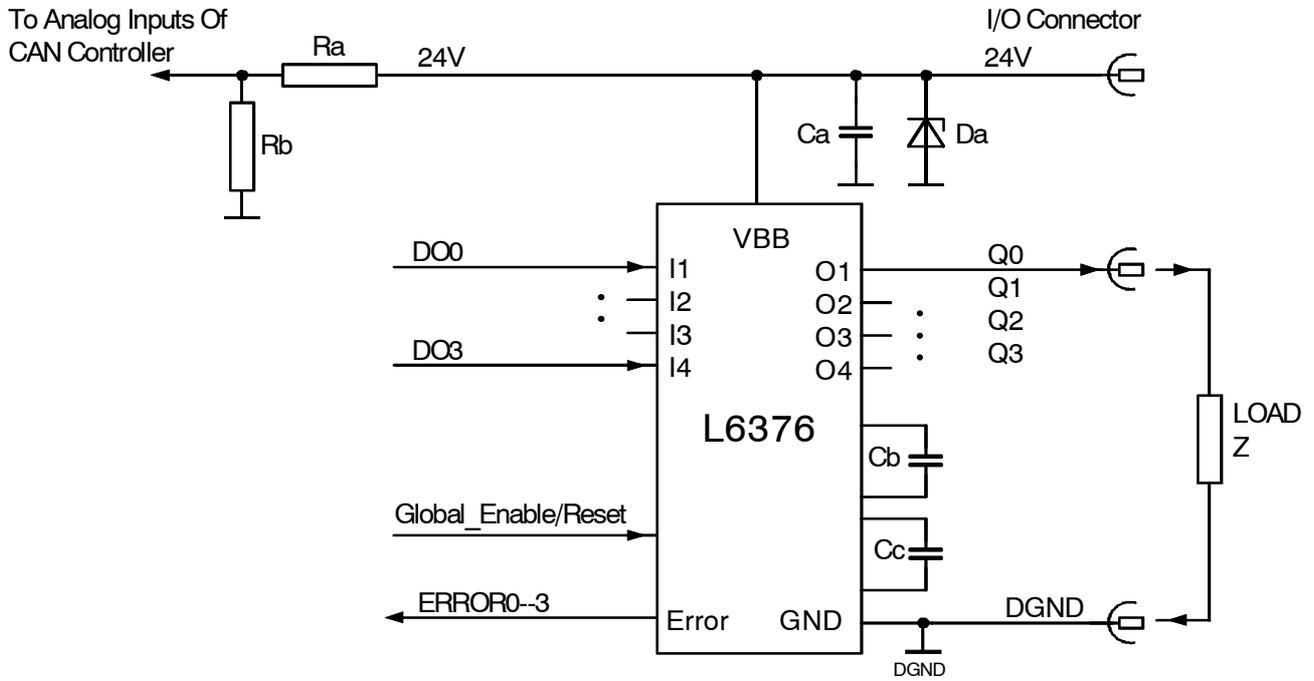


Fig. 2.2.1: Circuit of digital outputs

Circuit values:

Voltage divider

$$R_b / (R_a + R_b) \approx 1/10$$

Protection diode D_a :

$$U_z = 33 \text{ V (max. 50 V)}$$



2.3 Digital Inputs

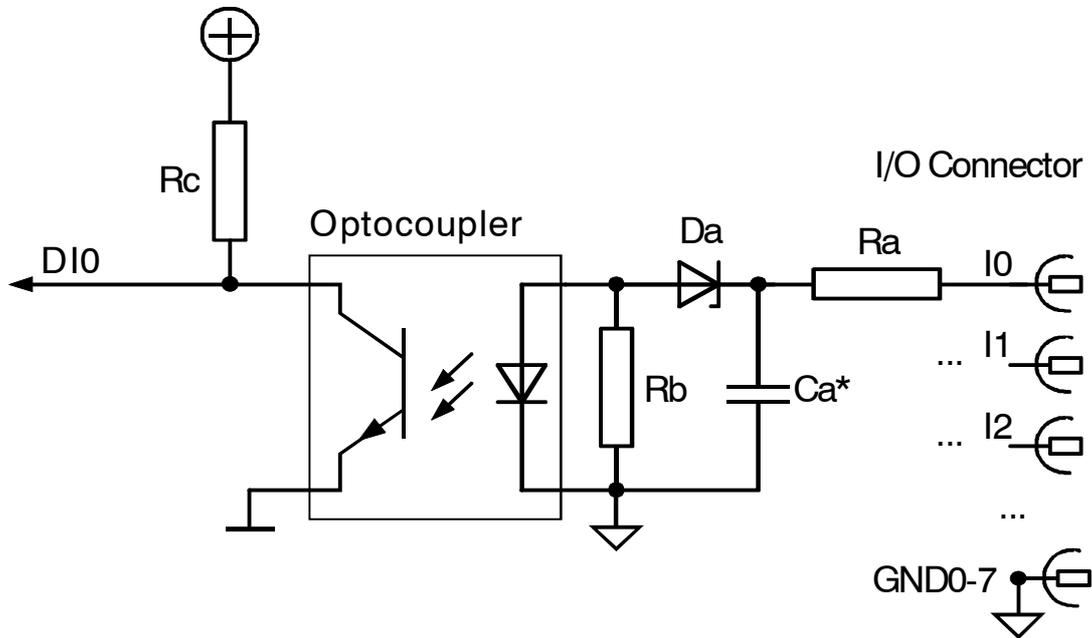
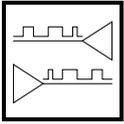


Fig. 2.3.1: Circuit of digital inputs (example: input 0...7)

The capacitor Ca* is equipped as an option.



CAN Interface

2.4 CAN Interface

The physical interface is designed according to ISO 11898 (without electrical isolation). The CAN interface can be accessed by two 9-pole DSUB connectors.

As an option external CAN interfaces can be supplied by the CAN-Control-I/O module using the CAN line CAN_V+. The supply voltage is generated by a 15 V- linear voltage controller.

The following figure shows the basic wiring of the CAN connectors at the CAN-Control-I/O module:

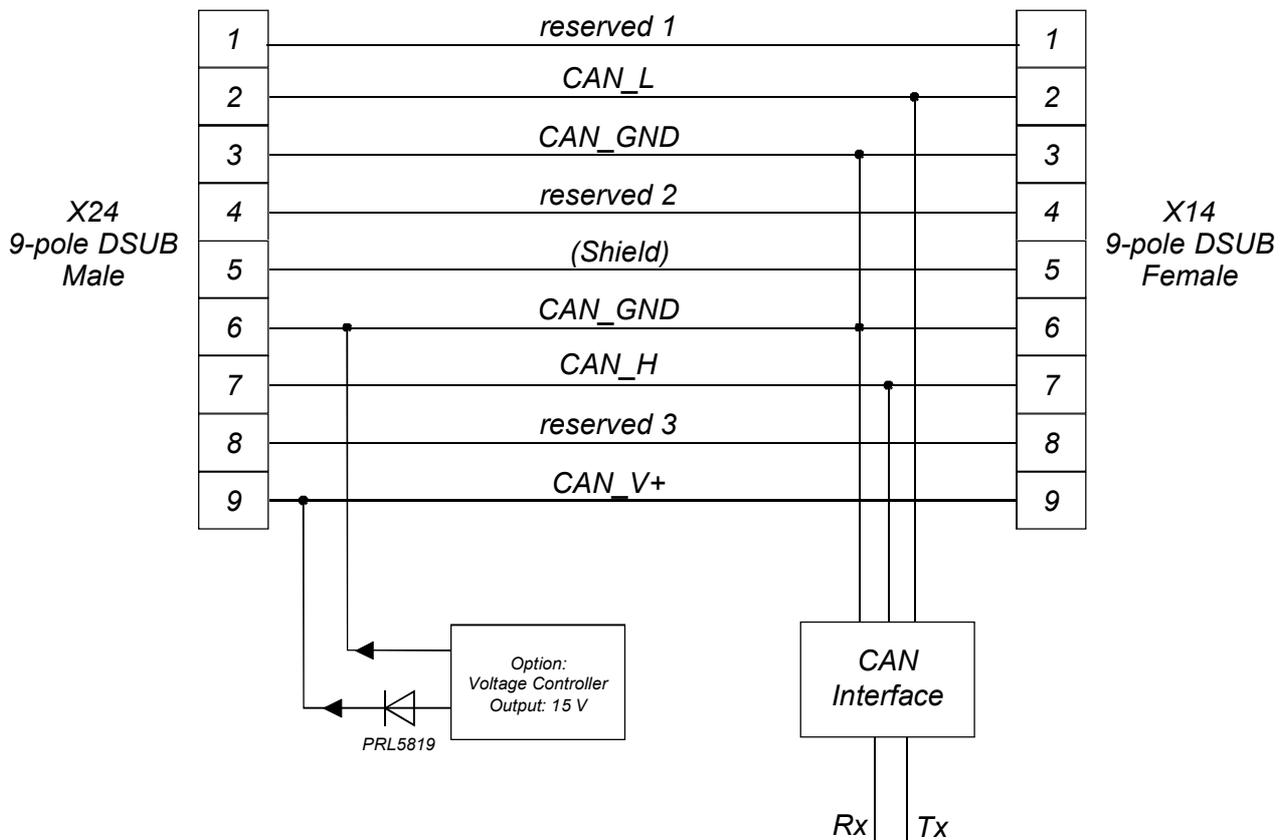
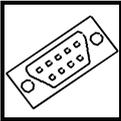


Fig. 2.4.1: Principle block diagram of CAN connector wiring

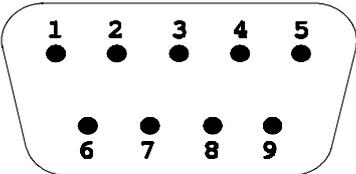


3. Appendix

3.1 Connector Pin Assignments

3.1.1 CAN Connector X24 (9-pole DSUB Male)

Pin Location:



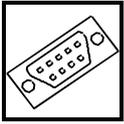
Pin Assignment:

Signal	Pin		Signal
CAN_GND	6	1	reserved 1
		2	CAN_L
CAN_H	7	3	CAN_GND
reserved 3	8	4	reserved 2
CAN_V+	9	5	(Shield)

9-pole DSUB connector

Signal Description:

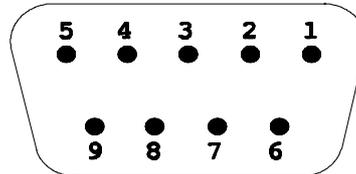
- CAN_L, CAN_H ... CAN signals
- CAN_GND ... reference GND of the physical layer
- CAN_V+ ... Option: supply voltage output for external CAN interfaces
- (Shield) ... pin for wiring the connectors shields
- reserved x ... pins, that are reserved for future use. Pins with the same name at X14 and X24 are connected.



Connector Pin Assignment

3.1.2 CAN Connector X14 (9-pole DSUB Female)

Pin Location:



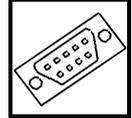
Pin Assignment:

Signal	Pin		Signal
reserved 1	1	6	CAN_GND
CAN_L	2		
CAN_GND	3	7	CAN_H
reserved 2	4	8	reserved 3
(Shield)	5	9	CAN_V+

9-pole DSUB Female

Signal Description:

CAN_L, CAN_H ...	CAN signals
CAN_GND ...	reference GND of the physical layer
CAN_V+ ...	Option: supply voltage output for external CAN interfaces
(Shield) ...	pin for wiring the connectors shields
reserved x ...	pins, that are reserved for future use. Pins with the same name at X14 and X24 are connected.



3.1.3 I/O Connectors X100...X170 (MCD 1,5/9-G1F-3,81)

The 64 digital inputs has to be connected to the I/O connectors X100...X170. Each male connector row is equipped with one separate female connector for one input group. The cables has to be connected by screws.

3.1.3.1 Inputs I0..I15 at X100 and X110

X110

upper row	Pin 1*)	10	11	12	13	14	15	16	17	18
	Pin 2*)	1	2	3	4	5	6	7	8	9
	Signal	I8	I9	I10	I11	I12	I13	I14	I15	GND 8-15

X100

lower row	Pin	1	2	3	4	5	6	7	8	9
	Signal	I0	I1	I2	I3	I4	I5	I6	I7	GND 0-7

MCD 1,5/9-G1F-3,81

1*) Pin numbers used in schematic diagrams.

2*) Pin numbers used for labelling plug-in connectors.

3.1.3.2 Inputs I16..I31 at X120 and X130

X130

upper row	Pin 1*)	10	11	12	13	14	15	16	17	18
	Pin 2*)	1	2	3	4	5	6	7	8	9
	Signal	I24	I25	I26	I27	I28	I29	I30	I31	GND 24-31

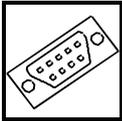
X120

lower row	Pin	1	2	3	4	5	6	7	8	9
	Signal	I16	I17	I18	I19	I20	I21	I22	I23	GND1 16-23

MCD 1,5/9-G1F-3,81

1*) Pin numbers used in schematic diagrams.

2*) Pin numbers used for labelling plug-in connectors.



Connector Pin Assignment

3.1.3.3 Inputs I32..I47 at X140 and X150

X150

upper row	Pin 1*)	10	11	12	13	14	15	16	17	18
	Pin 2*)	1	2	3	4	5	6	7	8	9
	Signal	I40	I41	I42	I43	I44	I45	I46	I47	GND 40-47

X140

lower row	Pin	1	2	3	4	5	6	7	8	9
	Signal	I32	I33	I34	I35	I36	I37	I38	I39	GND1 32-39

MCD 1,5/9-G1F-3,81

1*) Pin numbers used in schematic diagrams.

2*) Pin numbers used for labelling plug-in connectors.

3.1.3.4 Inputs I48..I63 at X160 and X170

X170

upper row	Pin 1*)	10	11	12	13	14	15	16	17	18
	Pin 2*)	1	2	3	4	5	6	7	8	9
	Signal	I56	I57	I58	I59	I60	I61	I62	I63	GND 56-63

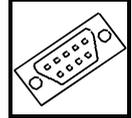
X160

lower row	Pin	1	2	3	4	5	6	7	8	9
	Signal	I48	I49	I50	I51	I52	I53	I54	I55	GND1 48-55

MCD 1,5/9-G1F-3,81

1*) Pin numbers used in schematic diagrams.

2*) Pin numbers used for labelling plug-in connectors.



3.1.4 I/O Connectors X200...X230 (MCD 1,5/12-G1F-3,81)

The 32 digital outputs are has to be connected via the I/O connectors X200...X230. Each male connector row is equipped with one separate female connector for two output groups. The cables has to be connected by screws.

3.1.4.1 Outputs Q0...Q15 at X200 and X210

X210

upper	Pin 1*)	13	14	15	16	17	18	19	20	21	22	23	24
	Pin 2*)	1	2	3	4	5	6	7	8	9	10	11	12
row	Signal	+24V Q8-11	Q8	Q9	Q10	Q11	GND Q8-11	+24V Q12-15	Q12	Q13	Q14	Q15	GND Q12-15

X200

lower	Pin	1	2	3	4	5	6	7	8	9	10	11	12
	Signal	+24V Q0-3	Q0	Q1	Q2	Q3	GND Q0-3	+24V Q4-7	Q4	Q5	Q6	Q7	GND Q4-7

MCD 1,5/12-G1F-3,81

1*) Pin numbers used in schematic diagrams.

2*) Pin numbers used for labelling plug-in connectors.

3.1.4.2 Outputs Q16...Q31 at X220 and X230

X230

upper	Pin 1*)	13	14	15	16	17	18	19	20	21	22	23	24
	Pin 2*)	1	2	3	4	5	6	7	8	9	10	11	12
row	Signal	+24V Q24-27	Q24	Q25	Q26	Q27	GND Q24-27	+24V Q28-31	Q28	Q29	Q30	Q31	GND Q28-31

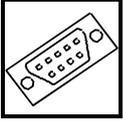
X220

lower	Pin	1	2	3	4	5	6	7	8	9	10	11	12
	Signal	+24V Q16-19	Q16	Q17	Q18	Q19	GND Q16-19	+24V Q20-23	Q20	Q21	Q22	Q23	GND Q20-23

MCD 1,5/12-G1F-3,81

1*) Pin numbers used in schematic diagrams.

2*) Pin numbers used for labelling plug-in connectors.



Connector Pin Assignment

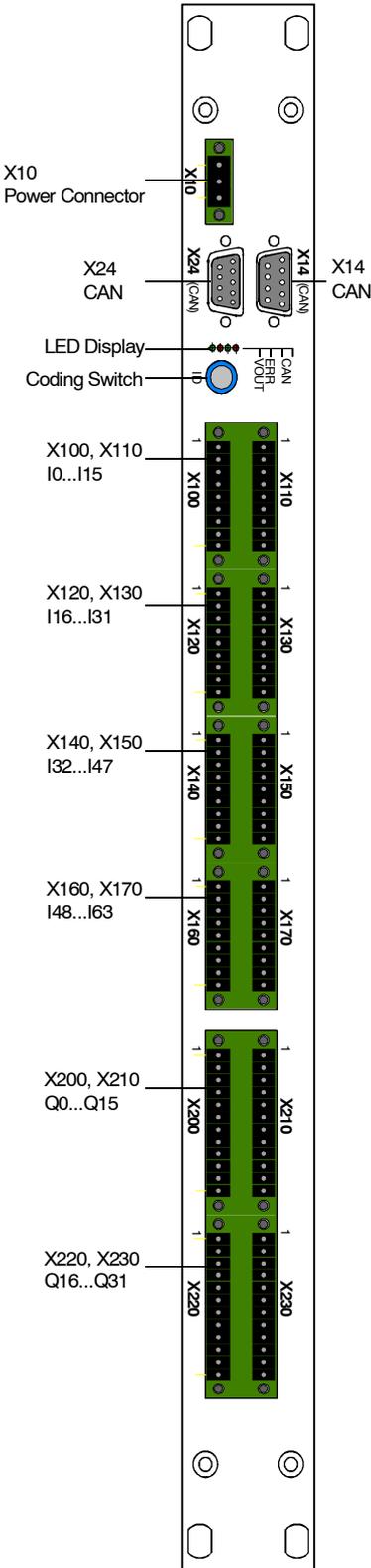
3.1.5 Supply Voltage Connector X10 (MSTB 2,5/3-GF-5,08)

Pin	Signal
1	GND
2	+24 V
3	GND

MSTB 2,5/3-GF-5,08



3.2 Front Panel



CAN-Control-I/O

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

Phone: +49-511-372-980

FAX: +49-511-633-650

E-mail: sales@esd-electronics.com

Internet: <http://www.esd-electronics.com>

This document shall not be duplicated, nor its contents used for any purpose, unless express permission has been granted.

Copyright by esd

Manual File:	I:\TEXTE\DOKU\MANUALS\CAN\LASCON\IO\LCIO-11S.EN6
Date of Print:	20.10.98

Described Software Revision:	
CAN Kernel :	see manual 'esd Protocol for CAN Modules'
esd Protocol :	
Module-specific Implementation:	Revision 'a'

The designation of the firmware implemented is labelled on the EPROM within the case of the module. The meaning of the characters is as follows:

```

                                CAN / lio6 1d a a
                                * * * *
                                * * * *
                                * * * *
Designation of module )          * * *
software (module name)          * * *
                                * * *
Revision no. of the CAN          * * *
kernel software (general )      * *
firmware)                        * *
                                * *
Revision no. of esd CAN          * *
protocol (general firmware) )    *
                                *
Revision letter of the module-
specific implementation )      *

```

The above shown example is from a CAN-Control-I/O module with the CAN kernel revision '1.d', esd protocol revision 'a' and the module-specific firmware revision 'a'.

Changes in the chapters

The changes in the user's manual listed below may encompass changes in the firmware, as well as changes in the description of the facts.

Firmware Manual Version	Chapter	Changes versus last manual revision
1.1	-	First English issue.

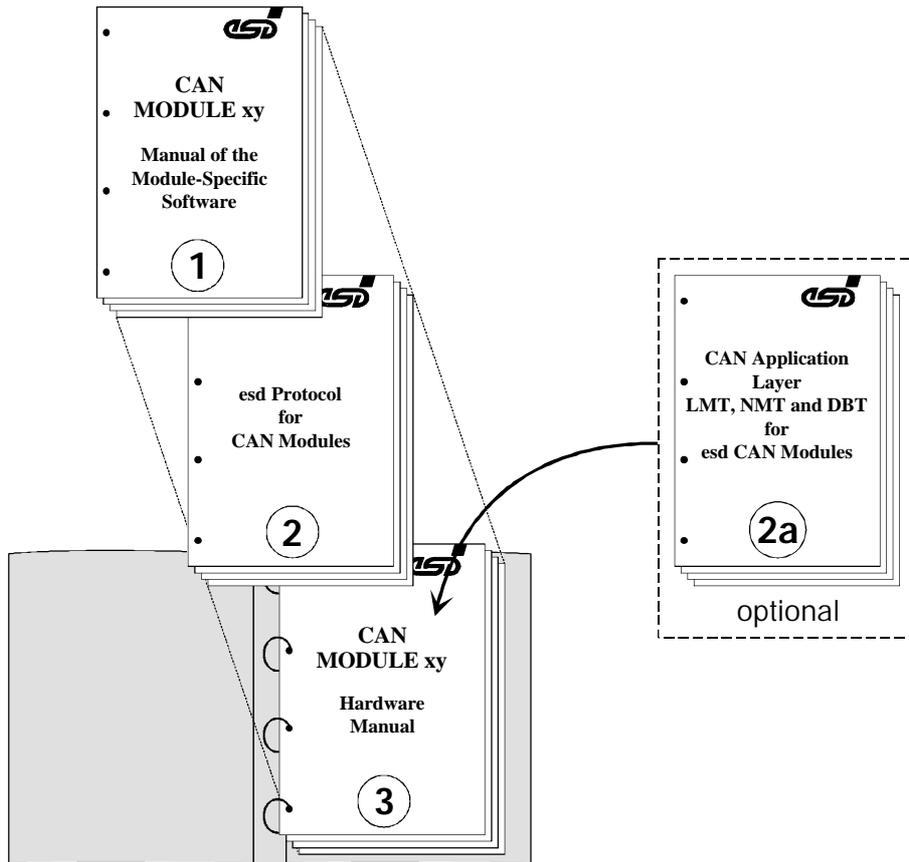
1. Overview	1 - 1
1.1 What is Where?	1 - 1
1.2 Default Settings	1 - 3
2. Local Software Mode of Operation	2 - 1
2.1 General on Setting and Reading the Digital Outputs and Inputs	2 - 1
2.1.1 Identifier	2 - 1
2.1.2 Module-No.	2 - 2
2.1.3 Count Mode of the Inputs and Outputs	2 - 3
2.2 Functions of the Tx identifiers	2 - 4
2.2.1 Start the Transmission of a Tx Frame	2 - 4
2.2.2 Transmitting the Input Status (I0...I64) by means of TxId1	2 - 5
2.2.3 Status Messages of Outputs by means of TxId2	2 - 6
2.2.4 Transmission the Output Driver's Supply Voltage Values at TxId3	2 - 8
2.3 Setting the Outputs by RxId1	2 - 9
3. User Parameter	3 - 1
3.1 Overview of the User Parameters	3 - 1
3.2 Description of the User Parameters	3 - 3
3.2.1 First Tx-Activate Delay (Parameter 0)	3 - 3
3.2.2 Mask for Output Error and Status Message (Parameter 2)	3 - 4
3.2.3 Setting of the Error-Hold-Time (Parameter 3)	3 - 5
3.2.4 Input Masks (Parameters 4..7)	3 - 6
3.2.5 Definition of the Voltage Thresholds (Parameters 8..15)	3 - 7
4. LED Display	4 - 1
4.1 Error LED	4 - 1
4.2 VCC OK LED	4 - 1



1. Overview

1.1 What is Where?

The description of the esd-CAN modules is divided into three manuals which are handed out in one file.



The first manual contains the software features and the parameters which are only significant for this module. The manual can be used independently from the chosen CAN protocol:

CAN-Control-I/O

Manual of the Module-Specific Software

Here, for instance, the functions of the type-specific firmware, the allocation of the COBs (CAL-Communication Objects) and identifiers and the allocation of the user parameters are described.



Overview

The second manual contains general software descriptions which are valid for all esd-CAN modules which are operated by means of 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, one of the following two manuals is therefore valid for the module:

The esd-CAN protocol is described in the manual:

esd Protocol for CAN Modules

It offers the user the possibility to parameterize the esd-CAN modules by means of an initialization identifier (\$700). By means of this protocol identifiers can be assigned to the module, user parameters can be set and watchdog functions can be activated.

Alternatively, the modules can be controlled by the CMS protocol. If this protocol is implemented (*), the manual

CAN-Application Layer LMT, NMT and DBT in esd Modules

for the CMS option is to be consulted. Here the translation of CMS services of the layer management (LMT), the network management (NMT) and the identifier distributor (DBT) in esd-CAN modules is explained.

The third manual contains the hardware description of the module. General as well as module-specific explanations about the hardware are included in the manual. For instance notes on installation and plug assignments can be found here.

CAN-Control-I/O

Hardware Manual

(*) At the moment (10.1997) a CMS implementation for this module is not yet available.



1.2 Default Settings

The default settings of the module are active, if one or more of the following conditions apply:

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

Individual parameters can be changed without affecting the default setting of other parameters. Changes in parameters only remain active after a RESET, if they have been stored in the EEPROM.

Default values in operation of the module with the esd-CAN protocol	
INIT Id.	in all operating modes \$700
Identifier (*)	3 TxIds and 1 RxId The default value of the identifiers is determined by the setting of the coding switch in the front panel. The 4 identifiers are distinguished by means of the identifier bits id1 and id2.
Module no.	= setting of the coding switch
CAN bitrate	= setting of the jumper J220 (1 Mbits/s)

(*) When allocating identifiers of other modules on the CAN, the user has to take care that no identifier is assigned twice!

Table 1.2.1: Default settings of the module in operation with the esd protocol



Overview

Default values in operation of the module with the CAL protocol	
Manufacturer name	ASCII 'esd_han'
Product name	ASCII 'LIO64'
Module ID	= setting of the coding switch
Module name	ASCII 'LIO64' + setting of the coding switches
CAN bitrate	= setting of jumper J220 (1 Mbits/s)
After a default RESET a <i>configuration download</i> to the module by means of the NMT protocol is absolutely necessary!	

The assignment of the COBs is not yet known (10.1997).

Table 1.2.2: Default adjustments of the module in operation with CAL



User parameter	Default setting [HEX]	Comment
<i>first Tx_activate_delay</i>	\$03E8 (1000 ms)	module transmits at the earliest 1 s after power-on or RESET on the bus
<i>outerr/vccmask</i>	\$FFFF	transmission of status message in case of rising and falling edge of a status bit
<i>error_hold_time</i>	\$0010 (16 ms)	error hold time
<i>input_change_mask_63-48</i>	\$FFFF	transmission of all input states, if at least one input changes it's state
<i>input_change_mask_47-32</i>	\$FFFF	
<i>input_change_mask_31-16</i>	\$FFFF	
<i>input_change_mask_15-0</i>	\$FFFF	
<i>vccdef_7</i>	\$8B53	switching level of the supply voltage monitor for each supply voltage is default $V_{MIN} = 17,8 \text{ V}$ and $V_{MAX} = 29,8 \text{ V}$
<i>vccdef_6</i>	\$8B53	
<i>vccdef_5</i>	\$8B53	
<i>vccdef_4</i>	\$8B53	
<i>vccdef_3</i>	\$8B53	
<i>vccdef_2</i>	\$8B53	
<i>vccdef_1</i>	\$8B53	
<i>vccdef_0</i>	\$8B53	

Table 1.2.3: Default settings of the user parameters of the module

The user parameters will be explained in detail in a special chapter.



2. Local Software Mode of Operation

2.1 General on Setting and Reading the Digital Outputs and Inputs

2.1.1 Identifier

When operating with the default parameters, the digital inputs and outputs of the CAN-Control-I/O module are read or set by means of the identifier set at the coding switches. The 2 LSB distinguish the various identifiers.

The identifiers can be determined as follows:

$$\text{identifier} = \$200 + [(\text{coding_switch_setting} - 1) \times 8]$$

For the setting of the coding switch only values between \$1 and \$F are permitted. (If the setting is \$0 when power is switched on, a default RESET is generated. The default RESET sets all parameters (even the parameters that are stored in the EEPROM) to the factory setting!)

Identifier bits					Identifier	Function
id11...id8	id7....id4	id3	id2	id1		
010.0	Setting of coding switch	0	0	0	RxId1	Setting the 32 outputs
010.0		0	0	1	TxId1	Transmitting the status of 64 inputs
-		0	1	0	TxId2	Transmitting the programmed value of 32 outputs
010.0		0	1	1	TxId3	Transmitting the status of the supply voltage of the output groups

Table 2.1.1: Assignment of the identifiers in default position



Local Software Mode of Operation

The following table shows the assignment of the identifiers for the coding switch settings 1, 2 and 3 as an example:

Coding switch setting (= module number)	Default identifier values [HEX]	Identifier
1	200 201 202 203	RxId1 TxId1 TxId2 TxId3
2	208 209 20A 20B	RxId1 TxId1 TxId2 TxId3
3	210 211 212 213	RxId1 TxId1 TxId2 TxId3

Table 2.1.2 Example for the default settings of the identifiers

The identifiers are freely programmable by means of the CAN. The programmed identifiers replace the set default identifiers.

The module transmits the level status of the 64 digital inputs on the Tx identifier TxId1. The status message of the output channels is transmitted on Tx identifier TxId2. The status message of the supply voltages for the output groups is transmitted on Tx identifier TxId3.

The outputs are set by means of Rx identifier RxId1.

2.1.2 Module-No.

The module-no. is used to identify the module during the initialization by the esd CAN protocol. In the default state of the module the value of the module-no. is set by the coding switch.

When the The module-no. that is used to identify the module during th configuration when it is configured



2.1.3 Count Mode of the Inputs and Outputs

The 64 inputs of the module are counted in this manual, similar to the hardware manual of the module, from 0 to 63 (I0...I63).

The 32 outputs are named Q0 to Q31.

The 32 outputs are divided into 8 groups. The outputs of each group are set by one driver circuit that has its own supply voltage connection pins. The output groups are named as follows:

Outputs	Output group
Q28 ... Q31	7
Q24 ... Q27	6
Q20 ... Q23	5
Q16 ... Q19	4
Q12 ... Q15	3
Q8 ... Q11	2
Q4 ... Q7	1
Q0 ... Q3	0

Table 2.1.3 Assignment of the outputs to the output groups



2.2 Functions of the Tx identifiers

2.2.1 Start the Transmission of a Tx Frame

The transmission of a Tx frame can be initiated as follows:

1. Cyclic

The transmission can be initiated cyclically, if the cycle time is set by the according parameter 'Tx-Activate-Time for TxId...'. This parameter can be set by use of the esd CAN protocol (refer to manual 'esd Protocol for CAN-Modules'). The factory setting of the cycle time for TxId3 is 10 s.

2. Remote Request

The transmission of a Tx frames can always be requested by a RTR frame.

3. Alternation of Tx Data

If the data of an identifier alternate, this can initiate the transmission of one or more identifiers, if this function is enabled.

The enabling is done by the user parameters, which are described in a separate chapter. With the factory settings of the user parameters this function is enabled for each Tx identifier.



2.2.2 Transmitting the Input Status (I0...I64) by means of TxId1

The module transmits the status of the 64 digital inputs by means of the eight-byte containing message of the identifier. The length of the message is always eight bytes.

If a 'high signal' applies to an input, the respective bit of the transmitted bytes is set to '1' (on). The same is valid for the 'low' level of an input.

TxId	Byte 1								Byte 2								Byte 3								Byte 4							
TxId1	Bit...								Bit...								Bit...								Bit...							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Input I...																															
	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32

TxId	Byte 5								Byte 6								Byte 7								Byte 8							
TxId1	Bit...								Bit...								Bit...								Bit...							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Input I...																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 2.2.1: Reading the inputs I0 to I64 by TxId1

The enabling of the transmission of the input status, if at least one input signal alternates it's state is done by the user parameters '*input_change_mask...*'.



2.2.3 Status Messages of Outputs by means of TxId2

The module can send status messages at Tx identifier TxId2. The data length is always 7 bytes.

The transmission can be started by an edge change of error signals of the output drivers or a change of the supply voltage level of the drivers (i.e. in common with TxId3). Furthermore a transmission of status messages in periodical intervals or by means of a remote request are possible.

The conditions for activating the transmission of the status message are transmitted by means of the user parameter '*outerr/vccmask*'.

TxId	Byte 1	Byte 2	Byte 3	Byte 4
TxId2	Bit...	Bit...	Bit...	Bit...
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
	Read back of the programmed desired values (prog. OUT) of output Q...			
	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
TxId	Byte 5	Byte 6	Byte 7	Byte 8
TxId2	errstat	vccerr	vccstat	not transmitted

Table 2.2.2: Status messages on TxId2

Explanation of the individual status messages:

prog. Out (programmed output value)

31...0... Byte 1 to 4 return the level of the outputs set by means of the CAN (desired value).
 '1' --> output active (on)
 '0' --> output not active (off)

errstat (error status)

Byte 5 returns the error status of the 8 output groups in binary code. If an error signal of a groups is active, the according bit is set to '1'.

Bits of parameter errstat	7	6	5	4	3	2	1	0
Errors signal of output groups	7	6	5	4	3	2	1	0

Table 2.2.3: Coding of the error signals of the output groups



`vccerr` and

`vccstat...` Byte 6 and 7 return the status of the output drivers' supply voltage in binary code. There is one bit for the VCC error and one bit for the VCC status for each output group. The assignment of the bits to the output groups is the same as used for the parameter `errstat` (refer table above).

The two bits from the parameters `vccerr` and `vccstat` that are assigned to one output group have together the following meaning:

Value of the parameter bits of one channel		Value of the supply voltage	Returned status
<code>vccerr</code>	<code>vccstat</code>		
0	0	$V_{\text{MIN}} < V_{\text{CC}} < V_{\text{MAX}}$	OK (V_{CC} is within the defined working range)
0	1	$V_{\text{ERR}} < V_{\text{CC}} < V_{\text{MIN}}$	V_{CC} is lower than the defined working range, but higher than V_{ERR}
1	0	$V_{\text{MAX}} < V_{\text{CC}}$	V_{CC} is too high
1	1	$V_{\text{CC}} < V_{\text{ERR}}$	V_{CC} is too low

Table 2.2.4: Coding of the supply voltages' status signals

Supply voltage monitor switching thresholds:

V_{ERR}	= 9,0 V	(fix, not programmable)
V_{MIN}	= 17,8 V	(default value, programmable by user parameter)
V_{MAX}	= 29,8 V	(default value, programmable by user parameter)



2.2.4 Transmission the Output Driver's Supply Voltage Values at TxId3

Via Tx identifier TxId3 the module can transmit the measured values of the output drivers supply voltages. Each value of the eight supply voltages transmitted in one byte:

TxId	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
TxId3	code_7	code_6	code_5	code_4	code_3	code_2	code_1	code_0

Table 2.2.5: Assignment of TxId3 with the values of the supply voltages

The measured voltage is determined by the following equation:

$$V_{CC} = 5,0V \cdot 11,0 \cdot \left(\frac{\text{code}}{256} \right)$$

The resolution of the measured values is 1 LSB, i.e. 0,2148 V.

The transmission of TxId3 can be initiated using different ways:

1. Cyclic

With the factory settings of the parameters the transmission of the voltage values is initiated cyclically, every 10 s. This time is set by the esd CAN protocol using the parameter 'Tx-Activate-Time for TxId3'.

2. Remote Request

The transmission can be requested by a RTR frame.

3. Alternation of the status bytes **vccerr** or **vccstat**

Only if the transmission of TxId3 (and TxId2) is enabled by the user parameter '**outerr/vccstat**', a transmission is initiated if the status bytes alternate. With the default setting (factory setting) of this user parameter the transmission is enabled for each group.



2.3 Setting the Outputs by RxId1

The CAN-Control-I/O module receives the data for setting the outputs on Rx identifier RxId1.

Always 4 bytes has to be transmitted. To activate the desired output, the according bit has to be set to '1'.

RxId	Byte 1								Byte 2								Byte 3								Byte 4							
RxId1	Bit...								Bit...								Bit...								Bit...							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Output Q...																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 2.3.1: Setting the outputs Q0 to Q31

The status of the outputs after a RESET is '0' in operation with the default parameters.



3. User Parameter

3.1 Overview of the User Parameters

If the module is run by the esd protocol, the user parameters are transmitted by means of the command 'setting the user parameters' (\$86) on byte 5 and 6 of the INIT-Id (\$700).

All user parameters have always to be transmitted as 16-bit value with byte 5 as MSB!

The sequence for transmitting and receiving the user parameters is described in detail in an other manual called 'esd Protocol for CAN Modules'

In this manual only the module-specific user parameters are described.

The variables transmitted by means of the user parameters partly become active instantly or only after the transmission of a 'save config'-command to the module and a following RESET.

If the CMS protocol is implemented, the user parameters are set by a configuration download (NMT). The format of the configuration-download file has not been known at the time this manual went into print (10.1997).



User parameter

Following table gives an overview of the user parameters of the module:

User parameter No.	Parameter	Value range	Default settings
\$00	<i>first Tx_activate_delay</i>	\$0000. . . \$FFFF (0...65535 ms)	\$03E8 (1000 ms)
\$01	reserved	-	-
\$02	<i>outerr/vccmask</i>	\$0000. . . \$FFFF	\$FFFF
\$03	<i>error_hold_time</i>	\$0001. . . \$00FF	\$0010
\$04	<i>input_change_mask_63- 48</i>	\$0000. . . \$FFFF	\$FFFF
\$05	<i>input_change_mask_47- 32</i>	\$0000. . . \$FFFF	\$FFFF
\$06	<i>input_change_mask_31- 16</i>	\$0000. . . \$FFFF	\$FFFF
\$07	<i>input_change_mask_15- 0</i>	\$0000. . . \$FFFF	\$FFFF
\$08	<i>vccdef_7</i>	\$0000. . . \$FFFF	\$8B53
\$09	<i>vccdef_6</i>	\$0000. . . \$FFFF	\$8B53
\$0A	<i>vccdef_5</i>	\$0000. . . \$FFFF	\$8B53
\$0B	<i>vccdef_4</i>	\$0000. . . \$FFFF	\$8B53
\$0C	<i>vccdef_3</i>	\$0000. . . \$FFFF	\$8B53
\$0D	<i>vccdef_2</i>	\$0000. . . \$FFFF	\$8B53
\$0E	<i>vccdef_1</i>	\$0000. . . \$FFFF	\$8B53
\$0F	<i>vccdef_0</i>	\$0000. . . \$FFFF	\$8B53

Table 3.1.1: User parameter of the module

On the following pages the individual user parameters of the module will be explained in detail.



3.2 Description of the User Parameters

3.2.1 First Tx-Activate Delay (Parameter 0)

Parameter 0 transmits the delay time which is to pass before the module initiates the transmission of Tx frames after a RESET. The delay time is to secure that all modules operate rigidly on the bus before the module starts transmitting.

The changed parameter becomes only active after a 'save config'-command (refer to esd protocol for CAN modules) with a following RESET.

User parameter no. (=sub command no.)	Parameter	Value range	Default setting
\$00	<i>first_Tx_activate_delay</i>	\$0000...\$FFFF 0...65535 ms	\$2710 10 s

Table 3.2.1: User parameter 0



User parameter

3.2.2 Mask for Output Error and Status Message (Parameter 2)

By means of parameter 2 for each output group of the CAN-Control-I/O module it can be chose, if a transmission shall be initiated, when the status bytes alternates (L-<H edge or H-<L edge of at least one parameter's bit).

User parameter no. (=sub command no.)	Parameter	Value range	Default setting
\$02	<i>outerr/vccmask</i>	\$0000. . . \$FFFF	\$FFFF

Table 3.2.2: User parameter 2

For the evaluation of this user parameter the two bytes are considered separately in the format \$xx.yy. The first byte 'xx' is used for masking the general error status of the group (**errstat**) and the second byte 'yy' is used for masking the VCC status of the groups (**vccerr/vccstat**):

Bits of user parameter <i>outerr/vccmask</i> ->	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
enables an error message, if the following error byte alternates ->	errstat								vccerr/vccstat							
...in the output group ->	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Table 3.2.3: Assignment of the output groups to the bits of the user parameter 2

Comments to the bits 15...8 (Byte 'xx'):

Is one of these bits set to '1', this has effects to the initiation of transmission of frames and to the LED display in the front panel:

1. An alternation of the common error status **errstat** of the according group initiates a transmission of an error message via TxId2.
2. An error in the according output group activates the signal 'Err-LED' (error display is activated).

Comments to the bits 7...0 (Byte 'yy'):

Is one of these bits set to '1', this has effects to the initiation of transmission of frames and to the LED display in the front panel, too:

1. An alternation of the supply voltage monitor's status **vccerr/vccstat** of the according output group initiates a transmission of an error message via TxId2 and a transmission of the voltage values via TxId3.
2. Is the signal **vccerr** of the according output group active, the signal 'VCC-OK' is deactivated (VCC LED is 'off' or 'flashing').



3.2.3 Setting of the Error-Hold-Time (Parameter 3)

By means of parameter 3 the Error-Hold-Time is set for the common error message (**errstat**) for all output groups together.

User parameter no. (=sub command no.)	Parameter	Value range	Default setting
\$03	<i>error_hold_time</i>	\$0000...\$FFFF	\$0010 (16 ms)

Table 3.2.4: User parameter 3

The Error-Hold-Time is used for holding the error signal active. This is necessary, because the error signal is set inactive together with the output signal by the automatically internal protection circuit of the driver. The output is set active again automatically after a wait time T_{OFF} , even if the reason for the error is still present. This would cause an alternating error output signal.

The following figure shows a principle timing example:

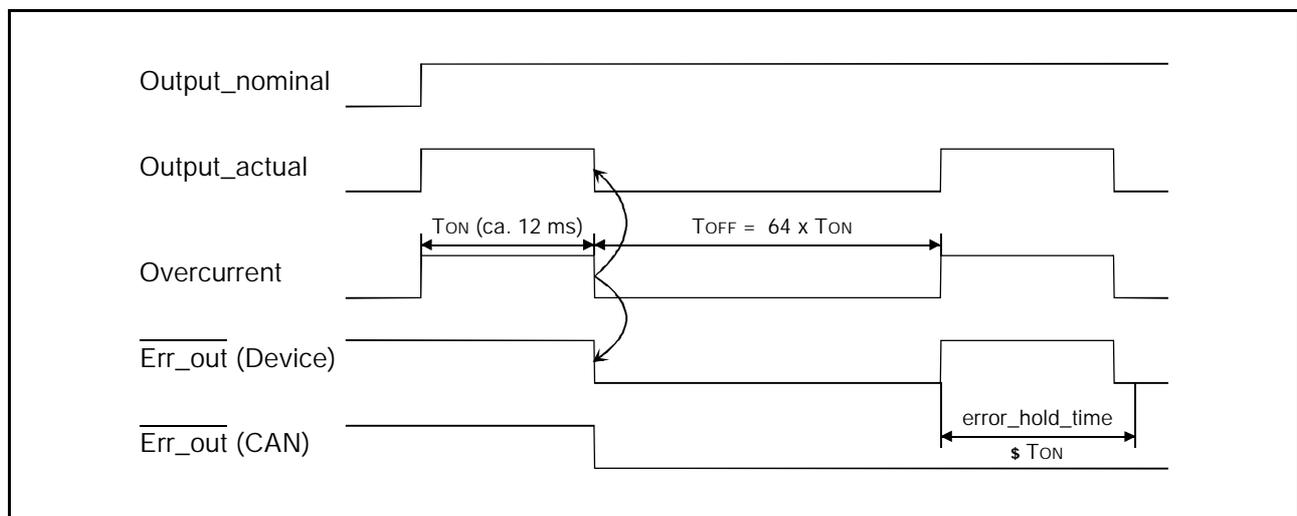


Fig. 3.2.5: Timing example of the error signal

After the first error signal is activated, the Error-Hold-Time shall prevent the deactivation of the following error signal for the duration of at least T_{ON} . Therefore the value of **error_hold_time** has always be set to values greater than T_{ON} .



User parameter

3.2.4 Input Masks (Parameters 4...7)

The parameters 4 to 7 define, which input shall be determined for the initiation of the transmission of the status of all inputs via TxId1. At least one input has to be selected. Both, the rising and the falling edges of an input signal can initiate a transmission (H-<L edge or L-<H edge).

If the parameter bit is set to '1', the module transmits the data to the CAN at an edge change of the according input.

User parameter no. (=sub command no.)	Parameter	Value range	Default setting
\$04	<i>input_change_mask_63-48</i>	\$0000. . . \$FFFF	\$FFFF
\$05	<i>input_change_mask_47-32</i>	\$0000. . . \$FFFF	\$FFFF
\$06	<i>input_change_mask_31-16</i>	\$0000. . . \$FFFF	\$FFFF
\$07	<i>input_change_mask_15-0</i>	\$0000. . . \$FFFF	\$FFFF

Table 3.2.6: User parameters 4...7

Bits of the user parameter <i>input_change_mask_63-48</i> ->	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mask input I...->	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bits of the user parameter <i>input_change_mask_47-32</i> ->	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mask input I...->	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bits of the user parameter <i>input_change_mask_31-16</i> ->	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mask input I...->	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bits of the user parameter <i>input_change_mask_15-0</i> ->	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mask input I...->	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 3.2.7: Assignment of the user parameter bits to the inputs



3.2.5 Definition of the Voltage Thresholds (Parameters 8...15)

By means of the parameters 8 to 15 the voltage thresholds of the output driver's voltage monitor can be defined. There is one user parameter for each output group.

User parameter no. (=sub command no.)	Parameter	Value range	Default setting
\$08	<i>vccdef_7</i>	\$0000. . . \$FFFF	\$8B53
\$09	<i>vccdef_6</i>	\$0000. . . \$FFFF	\$8B53
\$0A	<i>vccdef_5</i>	\$0000. . . \$FFFF	\$8B53
\$0B	<i>vccdef_4</i>	\$0000. . . \$FFFF	\$8B53
\$0C	<i>vccdef_3</i>	\$0000. . . \$FFFF	\$8B53
\$0D	<i>vccdef_2</i>	\$0000. . . \$FFFF	\$8B53
\$0E	<i>vccdef_1</i>	\$0000. . . \$FFFF	\$8B53
\$0F	<i>vccdef_0</i>	\$0000. . . \$FFFF	\$8B53

Table 3.2.8: User parameter 8...15



User parameter

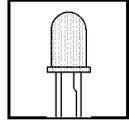
Bits of user parameter ->	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Output group for user parameter <i>vccdef_7</i> ->	V _{MAX} Output group 7								V _{MIN} Output group 7							
Output group for user parameter <i>vccdef_6</i> ->	V _{MAX} Output group 6								V _{MIN} Output group 6							
Output group for user parameter <i>vccdef_5</i> ->	V _{MAX} Output group 5								V _{MIN} Output group 5							
Output group for user parameter <i>vccdef_4</i> ->	V _{MAX} Output group 4								V _{MIN} Output group 4							
Output group for user parameter <i>vccdef_3</i> ->	V _{MAX} Output group 3								V _{MIN} Output group 3							
Output group for user parameter <i>vccdef_2</i> ->	V _{MAX} Output group 2								V _{MIN} Output group 2							
Output group for user parameter <i>vccdef_1</i> ->	V _{MAX} Output group 1								V _{MIN} Output group 1							
Output group for user parameter <i>vccdef_0</i> ->	V _{MAX} Output group 0								V _{MIN} Output group 0							

Table 3.2.9: Assignment of the user parameters 4...7 to the output groups

The value that must be set for a desired voltage threshold (V_{MAX} and V_{MIN}) has is determined as follows:

$$\text{code} = \left(\frac{V_{\text{MAX/MIN}} \cdot 256}{5,0\text{V} \cdot 11,0} \right)$$

with
code... value, that must be transmitted
V_{MIN/MAX}... desired voltage threshold



4. LED Display

4.1 Error LED

The error LED lights, if an error condition is active for at least one output and the error message is enabled.

The error condition is active, if an output driver has activated his error signal. The error message can be enabled by the user parameter **outerr/vccmask**.

The error message can also be transmitted in the byte **errstat** via TxId2.

4.2 VCC OK LED

Basic condition for the illumination of the VCC OK LED is the enabling by the user parameter **outerr/vccmask**. If an output group is not enabled, the status of the supply voltage is not considered for further evaluation.

The status of a supply voltage is 'OK' if the voltage level is between V_{MIN} and V_{MAX} .

The status of the supply voltages can also be transmitted in the bytes **vccerr** and **vccstat** via TxId2.

Luminous status	Meaning of the luminous status
Constantly OFF	None of the supply voltages is OK.
Flashing	The level of at least one supply voltage is OK and the level of at least one other supply voltage is not OK. (Evaluation only of groups, that are enabled.)
Constantly ON	The level of every enabled supply voltage is OK.

Table 4.2.1: Luminous states of the VCC OK LED

CAN-Control-I/O

(LASCON-I/O)

DS-301, DS-401
CANopen Implementation

Software-Manual

to product: C.2071.xx and K.3784.04

Manual File:	I:\texte\Doku\MANUALS\CAN\CAN-Control\IO\can-control-io_canopen_11s.en9
Manual Order no.:	C.2071.21
Date of Print:	18.04.2005

Described Firmware Version:	V1.SBJ/CanOpen
------------------------------------	----------------

Changes in the Software and/or Documentation

Alterations in this manual versus previous version	Alterations in software	Alterations in documentation
Node-ID setting changed/inserted.	x	x
-	-	-

Technical details are subject to change without notice.

NOTE

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. 207

30165 Hannover

Germany

Phone: +49-511-372 98-0

Fax: +49-511-372 98-68

E-mail: info@esd-electronics.com

Internet: www.esd-electronics.com

USA / Canada:

esd electronics Inc.

12 Elm Street

Hatfield, MA 01038-0048

USA

Phone: +1-800-732-8006

Fax: +1-800-732-8093

E-mail: us-sales@esd-electronics.com

Internet: www.esd-electronics.us

Content	Page
1. Introduction	3
1.1 About this Document	3
1.2 Reference	3
1.3 Common Terms and Abbreviations	3
1.4 Node-ID Setting via Coding Switch	4
2. Communication Profile Area	5
2.1 Terms and Abbreviations	5
2.2 Overview of Communication Parameters	6
2.3 Description of Parameters	7
2.3.1 Device Type 1000 _h	7
2.3.2 Error Register 1001 _h	8
2.3.3 Manufacturer Status Register 1002 _h	9
2.3.4 Pre-defined Error Field 1003 _h	10
2.3.5 Manufacturer's Device Name 1008 _h	12
2.3.6 Manufacturer's Hardware Version 1009 _h	13
2.3.7 Manufacturer's Software Version 100A _h	14
2.3.8 Store Parameters 1010 _h	15
2.3.9 Restore Default Parameters 1011 _h	16
2.3.10 COB_ID Emergency Object (1014 _h)	17
2.3.11 Heartbeat Protocol	18
2.3.11.1 Consumer Heartbeat Time 1016 _h	19
2.3.11.2 Producer Heartbeat Time 1017 _h	20
2.3.12 Identity Object 1018 _h	21
2.3.13 Verify Configuration 1020 _h	23
2.3.14 Error Behaviour Object 1029 _h	24
2.3.15 Firmware Date 1F52 _h	25
2.3.16 Object Receive PDO Communication Parameter 1400 _h	26
2.3.17 Receive PDO Mapping Parameter 1600 _h	27
2.3.18 Object Transmit PDO Communication Parameter 1800 _h ...1802 _h	28
2.3.19 Transmit PDO Mapping Parameter 1A00 _h ...1A02 _h	29
2.4 Transmission Types Supported in Accordance with DS-301	30
2.5 Table of Important Identifier and Messages of CANopen	31
2.6 PDO-Mapping Summary	32
2.6.1 Rx-PDO1: Setting the outputs Q0 to Q31	32
2.6.2 Tx-PDO1: Reading the Inputs I0 to I64	33
2.6.3 Tx-PDO2: Status Messages	33
2.6.4 Tx-PDO3: Values of the Supply Voltages	33
2.7 Device Profile Area	34
2.7.1 Implemented Objects 6000 _h -6400 _h	34
2.7.2 Read Input 8-Bit (6000 _h)	35
2.7.3 Global Interrupt Enable Digital (6005 _h)	36
2.7.4 Interrupt Mask Any Change 8-Bit (6006 _h)	37
2.7.5 Write Output 8-Bit (6200 _h)	38
2.7.6 Error Mode Output 8-Bit (6206 _h)	39
2.7.7 Error Value Output 8-Bit (6207 _h)	40

2.7.8 Read Analog Input 8-Bit (6400 _h)	41
2.8 Manufacturer Specific Profile Area	42
2.8.1 Implemented Objects 2800 _h -2804 _h	42
2.8.2 Output Driver Supply Voltage Threshold Low (2800 _h)	43
2.8.3 Output Driver Supply Voltage Threshold High (2801 _h)	44
2.8.4 Output Driver Supply Voltage Status (2802 _h)	45
2.8.5 Output Driver Status (2803 _h)	46
2.8.6 Output Driver Error Mask and Time (2804 _h)	48

1. Introduction

1.1 About this Document

This document describes the DS-301 and DS-302 objects that are supported by the CAN-Control-I/O and the LASCON-I/O module. Additionally the manufacturer specific objects that are implemented by esd are described.

1.2 Reference

- [1] CAN-Control-I/O (LASCON I/O) Hardware Manual, Rev. 1.2 (31.03.1999)
- [2] CiA DS-301 CANopen Application Layer and Communication Profile, V4.02 (02.2002)
- [3] CiA DS-401 CANopen Device Profile for Generic I/O Modules, V2.1 (05.2002)

1.3 Common Terms and Abbreviations

n.a.	not applicable
rw	read and write access supported
ro	read only access supported
hex, ... _h	hexadecimal value
dec	decimal value

1.4 Node-ID Setting via Coding Switch

The coding switch defines the CANopen Node-ID of the module. The Node-ID is determined as follows:

$$\text{Node-ID} = (\text{CodingSwitchSetting AND } C_n) \cdot 8 + (\text{CodingSwitchSetting AND } 3) + 4$$

The following table shows the resulting CANopen Node-IDs:

Coding switch setting	CANopen Node-ID
0	4
1	5
2	6
3	7
4	36
5	37
6	38
7	39
8	68
9	69
A	70
B	71
C	100
D	101
E	102
F	103

2. Communication Profile Area

2.1 Terms and Abbreviations

The following terms will be used in the tables representing the communication parameters:

PDO-Mapping	PDO-mapping is available for this sub-index of the PDO
Access Mode	permissible access modes to this parameter
	ro... read_only This parameter can only be read. Write accesses cause an error message.
	const.... constant This parameter cannot be changed by the user. It can only be read. Write accesses cause an error message.
	rw... read & write This parameter can be read or set.
Value Range	value range of the parameter
Default value	default setting of parameter when module leaves the manufacturer
Name/Description	name and short description of parameter

2.2 Overview of Communication Parameters

The format of the communication parameters can be taken from CiA DS-301. Special features of implementing the CAN-Control-I/O will be described in the following chapters of this manual.

The module only supports the communication parameters listed in the table below.

Index [Hex]	Name	Sub-Index	Type	Access	Default
1000	Device Type	-	Unsigned32	ro	00070191 _h
1001	Error Register	-	Unsigned8	rw	Error-Code
1002	Manufacturer Status Register	-	Unsigned32	ro	n.a.
1003	Pre-Defined Error Field	16	Unsigned32	rw	n.a.
1004	Number of sync/async PDOs	not supported			
1008	Manufacturer's Device Name 1*)	-	Visible String 1*)	ro	Can-Control-IO-64/32
1009	Manufacturer's Hardware Version	-	Visible String	ro	2*)
100A	Manufacturer's Software Version 1*)	-	Visible String 2*)	ro	2*)
100C	Guard Time	not supported			
100D	Life Time Factor	not supported			
1010	Store Parameter	0, 1	Unsigned32	rw	n.a.
1011	Restore Parameter	0, 1	Unsigned32	rw	n.a.
1014	COB-ID Emergency message	-	Unsigned32	rw	80 _h + Node-ID
1016	Consumer Heartbeat Time	0,1 ... 16	Unsigned32	rw	n.a.
1017	Producer Heartbeat Time	0	Unsigned16	rw	0 ms
1018	Identity Object	0,1 ... 4	Unsigned32	ro	see page 21
1020	Verify Configuration	0, 1, 2	Unsigned32	ro	n.a.
1029	Error Behaviour	0, 1	Unsigned8	rw	n.a.
1F52	Firmware Date	0, 1, 2	Unsigned32	ro	n.a.
1400	Receive PDO Communication Parameter	0 ... 3	PDOPar	ro	ref. chapter 2.3
1600	Receive PDO Mapping Parameter	0 ... 4	PDO Mapping	ro	ref. chapter 2.3
1800 - 1802	Transmit PDO Communication Parameter	0 ... 5	PDOPar	ro	ref. chapter 2.3
1A00 - 1A02	Transmit PDO Mapping Parameter	0 ... 8	PDO Mapping	ro	ref. chapter 2.3

ro - Read Only, rw - Read/Write

1*) Attention: length > 4 bytes, i.e. upload more complicated than with other parameters

2*) depends on rev. level of hard- and software

2.3 Description of Parameters

2.3.1 Device Type 1000_h

INDEX	1000_h
Name	<i>device type</i>
Data Type	unsigned 32
Default Value	00070191 _h

The value of *device type* is: 0007.0191_h (digital I/O: 0007_h
device profile number: 0191_h)

Example: Reading-out the Device Type

The CANopen master transmits the read request to the CAN-Control-I/O-module with module number 3 (Node-ID = 3_h) by means of identifier '603_h':

ID	RTR	LEN	DATA								
			1	2	3	4	5	6	7	8	
603 _h	0 _h	8 _h	40 _h Read Request	00 _h	10 _h Index=1000 _h	00 _h	00 _h Sub Index	00 _h	00 _h	00 _h	00 _h

The CAN-Control-I/O-module with Node-ID 3 responds to the master by means of the read response via identifier '583_h' (580 + Node-ID) the value of the device type:

ID	RTR	LEN	DATA								
			1	2	3	4	5	6	7	8	
583 _h	0 _h	8 _h	43 _h Read Response	00 _h	10 _h Index=1000 _h	00 _h	91 _h Sub Index	01 _h Dig. Profile No.191	07 _h	00 _h Digital I/O	00 _h

value of the device type: 0007.0191_h

The data field is always structured 'LSB first, MSB last'.

2.3.2 Error Register 1001_h

The CAN-Control-I/O-module uses the error register in order to show error messages.

INDEX	1001_h
Name	<i>error register</i>
Data Type	unsigned 8
Default Value	No

At the moment the following bits are supported:

Bit	Meaning
0	generic
1	-
2	-
3	-
4	communication
5	-
6	-
7	manufacturer-specific error

Unsupported bits return the value '0'.

Error Bit	Error
0	no error
1	error

The following error messages are implemented:

- 00_h - no error
- 81_h - any manufacturer-specific error has appeared
- 11_h - communication error has appeared

2.3.3 Manufacturer Status Register 1002_h

INDEX	1002_h
Name	<i>manufacturer status register</i>
Data Type	unsigned32
Default Value	n.a.

The bits of the status register are assigned as represented below:

Register-Bit (Assembler)	Description	Level Assignment	
D31 ... D8	This bits are reserved for future use.	0	reserved, may be returned as '0' or '1'
		1	
D7	<i>New on Bus</i>	0	reserved, may be returned as '0' or '1'
		1	
D6	<i>Default wake up</i>	0	normal start
		1	module is started with default parameters
D5	<i>I²C busy</i>	0	no access to EEPROM
		1	local SW accesses I ² C-EEPROM
D4	<i>guard_error</i>	0	no error
		1	<i>guard_error</i> (heartbeat-consumer-error)
D3	<i>I²C-Error</i>	0	no I ² C-error
		1	I ² C-error
D2	<i>Error on CAN</i>	0	no CAN-bus error detected
		1	CAN-bus error
D1	<i>Suspend-Bit</i>	0	module is in state 'operational'
		1	module is in state 'preoperational' or 'stopped'
D0	<i>Power-Up-Reset</i>	0	last reset is not generated by power-up
		1	last reset is generated by power-up

Notes:

- Bits D31...D24 are always read as '01_h'.
- Bits D23...D16 return the actual setting at jumper J220 (ref. hardware manual [1]). This setting is determined for the hardware bit-rate setting after restart.
- Bits D15...D8 return the actual setting of the codiung switch SW220 (ref. hardware manual [1]). This setting is determined for the hardware Node-ID setting after restart.

2.3.4 Pre-defined Error Field 1003_h

INDEX	1003_h
Name	<i>pre-defined error field</i>
Data Type	unsigned 32
Default Value	No

In the *pre-defined error field* a list of the errors which occurred last is stored. Sub-index 0 contains the current number of errors stored in the list. Under sub-index 1 the last error which occurred is stored. If a new error occurs, the previous error is stored under sub-index 2 and the new error under sub-index 1, etc. In this way a list of the error history is created.

The error buffer is structured like a ring buffer. If it is full, the oldest entry is deleted for the latest entry.

This module supports a maximum of 16 error entries. When the 17th error occurs the oldest error entry is deleted. In order to delete the entire error list, sub-index '0' has to be set to '0'. This is the only permissible write access to the object.

With every new entry to the list the module transmits an **Emergency Frame** to report the error.

Index [Hex]	Sub-index [Dec]	Description	Value range [Hex]	Default [Dec]	Data type	R/W
1003	0	<i>no_of_errors_in_list</i>	0, 1...40	-	unsigned 8	rw
	1	<i>error-code n</i>	0...FFFFFFFF	-	unsigned 32	ro
	2	<i>error-code (n-1)</i>	0...FFFFFFFF	-	unsigned 32	ro
	:	:	:	:	:	ro
	16	<i>error-code (n-15)</i>	0...FFFFFFFF	-	unsigned 32	ro

Parameter Description:

- no_of_errors_in_list* - contains the number of error codes currently on the list
 - n* = number of error which occurred last
 - in order to delete the error list this parameter has to be set '0'
 - if *no_of_errors_in_list* ≠ 0, the error register (Object 1001_h) is set

error-code x The 32-bit long error code is combined from the CANopen-Emergency-Error-Code described in DS-301, Table 21 and the ‘manufacturer-specific error field’.

Bit:	31 16	15 0
Contents:	<i>manufacturer-specific error field</i>	<i>emergency-error-code</i>

manufacturer-specific error field: always ‘00’ at CAN-Control-I/O

emergency-error-code: only the following error codes are being supported:
 0000_h - Error-Reset or no Error
 8120_h - CAN-BUS-Off (CAN in Error Passive Mode)
 8130_h - Heartbeat-Error
 8140_h - recovered from CAN-BUS-OFF

Emergency Frame

The data of the emergency frame transmitted by the CAN-Control-I/O is structured as follows:

Byte:	0	1	2	3	4	5	6	7
Contents:	<i>emergency-error-code</i> (see above)	<i>error-register</i> 1001 _h	manufacturer specific error field (not supported)					

2.3.5 Manufacturer's Device Name 1008_h

INDEX	1008_h
Name	<i>manufacturer's device name</i>
Data Type	visible string
Default Value	string: 'Can-Control-IO-64/32'

2.3.6 Manufacturer's Hardware Version 1009_h

INDEX	1009_h
Name	<i>manufacturer's hardware version</i>
Data Type	visible string
Default Value	0

This parameter returns a one byte ASCII-code.

2.3.7 Manufacturer's Software Version 100A_h

INDEX	100A_h
Name	<i>manufacturer's software version</i>
Data Type	visible string
Default Value	string: e.g: 'V1.SBJ/CANopen'

Reading the software version is similar to reading the manufacturer device name via the domain upload protocol. Please refer to CiA DS-202-2 (CMS-Protocol Specification) for a detailed description of the upload.

2.3.8 Store Parameters 1010_h

This command stores the parameter in the EPROM. Only the command ‘Save *all* Parameters’ is supported.

With the write access the below shown byte order has to be send.

A read access returns information about the implemented store function. In this case a read access to the object 1010_h/subindex 1 always returns the value 00000001_h, that indicates ‘save all parameters’ (for further information see CiA DS-301).

INDEX	1010_h
Name	<i>store parameters</i>
Data Type	unsigned 32

Index [Hex]	Sub-index [Dec]	Description	Value range [Hex]	Data type	R/W
1010	0	<i>number_of_entries</i>	1 _h	unsigned 8	ro
	1	<i>save_all_parameters</i>	no default, write: 65 76 61 73 h (= ASCII: 'e' 'v' 'a' 's')	unsigned 32	rw

Parameters which can be stored or loaded

Communication parameters:

1016_h *Consumer Heartbeat Time*

1017_h *Producer Heartbeat Time*

1020_h Verify Configuration: *Configuration_Date, Configuration_Time*

1029_h Error Behaviour Object: *Communication_Error*

1400_h only COB-ID

180x_h only COB-ID, Transmission Time and Timer values

Application parameters:

6005_h

6006_h

6200_h

6206_h

Manufacturer-specific parameters:

2800_h

2801_h

2804_h

2.3.9 Restore Default Parameters 1011_h

Via this command the default parameters, valid when leaving the manufacturer, are activated again. Every individual setting stored in the EEPROM will be lost. Only command ‘Restore all Parameters’ is being supported.

When writing the index the byte sequence represented below has to be transmitted. Reading the index returns information about the implemented restore function (for details please refer to CiA DS-301).

INDEX	1011_h
Name	<i>restore default parameters</i>
Data Type	unsigned 32

Index [Hex]	Sub-index [Dec]	Description	Value range [Hex]	Data type	R/W
1011	0	<i>number_of_entries</i>	3	unsigned 8	ro
	1	<i>load_all_default_parameters</i>	no default, write: 64 61 6F 6C _h (= ASCII: 'd' 'a' 'o' 'l')	unsigned 32	rw

The communication parameters and the application parameters which can be stored or loaded are listed in object 1010_h (see page 15).

2.3.10 COB_ID Emergency Object (1014_h)

INDEX	1014_h
Name	<i>COB-ID emergency object</i>
Data type	unsigned 32
Default value	80 _h + Node-ID

This object defines the COB-ID of the Emergency Object (EMCY).

The structure of this object is shown in the following table:

Bit-No.	Value	Meaning
31 (MSB)	0/1	0: EMCY exists / is valid 1: EMCY does not exist / EMCY is not valid
30	0	reserved (always 0)
29	0	always 0 (11-bit ID)
28...11	0	always 0 (29-bit IDs are not supported)
10...0 (LSB)	x	bits 0...10 of COB-ID

The identifier can take values between 0...7FF_h.

2.3.11 Heartbeat Protocol

The heartbeat protocol defines an error control service without need for remote frames.

A heartbeat *producer* transmits a heartbeat message cyclically. One or more heartbeat *consumer* receive the indication. The relationship between *producer* and *consumer* is configurable via the object dictionary.

The heartbeat *consumer* guards the reception of the heartbeat within the heartbeat consumer time. The guarding of the heartbeat *consumer* starts after the reception of the first heartbeat-CAN frame with $\text{data}(0) \neq '0'$.

If the heartbeat is not received within the heartbeat consumer time a heartbeat event will be generated. At the CAN-Control-I/O module the heartbeat event generates a heartbeat error.

If the heartbeat *producer* time is set to a value unequal '0' (object 1017_h) the heartbeat *producer* at that device starts immediately to send heartbeat frames.

If a device starts, that has already set the value of the heartbeat producer time unequal '0' in the EEPROM, the heartbeat protocol starts on the state transition from INITIALISING to PRE-OPERATIONAL.

2.3.11.1 Consumer Heartbeat Time 1016_h

INDEX	1016_h
Name	<i>consumer heartbeat time</i>
Data Type	unsigned 32
Default Value	0

The CAN-Control-I/O module supports the monitoring of 1 heartbeat producer only.

Index [Hex]	Sub-index [Dec]	Description	Value range [Hex]	Default [Dec]	Data type	R/W
1016	0	<i>number_of_entries</i>	0,1	1	unsigned 8	ro
	1	<i>consumer-heartbeat_time</i>	0... 00 7F FF FF _h	0	unsigned 32	rw

Description of the parameter *consumer-heartbeat_time*:

<i>consumer-heartbeat_time</i>			
Bit	3124	2316	150
Description	reserved (always '0')	<i>Node-ID</i> (unsigned 8)	<i>heartbeat_time</i> (unsigned 16)

Node-ID Node-Id of the heartbeat producer module that has to be monitored.

heartbeat_time If the heartbeat producer does not send a message at the Node Guarding-ID within the time *heartbeat_time*, a heartbeat event will be generated.

The consumer *heartbeat_time* has to be higher than the corresponding producer *heartbeat_time* configured on the device producing this heartbeat.

2.3.11.2 Producer Heartbeat Time 1017_h

INDEX	1017_h
Name	<i>producer heartbeat time</i>
Data Type	unsigned 16
Default Value	0 ms

Here you specify the time with which the CAN-Control-I/O-module cyclically transmits a heartbeat frame on the node-guarding ID.

If a value larger than zero is set for producer-heartbeat time, it is active. If the producer-heartbeat time is set '0', the cyclic transmission of the heartbeat by this module is terminated.

Index [Hex]	Sub-index [Dec]	Description	Value range [Hex]	Default [Dec]	Data type	R/W
1017	0	<i>producer-heartbeat_time</i>	0...FFFF	0 ms	unsigned 16	rw

producer-heartbeat_time Cycle period of heartbeat producer to transmit the heartbeat on the node-guarding ID.
 The consumer-heartbeat time of the monitoring module must always be larger than the producer-heartbeat time of the heartbeat-transmitting module.

2.3.12 Identity Object 1018_h

INDEX	1018_h
Name	<i>identity object</i>
Data Type	unsigned 32
Default Value	No

The identity object contains general information about the CAN-module.

Index [Hex]	Sub-index [Dec]	Description	Value range [Hex]	Default [Hex]	Data type	R/W
1018	0	<i>no_of_entries</i>	1	1	unsigned 8	ro
	1	<i>vendor_id</i>	0...FFFFFFFF	0000 0017	unsigned 32	ro
	2	<i>product_code</i>	0...FFFFFFFF	1378 4040	unsigned 32	ro
	3	<i>revision_number</i>	0...FFFFFFFF	'current SW-version'	unsigned 32	ro
	4	<i>serial_number</i>	0...FFFFFFFF	'serial_number'	unsigned 32	ro

Parameter Description:

vendor_id This parameter returns the esd-vendor ID. The value of the vendor ID is constant 00 00 00 17_h.

product_code This parameter returns the esd-order number of the module.
Example:
In the value '13 78 40 40_h' the order number 'K.3784.04' is coded.

revision_no This parameter returns the software version. The upper two bytes return the revision numbers of the major changes according to DS-301 and the lower two bytes return the revision number of minor changes. The software version contains of the following sections:

- Kernel Software (described by the parameters '*lev*' (level) and '*rev*' (revision))
- Application Software (described by the parameter '*ext*' (extension))
- Protocol Software (described by the parameter '*plev*' (protocol level))

The returned software version is coded for 'major' and for 'minor'-revision number as follows

revision_no = xxyy

and is determined by the number *s* of the ASCII-code of the characters

with $xx = (lev - '0') \cdot 26 + (rev - 'A')$

$yy = (plev - 'H') \cdot 16 + (ext - 'A')$

Example:

If the actual software has the revision:

Kernel: $lev = '1'$ $rev = 'M'$

Application: $ext = 'A'$

Protocol: $plev = 'H'$

the returned value for *revision_no* will be 26 00 00 00_h.

serial_no

This parameter returns the serial number code of the PCB-hardware.

The higher two bytes of *serial_no* contain the letters that code the production lot. They return the ASCII-code of the letters with the most significant bit set to '1', to distinguish letters from numbers:

$(\text{ASCII-code}) + 80_{\text{h}} = \text{returned bytes}$

The following numbers code the number of each module as BCD-value.

Example:

If the value 'C1 C1 12 34_h' is returned, this will mean the hardware serial number code 'AA 1234'. This value has to match with the value labelled at the module.

2.3.13 Verify Configuration 1020_h

INDEX	1020_h
Name	<i>verify configuration</i>
Data Type	unsigned 32

In this object date and time of the last configuration can be stored in order to be able to check at a later time, whether the configuration stored corresponds to the one expected.

Index [Hex]	Sub-index [Dec]	Description	Value range [Hex]	Default [Dec]	Data type	R/W
1020	0	<i>no_of_entries</i>	2	2	unsigned 8	ro
	1	<i>configuration_date</i>	0...FFFFFFFF	0	unsigned 32	rw
	2	<i>configuration_time</i>	0...FFFFFFFF	0	unsigned 32	rw

Parameter Description:

configuration_date Date of last configuration of module, represented in days since the 01.01.1984.

configuration_time Time in ms since midnight of the day of the last configuration.

2.3.14 Error Behaviour Object 1029_h

INDEX	1029_h
Name	<i>error behaviour object</i>
Data Type	unsigned 8
Default Value	No

If an error event occurs (such as a heartbeat error), the module changes into the status which has been defined in parameter *communication_error*.

Index [Hex]	Sub-index [Dec]	Description	Value range [Hex]	Default [Dec]	Data type	R/W
1029	0	<i>no_of_error_classes</i>	1	1	unsigned 8	ro
	1	<i>communication_error</i>	0, 1, 2	0	unsigned 8	rw

Parameter Description:

no_of_error_classes Number of error classes (here always '1')

communication_error 0 - pre-operational (only, if current status = operational)
 1 - no state change
 2 - stopped

2.3.15 Firmware Date 1F52_h

INDEX	1F52_h
Name	<i>verify application software</i>
Data Type	unsigned 32
Default Value	No

The object contains the generation date and time of the application software.

Index [Hex]	Sub-index [Dec]	Description	Value range [Hex]	Default [Dec]	Data type	R/W
1F52	0	<i>no_of_entries</i>	2	2	unsigned 8	ro
	1	<i>application_software_date</i>	0...FFFFFFFF	-	unsigned 32	ro
	2	<i>application_software_time</i>	0...05265BFF	-	unsigned 32	ro

Parameter Description:

application_software_date date of the modules application software, counted in days since the 01.01.1984.

Example: 0000 196D_h -> 26.10.2001

application_software_time time in ms since midnight at the day of the generation of the last application software

Example: 029FFF40_h -> 12:14

2.3.16 Object Receive PDO Communication Parameter 1400_h

Via this object the features of a receive PDO are defined.

INDEX	1400_h
Name	<i>receive PDO parameter</i>
Data Type	PDOCommPar

Index	Sub-index [Dec]	Description	Value range [Hex]	Default [Hex]	Data type	R/W
1400	0	number of entries	3	3	unsigned 8	const.
	1	COB-ID used by PDO	0...FFFFFFFF	200 _h + Node-ID	unsigned 32	rw
	2	transmission type	no	255	unsigned 8	ro
	3	inhibit time	no	0	unsigned 16	ro

2.3.17 Receive PDO Mapping Parameter 1600_h

Via object 'Receive PDO Mapping Parameter 1600_h' the assignment of receive data to the Rx-PDO can be changed.

INDEX	1600_h
Name	<i>receive PDO mapping</i>
Data Type	PDO Mapping

The following table represents the assignment of receive PDO-mapping parameters for default configuration:

Index	Sub-index [Dec]	Description	Value range [Hex]	Default [Hex]	Data type	R/W
1600	0	number of entries	4	4	unsigned 8	ro
	1	mapping for object 1	0...FFFFFFFF	6200 0108 _h	unsigned 32	ro
	2	mapping for object 2	0...FFFFFFFF	6200 0208 _h	unsigned 32	ro
	3	mapping for object 3	0...FFFFFFFF	6200 0308 _h	unsigned 32	ro
	4	mapping for object 4	0...FFFFFFFF	6200 0408 _h	unsigned 32	ro

2.3.18 Object Transmit PDO Communication Parameter 1800_h ...1802_h

Via this object the features of a transmit PDO are defined.

INDEX	1800_h ... 1802_h
Name	<i>transmit PDO parameter</i>
Data Type	PDOCommPar

Index	Sub-index [Dec]	Description	Value range [Hex]	Default [Hex]	Data type	R/W
1800	0	number of entries	5	5 _h	unsigned 8	const.
	1	COB-ID used by PDO	0...FFFFFFFF	180 _h + Node-ID	unsigned 32	rw
	2	transmission type	0...FF	255 _d	unsigned 8	rw
	3	inhibit time	0...FFFF	0	unsigned 16	ro
	4	CMS priority group	0...FF	6	unsigned 8	ro
	5	event timer	0...FFFFFFFF	0 (3E8 _h)	unsigned 16	rw
1801	0	number of entries	5	5 _h	unsigned 8	const.
	1	COB-ID used by PDO	0...FFFFFFFF	280 _h + Node-ID	unsigned 32	rw
	2	transmission type	0...FF	255 _d	unsigned 8	rw
	3	inhibit time	0...FFFF	0	unsigned 16	ro
	4	CMS priority group	0...FF	6	unsigned 8	ro
	5	event timer	0...FFFFFFFF	0	unsigned 16	rw
1802	0	number of entries	5	5 _h	unsigned 8	const.
	1	COB-ID used by PDO	0...FFFFFFFF	380 _h + Node-ID	unsigned 32	rw
	2	transmission type	0...FF	255 _d	unsigned 8	rw
	3	inhibit time	0...FFFF	0	unsigned 16	ro
	4	CMS priority group	0...FF	6	unsigned 8	ro
	5	event timer	0...FFFFFFFF	0 (270F _h)	unsigned 16	rw

Value range in accordance with DS-301

2.3.19 Transmit PDO Mapping Parameter 1A00_h ...1A02_h

Via object ‘Transmit PDO Mapping Parameter 1A00_h’ the assignment of transmit data to Tx-PDOs can be changed.

INDEX	1A00_h ...1A02_h
Name	<i>transmit PDO mapping</i>
Data Type	PDO Mapping

The following table represents the assignment of transmit PDO-mapping parameters for default configuration:

Index	Sub-index [Dec]	Description	Value range [Hex]	Default [Hex]	Data type	R/W
1A00	0	number of entries	8	8 _h	unsigned 8	ro
	1	mapping for object 1	0...FFFFFFFF	6000 0108 _h	unsigned 32	ro
	2	mapping for object 2	0...FFFFFFFF	6000 0208 _h	unsigned 32	ro
	3	mapping for object 3	0...FFFFFFFF	6000 0308 _h	unsigned 32	ro
	4	mapping for object 4	0...FFFFFFFF	6000 0408 _h	unsigned 32	ro
	5	mapping for object 5	0...FFFFFFFF	6000 0508 _h	unsigned 32	ro
	6	mapping for object 6	0...FFFFFFFF	6000 0608 _h	unsigned 32	ro
	7	mapping for object 7	0...FFFFFFFF	6000 0708 _h	unsigned 32	ro
	8	mapping for object 8	0...FFFFFFFF	6000 0808 _h	unsigned 32	ro
1A01	0	number of entries	7	7 _h	unsigned 8	ro
	1	mapping for object 1	0...FFFFFFFF	6200 0108 _h	unsigned 32	ro
	2	mapping for object 2	0...FFFFFFFF	6200 0208 _h	unsigned 32	ro
	3	mapping for object 3	0...FFFFFFFF	6200 0308 _h	unsigned 32	ro
	4	mapping for object 4	0...FFFFFFFF	6200 0408 _h	unsigned 32	ro
	5	mapping for object 5	0...FFFFFFFF	2803 0108 _h	unsigned 32	ro
	6	mapping for object 6	0...FFFFFFFF	2803 0208 _h	unsigned 32	ro
	7	mapping for object 7	0...FFFFFFFF	2803 0308 _h	unsigned 32	ro
1A02	0	number of entries	8	8 _h	unsigned 8	ro
	1	mapping for object 1	0...FFFFFFFF	6400 0108 _h	unsigned 32	ro
	2	mapping for object 2	0...FFFFFFFF	6400 0208 _h	unsigned 32	ro
	3	mapping for object 3	0...FFFFFFFF	6400 0308 _h	unsigned 32	ro
	4	mapping for object 4	0...FFFFFFFF	6400 0408 _h	unsigned 32	ro
	5	mapping for object 5	0...FFFFFFFF	6400 0508 _h	unsigned 32	ro
	6	mapping for object 6	0...FFFFFFFF	6400 0608 _h	unsigned 32	ro
	7	mapping for object 7	0...FFFFFFFF	6400 0708 _h	unsigned 32	ro
	8	mapping for object 8	0...FFFFFFFF	6400 0808 _h	unsigned 32	ro

Value range in accordance with DS-301

2.4 Transmission Types Supported in Accordance with DS-301

Transmission Type	PDO transmission					Supported by esd-module
	cyclic	acyclic	synchronous	asynchronous	RTR only	
0		X	X			NO
1...240	X		X			NO
241... 251	reserved					n.a.
252			X		X	NO
253				X	X	YES
254				X		YES
255				X		YES

2.5 Table of Important Identifier and Messages of CANopen

CAN Identifier [hex]	Name of Object	Length	Data [hex]	Note
0	NMT	2	02 xx	module changes to state 'Stopped'
0	NMT	2	01 xx	Start (module changes to state 'Operational')
0	NMT	2	80 xx	module changes to state 'Preoperational'
0	NMT	2	81 xx	reset module
0	NMT	2	82 xx	reset communication (here same function as 81xx)
700 _h + Node-ID	Heartbeat	1 byte	state	NMT error control identifier (see DS-301)
580 _h + Node-ID	SDO	8 bytes	parameter	acknowledge of the modules communication parameters
600 _h + Node-ID	SDO	8 bytes	parameter	transfer of the communication parameters to the module (Rx)

xx = Node-ID

Node-ID. = 1...7F_h

2.6 PDO-Mapping Summary

2.6.1 Rx-PDO1: Setting the outputs Q0 to Q31

Rx-PDO1 (default-setting)	Data Byte 1								Data Byte 2								Data Byte 3								Data Byte 4							
	Bit...								Bit...								Bit...								Bit...							
201 _h	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Output Q...																															
	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24

2.6.2 Tx-PDO1: Reading the Inputs I0 to I64

Tx-PDO1 (default-setting)	Data Byte 5								Data Byte 6								Data Byte 7								Data Byte 8							
	Bit...								Bit...								Bit...								Bit...							
181 _h	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Input I...																															
	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56

Tx-PDO1 (default-setting)	Data Byte 1								Data Byte 2								Data Byte 3								Data Byte 4							
	Bit...								Bit...								Bit...								Bit...							
181 _h	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Input I...																															
	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24

2.6.3 Tx-PDO2: Status Messages

Tx-PDO2 (default-setting)	Data Byte 1								Data Byte 2								Data Byte 3								Data Byte 4							
	Bit...								Bit...								Bit...								Bit...							
281 _h	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Read back of the programmed desired values (prog. OUT) of output Q...																															
	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24

Tx-PDO2	Data Byte 5								Data Byte 6								Data Byte 7								Data Byte 8							
	<i>Output_Error_Group_Q0-Q31</i>								<i>Output Supply VCC-Error Group Q0-Q31</i>								<i>Output Supply VCC-Status Group Q0-Q31</i>								not transmitted							
281 _h																																

2.6.4 Tx-PDO3: Values of the Supply Voltages

Tx-PDO2 (default-setting)	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7	Data Byte 8
	<i>code_7</i>	<i>code_6</i>	<i>code_5</i>	<i>code_4</i>	<i>code_3</i>	<i>code_2</i>	<i>code_1</i>	<i>code_0</i>
381 _h								

2.7 Device Profile Area

2.7.1 Implemented Objects 6000_h-6400_h

Index [HEX]	Name	Data type
6000	<i>Read Input 8-bit</i>	unsigned8
6005	<i>Global Interrupt Enable Digital</i>	boolean
6006	<i>Interrupt Mask Any Change 8-bit</i>	unsigned8
6200	<i>Write Output 8-bit</i>	unsigned8
6206	<i>Error Mode Output 8-bit</i>	unsigned8
6207	<i>Error Value Output 8-bit</i>	unsigned8
6400	<i>Read Analog Input 8-Bit</i>	unsigned8

2.7.2 Read Input 8-Bit (6000_h)

Index [Hex]	Sub-index	Description	Value range [Hex]	Default	Data type	R/W
6000	0	<i>Number_of_entries</i>	8	8	unsigned 8	ro
	1	<i>Read_Input_I7-I0</i>	00...FF	-	unsigned 8	ro
	2	<i>Read_Input_I15-I8</i>	00...FF	-	unsigned 8	ro
	3	<i>Read_Input_I23-I16</i>	00...FF	-	unsigned 8	ro
	4	<i>Read_Input_I31-I24</i>	00...FF	-	unsigned 8	ro
	5	<i>Read_Input_I39-I32</i>	00...FF	-	unsigned 8	ro
	6	<i>Read_Input_I47-I40</i>	00...FF	-	unsigned 8	ro
	7	<i>Read_Input_I55-I48</i>	00...FF	-	unsigned 8	ro
	8	<i>Read_Input_I63-I56</i>	00...FF	-	unsigned 8	ro

Assignment of the Variables *Read_Input_I...* to the Inputs I0...I63:

Index: 6000_h, Sub-index: 1...8

Sub-index \ Bit	Bit							
	7	6	5	4	3	2	1	0
1	<i>I7</i>	<i>I6</i>	<i>I5</i>	<i>I4</i>	<i>I3</i>	<i>I2</i>	<i>I1</i>	<i>I0</i>
2	<i>I15</i>	<i>I14</i>	<i>I13</i>	<i>I12</i>	<i>I11</i>	<i>I10</i>	<i>I9</i>	<i>I8</i>
3	<i>I23</i>	<i>I22</i>	<i>I21</i>	<i>I20</i>	<i>I19</i>	<i>I18</i>	<i>I17</i>	<i>I16</i>
4	<i>I31</i>	<i>I30</i>	<i>I29</i>	<i>I28</i>	<i>I27</i>	<i>I26</i>	<i>I25</i>	<i>I24</i>
5	<i>I39</i>	<i>I38</i>	<i>I37</i>	<i>I36</i>	<i>I35</i>	<i>I34</i>	<i>I33</i>	<i>I32</i>
6	<i>I47</i>	<i>I46</i>	<i>I45</i>	<i>I44</i>	<i>I43</i>	<i>I42</i>	<i>I41</i>	<i>I40</i>
7	<i>I55</i>	<i>I54</i>	<i>I53</i>	<i>I52</i>	<i>I51</i>	<i>I50</i>	<i>I49</i>	<i>I48</i>
8	<i>I63</i>	<i>I62</i>	<i>I61</i>	<i>I60</i>	<i>I59</i>	<i>I58</i>	<i>I57</i>	<i>I56</i>

An input bit is read as '1', if the corresponding input is active, i.e. voltage is 'on'.

2.7.3 Global Interrupt Enable Digital (6005_h)

Index [Hex]	Sub-index	Description	Value range [Boolean]	Default	Data type	R/W
6005	0	<i>Global Interrupt Enable-Bit</i> = false : interrupts disabled <i>Global Interrupt Enable-Bit</i> = true : interrupts enabled	True, False	1	boolean	rw

2.7.4 Interrupt Mask Any Change 8-Bit (6006_h)

Index [Hex]	Sub-index	Description	Value range [Hex]	Default	Data type	R/W
6006	0	<i>No_of_entries</i>	8	8	unsigned 8	const
	1	<i>IRQ-Mask_Any_I7-I0</i>	00...FF	FF	unsigned 8	r/w
	2	<i>IRQ-Mask_Any_I15-I8</i>	00...FF	FF	unsigned 8	r/w
	3	<i>IRQ-Mask_Any_I23-I16</i>	00...FF	FF	unsigned 8	r/w
	4	<i>IRQ-Mask_Any_I31-I24</i>	00...FF	FF	unsigned 8	r/w
	5	<i>IRQ-Mask_Any_I39-I32</i>	00...FF	FF	unsigned 8	r/w
	6	<i>IRQ-Mask_Any_I47-I40</i>	00...FF	FF	unsigned 8	r/w
	7	<i>IRQ-Mask_Any_I55-I48</i>	00...FF	FF	unsigned 8	r/w
	8	<i>IRQ-Mask_Any_I63-I56</i>	00...FF	FF	unsigned 8	r/w

Assignment of Variables *IRQ-Mask_Any_I...* to the Inputs I0...I63:

Index: 6006_h, Sub-index: 1...8

Bit Sub-index	7	6	5	4	3	2	1	0
1	<i>I7</i>	<i>I6</i>	<i>I5</i>	<i>I4</i>	<i>I3</i>	<i>I2</i>	<i>I1</i>	<i>I0</i>
2	<i>I15</i>	<i>I14</i>	<i>I13</i>	<i>I12</i>	<i>I11</i>	<i>I10</i>	<i>I9</i>	<i>I8</i>
3	<i>I23</i>	<i>I22</i>	<i>I21</i>	<i>I20</i>	<i>I19</i>	<i>I18</i>	<i>I17</i>	<i>I16</i>
4	<i>I31</i>	<i>I30</i>	<i>I29</i>	<i>I28</i>	<i>I27</i>	<i>I26</i>	<i>I25</i>	<i>I24</i>
5	<i>I39</i>	<i>I38</i>	<i>I37</i>	<i>I36</i>	<i>I35</i>	<i>I34</i>	<i>I33</i>	<i>I32</i>
6	<i>I47</i>	<i>I46</i>	<i>I45</i>	<i>I44</i>	<i>I43</i>	<i>I42</i>	<i>I41</i>	<i>I40</i>
7	<i>I55</i>	<i>I54</i>	<i>I53</i>	<i>I52</i>	<i>I51</i>	<i>I50</i>	<i>I49</i>	<i>I48</i>
8	<i>I63</i>	<i>I62</i>	<i>I61</i>	<i>I60</i>	<i>I59</i>	<i>I58</i>	<i>I57</i>	<i>I56</i>

This object determines, which input port lines shall activate an interrupt by **positive or negative edge** detection.

Bit-value <i>I_x</i>	Interrupt-enable
0	interrupt disabled
1	interrupt enabled

2.7.5 Write Output 8-Bit (6200_h)

Index [Hex]	Sub-index	Description	Value range [Hex]	Default	Data type	R/W
6200	0	<i>No_of_entries</i>	4	4	unsigned 8	const
	1	<i>Write_Output_Q7-Q0</i>	00...FF	0	unsigned 8	rw
	2	<i>Write_Output_Q15-Q8</i>	00...FF	0	unsigned 8	rw
	3	<i>Write_Output_Q23-Q16</i>	00...FF	0	unsigned 8	rw
	4	<i>Write_Output_Q31-Q24</i>	00...FF	0	unsigned 8	rw

Assignment of the Variable *Write_Output_Q...* to the outputs Q0...Q31:

Index: 6200_h, Sub-index: 1...4

Bit Sub-index	7	6	5	4	3	2	1	0
1	<i>Q7</i>	<i>Q6</i>	<i>Q5</i>	<i>Q4</i>	<i>Q3</i>	<i>Q2</i>	<i>Q1</i>	<i>Q0</i>
2	<i>Q15</i>	<i>Q14</i>	<i>Q13</i>	<i>Q12</i>	<i>Q11</i>	<i>Q10</i>	<i>Q9</i>	<i>Q8</i>
3	<i>Q23</i>	<i>Q22</i>	<i>Q21</i>	<i>Q20</i>	<i>Q19</i>	<i>Q18</i>	<i>Q17</i>	<i>Q16</i>
4	<i>Q31</i>	<i>Q30</i>	<i>Q29</i>	<i>Q28</i>	<i>Q27</i>	<i>Q26</i>	<i>Q25</i>	<i>Q24</i>

If an output bit is set to '1', the corresponding output is activated, i.e. the output voltage is 'on'.

2.7.6 Error Mode Output 8-Bit (6206_h)

The error mode is evaluated if the module is in the *Stopped* state. This object together with object 1029_h indicate, whether an output is set to an error-value defined in object 6207_h in case of an internal device error.

Index [Hex]	Sub-index	Description	Value range [Hex]	Default [Hex]	Data type	R/W
6206	0	<i>No_of_entries</i>	4	4	unsigned 8	const
	1	<i>Error_Mode_Output_Q7-Q0</i>	00...FF	FF	unsigned 8	rw
	2	<i>Error_Mode_Output_Q15-Q8</i>	00...FF	FF	unsigned 8	rw
	3	<i>Error_Mode_Output_Q23-Q16</i>	00...FF	FF	unsigned 8	rw
	4	<i>Error_Mode_Output_Q31-Q24</i>	00...FF	FF	unsigned 8	rw

Assignment of the Variable *Error_Mode_Output_Q...* to the outputs Q0...Q31:

Index: 6206_h, Sub-index: 1...4

Bit Sub-index	7	6	5	4	3	2	1	0
1	<i>Q7</i>	<i>Q6</i>	<i>Q5</i>	<i>Q4</i>	<i>Q3</i>	<i>Q2</i>	<i>Q1</i>	<i>Q0</i>
2	<i>Q15</i>	<i>Q14</i>	<i>Q13</i>	<i>Q12</i>	<i>Q11</i>	<i>Q10</i>	<i>Q9</i>	<i>Q8</i>
3	<i>Q23</i>	<i>Q22</i>	<i>Q21</i>	<i>Q20</i>	<i>Q19</i>	<i>Q18</i>	<i>Q17</i>	<i>Q16</i>
4	<i>Q31</i>	<i>Q30</i>	<i>Q29</i>	<i>Q28</i>	<i>Q27</i>	<i>Q26</i>	<i>Q25</i>	<i>Q24</i>

Value	Meaning
1	Output shall take the pre-defined value specified in object 6207 _h .
0	Output shall be kept unchanged if an error occurs.

2.7.7 Error Value Output 8-Bit (6207_h)

On condition that the corresponding error mode (object 6206_h) is active, device failures shall set the output to the value configured by this object.

Index [Hex]	Sub-index	Description	Value range [Hex]	Default [Hex]	Data type	R/W
6207	0	<i>No_of_entries</i>	1	1	unsigned 8	const
	1	<i>Error_Value_Output_Q7-Q0</i>	00...FF	0	unsigned 8	rw
	2	<i>Error_Value_Output_Q15-Q8</i>	00...FF	0	unsigned 8	rw
	3	<i>Error_Value_Output_Q23-Q16</i>	00...FF	0	unsigned 8	rw
	4	<i>Error_Value_Output_Q31-Q24</i>	00...FF	0	unsigned 8	rw

Assignment of the Variable *Error_Value_Output_Q...* to the outputs Q0...Q31:

Index: 6207_h, Sub-index: 1...4

Bit Sub-index	7	6	5	4	3	2	1	0
1	<i>Q7</i>	<i>Q6</i>	<i>Q5</i>	<i>Q4</i>	<i>Q3</i>	<i>Q2</i>	<i>Q1</i>	<i>Q0</i>
2	<i>Q15</i>	<i>Q14</i>	<i>Q13</i>	<i>Q12</i>	<i>Q11</i>	<i>Q10</i>	<i>Q9</i>	<i>Q8</i>
3	<i>Q23</i>	<i>Q22</i>	<i>Q21</i>	<i>Q20</i>	<i>Q19</i>	<i>Q18</i>	<i>Q17</i>	<i>Q16</i>
4	<i>Q31</i>	<i>Q30</i>	<i>Q29</i>	<i>Q28</i>	<i>Q27</i>	<i>Q26</i>	<i>Q25</i>	<i>Q24</i>

The variable contains the value, the outputs shall be set to, in case of fault.

Value	Meaning
1	Output shall be set to '1' (enabled) in case of fault, if object 6206 _h is enabled.
0	Output shall be set to '0' (disabled) in case of fault, if object 6206 _h is enabled.

2.7.8 Read Analog Input 8-Bit (6400_h)

Index [Hex]	Sub-index	Description	Value range [Hex]	Default [Hex]	Data type	R/W
6400	0	<i>No_of_entries</i>	8	8	unsigned 8	const
	1	<i>Analog_Input_8_code_0</i>	00..FF	-	unsigned 8	ro
	2	<i>Analog_Input_8_code_1</i>	00..FF	-	unsigned 8	ro
	3	<i>Analog_Input_8_code_2</i>	00..FF	-	unsigned 8	ro
	4	<i>Analog_Input_8_code_3</i>	00..FF	-	unsigned 8	ro
	5	<i>Analog_Input_8_code_4</i>	00..FF	-	unsigned 8	ro
	6	<i>Analog_Input_8_code_5</i>	00..FF	-	unsigned 8	ro
	7	<i>Analog_Input_8_code_6</i>	00..FF	-	unsigned 8	ro
	8	<i>Analog_Input_8_code_7</i>	00..FF	-	unsigned 8	ro

Assignment of the Variable *Analog_Input_8_code_...* to the monitored power supplies of the outputs Q0...Q31:

Index: 6400_h, Sub-index: 1...8

Sub-index	Variable	Returns supply voltage code of outputs
1	<i>Analog_Input_8_code_0</i>	<i>Q0...Q3</i>
2	<i>Analog_Input_8_code_1</i>	<i>Q4...Q7</i>
3	<i>Analog_Input_8_code_2</i>	<i>Q8...Q11</i>
4	<i>Analog_Input_8_code_3</i>	<i>Q12...Q15</i>
5	<i>Analog_Input_8_code_4</i>	<i>Q16...Q19</i>
6	<i>Analog_Input_8_code_5</i>	<i>Q20...Q23</i>
7	<i>Analog_Input_8_code_6</i>	<i>Q24...Q27</i>
8	<i>Analog_Input_8_code_7</i>	<i>Q28...Q31</i>

Value Range of the Variables *Analog_Input_8_code_...*:

The measured voltage is determined by the following equation:

$$V_{CC} = 5.0V \cdot 11 \cdot \frac{\text{Analog_Input_8_code_x}}{256}$$

The resolution of the measured values is 1 LSB, i.e. 0.2148 V.

2.8 Manufacturer Specific Profile Area

2.8.1 Implemented Objects 2800_h-2804_h

Index [HEX]	Name	Data type
2800	<i>Output Driver Supply Voltage Threshold Low</i>	unsigned8
2801	<i>Output Driver Supply Voltage Threshold High</i>	unsigned8
2802	<i>Output Driver Supply Voltage Status</i>	unsigned8
2803	<i>Output Driver Status</i>	unsigned8
2804	<i>Output Driver Error Mask and Time</i>	unsigned8

2.8.2 Output Driver Supply Voltage Threshold Low (2800_h)

Index [Hex]	Sub-index	Description	Value range [Hex]	Default [Hex]	Data type	R/W
2800	0	<i>Number_of_entries</i>	8	8	unsigned 8	ro
	1	<i>Voltage_Threshold_Low_Q0-Q3</i>	00...FF	53	unsigned 8	rw
	2	<i>Voltage_Threshold_Low_Q4-Q7</i>	00...FF	53	unsigned 8	rw
	3	<i>Voltage_Threshold_Low_Q8-Q11</i>	00...FF	53	unsigned 8	rw
	4	<i>Voltage_Threshold_Low_Q12-Q15</i>	00...FF	53	unsigned 8	rw
	5	<i>Voltage_Threshold_Low_Q16-Q19</i>	00...FF	53	unsigned 8	rw
	6	<i>Voltage_Threshold_Low_Q20-Q23</i>	00...FF	53	unsigned 8	rw
	7	<i>Voltage_Threshold_Low_Q24-Q27</i>	00...FF	53	unsigned 8	rw
	8	<i>Voltage_Threshold_Low_Q28-Q31</i>	00...FF	53	unsigned 8	rw

The value that must be set for a desired voltage threshold (V_{MIN}) has is determined as follows:

$$Voltage_Threshold_Low_x = \frac{V_{MIN} \cdot 256}{5.0V \cdot 11}$$

with

Voltage_Threshold_Low_x ... value, that must be transmitted
 V_{MIN} ... desired voltage threshold

Example:

Default value 53_h => $V_{MAX} = 17.8 V$

2.8.3 Output Driver Supply Voltage Threshold High (2801_h)

Index [Hex]	Sub-index	Description	Value range [Hex]	Default [Hex]	Data type	R/W
2801	0	<i>Number_of_entries</i>	8	8	unsigned 8	ro
	1	<i>Voltage_Threshold_High_Q0-Q3</i>	00...FF	8B	unsigned 8	rw
	2	<i>Voltage_Threshold_High_Q4-Q7</i>	00...FF	8B	unsigned 8	rw
	3	<i>Voltage_Threshold_High_Q8-Q11</i>	00...FF	8B	unsigned 8	rw
	4	<i>Voltage_Threshold_High_Q12-Q15</i>	00...FF	8B	unsigned 8	rw
	5	<i>Voltage_Threshold_High_Q16-Q19</i>	00...FF	8B	unsigned 8	rw
	6	<i>Voltage_Threshold_High_Q20-Q23</i>	00...FF	8B	unsigned 8	rw
	7	<i>Voltage_Threshold_High_Q24-Q27</i>	00...FF	8B	unsigned 8	rw
	8	<i>Voltage_Threshold_High_Q28-Q31</i>	00...FF	8B	unsigned 8	rw

The value that must be set for a desired voltage threshold (V_{MAX}) has is determined as follows:

$$Voltage_Threshold_High_x = \frac{V_{MAX} \cdot 256}{5.0V \cdot 11}$$

with

Voltage_Threshold_High_x ... value, that must be transmitted
 V_{MAX} ... desired voltage threshold

Example:

Default value $8B_h \Rightarrow V_{MAX} = 29.8 V$

2.8.4 Output Driver Supply Voltage Status (2802_h)

Index [Hex]	Sub-index	Description	Value range [Hex]	Default [Hex]	Data type	R/W
2802	0	<i>Number_of_entries</i>	8	8	unsigned 8	ro
	1	<i>Supply Voltage Status Output_Q0-Q3</i>	00...03	-	unsigned 8	ro
	2	<i>Supply Voltage Status Output_Q4-Q7</i>	00...03	-	unsigned 8	ro
	3	<i>Supply Voltage Status Output_Q8-Q11</i>	00...03	-	unsigned 8	ro
	4	<i>Supply Voltage Status Output_Q12-Q15</i>	00...03	-	unsigned 8	ro
	5	<i>Supply Voltage Status Output_Q16-Q19</i>	00...03	-	unsigned 8	ro
	6	<i>Supply Voltage Status Output_Q20-Q23</i>	00...03	-	unsigned 8	ro
	7	<i>Supply Voltage Status Output_Q24-Q27</i>	00...03	-	unsigned 8	ro
	8	<i>Supply Voltage Status Output_Q28-Q31</i>	00...03	-	unsigned 8	ro

Parameter Description:

These parameters return the output drivers's supply status in binary code. There is one bit for the VCC error and one bit for the VCC status for each output group. The least significant bits of the parameter *Supply Voltage Status Output_x* together have the following meaning:

Bits of the parameter <i>Supply Voltage Status Output_x</i>			Value of the supply voltage	Returned status
Bit 7...2	Bit 1 (<i>vccerr</i>)	Bit 0 (<i>vccstat</i>)		
0	0	0	$V_{MIN} < V_{CC} < V_{MAX}$	OK (V_{CC} is within the defined working range)
0	0	1	$V_{ERR} < V_{CC} < V_{MIN}$	V_{CC} is lower than the defined working range, but higher than V_{ERR}
0	1	0	$V_{MAX} < V_{CC}$	V_{CC} is too high
0	1	1	$V_{CC} < V_{ERR}$	V_{CC} is too low

Bit 7..2 are always returned as '0'.

Supply voltage monitor switching thresholds:

- V_{ERR} = 9.0 V (fix, not programmable)
- V_{MIN} = 17.8 V (default value, programmable by object 2800_h)
- V_{MAX} = 29.8 V (default value, programmable by object 2801_h)

2.8.5 Output Driver Status (2803_h)

Index [Hex]	Sub-index	Description	Value range [Hex]	Default	Data type	R/W
2803	0	<i>Number_of_entries</i>	3	3	unsigned 8	ro
	1	<i>Output_Error_Group_Q0-Q31</i>	00...FF	-	unsigned 8	ro
	2	<i>Output Supply VCC-Error Group Q0-Q31</i>	00...FF	-	unsigned 8	ro
	3	<i>Output Supply VCC-Status Group Q0-Q31</i>	00...FF	-	unsigned 8	ro

Parameter Description:

Output_Error_Group_Q0-Q31

This byte returns the error status of the 8 output groups in binary code. If an error signal of a groups is active, the according bit is set to '1'.

Bits of parameter <i>Output_Error_Group_Q0-Q31</i>	7	6	5	4	3	2	1	0
Errors signal of output group...	<i>Q28-Q31</i>	<i>Q24-Q27</i>	<i>Q20-Q23</i>	<i>Q16-Q19</i>	<i>Q12-Q15</i>	<i>Q8-Q11</i>	<i>Q4-Q7</i>	<i>Q0-Q3</i>

**Output Supply VCC-Error Group Q0-Q31 and
Output Supply VCC-Status Group Q0-Q31**

These bytes return the status of the output drivers's supply voltage in binary code. Each bytes has got one bit for the VCC error and one bit for the VCC status for each output group. The assignment of the bits to the output groups is the same as used for the parameter *Output_Error_Group_Q0-Q31* (refer table above).

The two bits from the parameters *Output Supply VCC-Error Group Q0-Q31* and *Output Supply VCC-Status Group Q0-Q31* that are assigned to one output group have together the following meaning:

Value of the parameter bits of one channel		Value of the supply voltage	Returned status
<i>Output Supply VCC-Error Group Q0-Q31</i>	<i>Output Supply VCC-Status Group Q0-Q31</i>		
0	0	$V_{MIN} < V_{CC} < V_{MAX}$	OK (V_{CC} is within the defined working range)
0	1	$V_{ERR} < V_{CC} < V_{MIN}$	V_{CC} is lower than the defined working range, but higher than V_{ERR}
1	0	$V_{MAX} < V_{CC}$	V_{CC} is too high
1	1	$V_{CC} < V_{ERR}$	V_{CC} is too low

Supply voltage monitor switching thresholds:

- V_{ERR} = 9.0 V (fix, not programmable)
- V_{MIN} = 17.8 V (default value, programmable by object 2800_h)
- V_{MAX} = 29.8 V (default value, programmable by object 2801_h)

2.8.6 Output Driver Error Mask and Time (2804_h)

Index [Hex]	Sub-index	Description	Value range [Hex]	Default [Hex]	Data type	R/W
2804	0	<i>Number_of_entries</i>	3	3	unsigned 8	ro
	1	<i>Output_Error_Group_Mask_Q0-Q31</i>	00...FF	00	unsigned 8	rw
	2	<i>Output Supply VCC-Error-Status Group Mask Q0-Q31</i>	00...FF	00	unsigned 8	rw
	3	<i>Output Error Hold Time Q0-Q31</i>	00...FF	10	unsigned 8	rw

Parameter Description:

Output_Error_Group_Mask_Q0-Q31

This byte defines the error message determination of the 8 output groups. The assignment of the output groups to the parameter bits is as follows:

Bits of parameter <i>Output_Error_Group_Mask_Q0-Q31</i>	7	6	5	4	3	2	1	0
Errors signal of output group...	<i>Q28-Q31</i>	<i>Q24-Q27</i>	<i>Q20-Q23</i>	<i>Q16-Q19</i>	<i>Q12-Q15</i>	<i>Q8-Q11</i>	<i>Q4-Q7</i>	<i>Q0-Q3</i>

Is one of these bits set to '1', this has effects to the initiation of transmission of frames and to the LED display in the front panel:

1. An alternation of the common error status of the according group initiates a transmission of an error message via Tx-PDO2.
2. An error in the according output group activates the signal 'Err-LED' (error display is activated).

Output Supply VCC-Error-Status Group Mask Q0-Q31

Is one of these bits set to '1', this has effects to the initiation of transmission of frames and to the LED display in the front panel, too. The assignment of the output groups to the parameter bits is the same as at parameter *Output_Error_Group_Mask_Q0-Q31*. It is shown in the table above.

1. An alternation of the supply voltage monitor's status of the according output group initiates a transmission of an error message via Tx-PDO2 and a transmission of the voltage values via Tx-PDO3.
2. Is the signal of the according output group active, the signal 'VCC-OK' is deactivated (VCC LED is 'off' or 'flashing').

Output Error Hold Time Q0-Q31

By means of this parameter the Error-Hold-Time is set for the common error message for all output groups together.

The Error-Hold-Time is used for holding the error signal active. This is necessary, because the error signal is set inactive together with the output signal by the automatically internal protection circuit of the driver. The output is set active again automatically after a wait time T_{OFF} , even if the reason for the error is still present. This would cause an alternating error output signal.

The following figure shows a principle timing example:

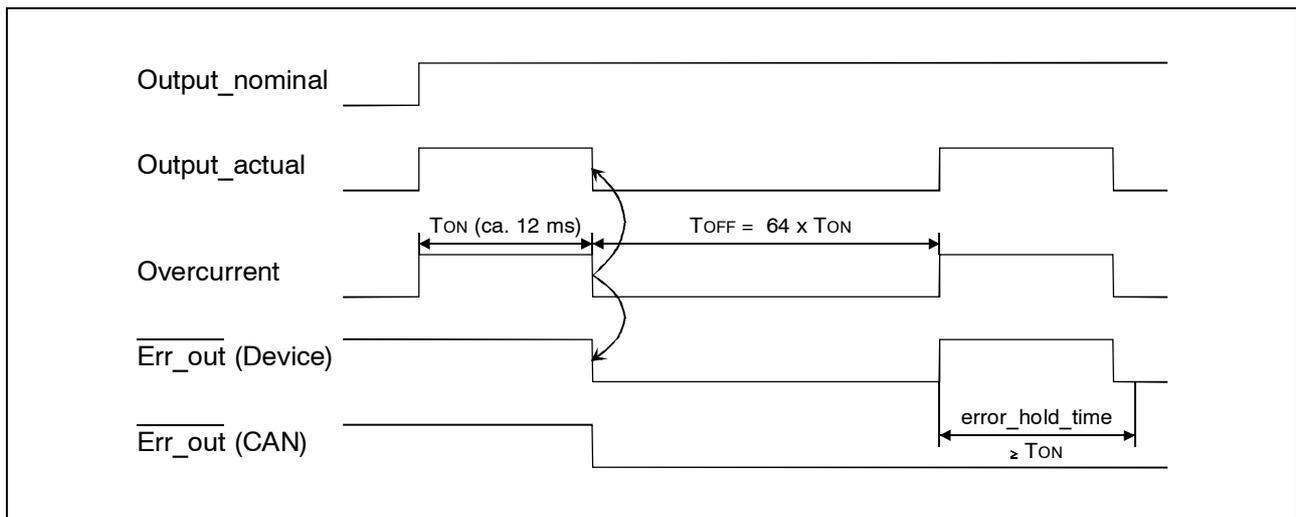


Fig. : Timing example of the error signal

After the first error signal is activated, the Error-Hold-Time shall prevent the deactivation of the following error signal for the duration of at least T_{ON} . Therefore the value of *Output Supply Error Hold Time Q0-Q31* has always be set to values greater than T_{ON} .