SYS TEC
ELECTRONIC

CANopen IO-C12

Systems Manual

Edition October 2013

system house for distributed automation

In this manual are descriptions for copyrighted products which are not explicitly indicated as such. The absence of the trademark (©) symbol does not infer that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this manual

The information in this document has been carefully checked and is believed to be entirely reliable. However, SYS TEC electronic GmbH assumes no responsibility for any inaccuracies. SYS TEC electronic GmbH neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. SYS TEC electronic GmbH reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages which might result.

Additionally, SYS TEC electronic GmbH offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. SYS TEC electronic GmbH further reserves the right to alter the layout and/or design of the hardware without prior notification and accepts no liability for doing so.

© Copyright 2013 SYS TEC electronic GmbH. rights – including those of translation, reprint, broadcast, photomechanical or similar reproduction and storage or processing in computer systems, in whole or in part – are reserved. No reproduction may occur without the express written consent from SYS TEC electronic GmbH.

Contact	**Direct**	**Your local distributor**
Address:	SYS TEC electronic GmbH Am Windrad 2 D-08468 Heinsdorfergrund GERMANY	Please find a list of our distributors under http://www.systec-electronic.com/distributors
Ordering Information:	+49 (3765) / 38600-0 info@systec-electronic.com	
Technical Support:	+49 (3765) 38600-0 support@systec-electronic.com	
Fax:	+49 (3765) 38600-4100	
Web Site:	http://www.systec-electronic.com	

7th Edition October 2013

1 Preface 1
2 Introduction to the CANopen IO-C12 3
2.1 Technical Highlights 3
2.2 Device Pinout 4
2.3 Connector description 5
2.4 Pin Assignment and Description 5
2.5 Board Configuration 8
2.5.1 DIP-Switch S202 8
2.5.2 HEX-encoding Switch 9
2.5.3 Restore factory default settings 11
2.6 CAN Interface 12
2.7 Technical Specification 13
3 Commissioning and configuration of the CANopen IO-C12 15
3.1 Power Supply 15
3.2 CAN Interface 15
3.3 Relay contact 16
3.4 Interface connection digitalen Output 16
3.5 Interface connection PWM Output 17
4 QuickStart 19
4.1 Start-Up of the CANopen IO-C12 19
4.2 Shut-Down of the CANopen IO-C12 19
4.3 CAN Message and Identifier 20
4.4 PDO Mapping for I/O’s 20
4.5 Board Reset 21
4.6 Node-Guarding 21
5 Controller Area Network – CAN 25
5.1 Communication with CANopen 25
5.2 CAN Application Layer 27
5.3 CANopen – Open Industrial Communication 28
6 CANopen Communication 31
6.1 CANopen Fundamentals 31
6.2 CANopen Device Profiles 32
6.3 Communication Profile 32
6.4 Service Data Objects 33
6.5 Process Data Objects 34
6.6 PDO-Mapping 37
6.7 Error Handling 40
6.8 Network Services 40
6.8.1 Life-Guarding 40
6.8.2 Heartbeat 41
6.8.3 Heartbeat Producer 41

6.8.4 Heartbeat Consumer 42
6.9 Network Boot-Up 42
6.10 Object Dictionary Entries 45
6.11 Input/Output Assignment to Object Dictionary Entries 47
7 CANopen IO-C12 Operation 49
7.1 CANopen State Transitions 49
7.2 Power On 50
7.3 PRE-OPERATIONAL 50
7.4 OPERATIONAL 50
7.5 STOPPED 50
7.6 Restart Following Reset / Power-On 50
7.7 Functions of the digital Inputs 52
7.7.1 Global interrupt enabling of digital inputs 6005H 52
7.7.2 Interrupt Mask rising and falling edge - 6006H 52
7.7.3 Interrupt Mask rising edge - 6007H 53
7.7.4 Interrupt Mask falling edge - 6008H 53
7.8 Functions of the digital outputs 53
7.9 Analog Input Operation 53
7.9.1 Handling Analog Values 53
7.9.2 Formula for Calculating the Analog Input Value 54
7.9.3 Selecting the Interrupt Trigger 55
7.9.4 Interrupt Source 55
7.9.5 Interupt Enable 56
7.9.6 Interrupt Upper and Lower Limit 56
7.9.7 Delta Function 58
7.10 Functions of the Analog Outputs 58
7.11 Functions of the PWM outputs 59
7.11.1 Special functionality using phyPS-409-KSM01 modules 59
7.12 Emergency Message 60
7.12.1 Error Code 61
7.12.2 Error Register 61
7.13 Status LEDs 61
7.13.1 RUN LED 61
7.13.2 Error Led 62
7.14 Hardware versions 63
7.15 Manufacturing data 63
8 Operations in the Event of Errors 67
8.1 State of the CANopen IO-C12 in the Event of Errors 67
8.2 Output Handling in the Event of Errors 67
8.3 Changing from Error State to Normal Operation 68

9 CANopen IO-C12 Object Dictionary 71
10 Firmware-Update 73
Revision History of this Document 79
Index 81

 © SYS TEC electronic GmbH 2013

Figure 1: Device pinout ... 4
Figure 2: Pinout for two rowed connectors ... 5
Figure 3: Pinout for the RJ-11 connector ... 5
Figure 4: DIP-switch ... 8
Figure 5: HEX-encoding Switch S200 and S201 ... 10
Figure 6: Example for node address 62H ... 10
Figure 7: Description of relay contact (NO normally open and change over contact) ... 16
Figure 8: Example connection digitalen outputs ... 16
Figure 9: Example connection PWM outputs ... 17
Figure 10: State Diagram of a CANopen Device ... 44
Figure 11: Firmware-Update – Connection to PC ... 73
Figure 12: Firmware-Update – Selection of COM-interface ... 74
Figure 13: Firmware-Update – Target configuration ... 74
Figure 14: Firmware-Update – Program-mode ... 75
Figure 15: Firmware-Update – Position Boot and Reset ... 75
Figure 16: Firmware-Update – Dialog for program-mode ... 76
Figure 17: Firmware-Update – Dialog after download ... 76
Figure 18: Firmware-Update – Dialog after erase ... 77
Figure 19: Firmware-Update – Dialog after programming ... 77

Table 1: Pin assignment for all connectors 7

Table 2: Configuration of CAN Bit Rate 9

Table 3: list of node addresses in hexadecimal and decimal notation.... 11

Table 4: Maximum cable length depending on cable profile and number of connected nodes 12

Table 5: Environmental Parameters 13

Table 6: Communication Interfaces 13

Table 7: IO configuration 14

Table 8: CAN ID for Different PDO Types 20

Table 9: PDO Mapping for I/O‘s 21

Table 10: COB-Identifier (Communication Target Object Identifier)..... 36

Table 11: PDO Mapping Example 39

Table 12: Emergency-Message Contents 40

Table 13: Heartbeat Message Structure.. 41

Table 14: Structure of a Consumer Heartbeat Time Entry....................... 42

Table 15: Calculation of the COB-Identifier from the Node Addresses.. 43

Table 16: Base Identifier 43

Table 17: Description of State Flow Diagram Symbols........................... 44

Table 18: Object Dictionary Input/Output Entries 47

Table 19: NMT-Master Messages for Status Control 49

Table 20: Storage of Analog Values 54

Table 21: Interrupt Trigger Bits 55

Table 22: Emergency Message... 60

Table 23 States of the RUN LED.. 62

Table 24: States of the ERROR LED 63

Table 25 extendet error indication via LED DI0 to DI7 63

Table 26: Example for Error Handling on relay outputs.......................... 68

Table 27: : Example for Error Handling on analog outputs 68

Table 28 : Object Dictionary of the CANopen IO-C12 module 72

 © SYS TEC electronic GmbH 2013

1 Preface

This manual describes the functions and technical specifications of the CANopen IO-C12 module.
Further information on CANopen can be found in the appropriate technical documentation.

Declaration of the Electro Magnetic Conformity for the CANopen IO-C12

The CANopen IO-C12 (henceforth product) is designed for installation in electrical appliances or as dedicated Evaluation Boards (i.e.: for use as a test and prototype platform for hardware/software development) in laboratory environments.

Note:
It is necessary that only appropriately trained personnel (such as electricians, technicians and engineers) handle and/or operate these products.
SYS TEC products fulfill the norms of the European Union's Directive for Electro Magnetic Conformity only in accordance to the descriptions and rules of usage indicated in this hardware manual (particularly in respect to the connectors, power connector and serial interface).

Implementation of SYS TEC products into target applications, as well as user modifications and extensions of SYS TEC products, is subject to renewed establishment of conformity to, and certification of, Electro Magnetic Directives. Users should ensure conformance following any modifications to the products as well as implementation of the products into target systems.

 © SYS TEC electronic GmbH 2013

© SYS TEC electronic GmbH 2013

2 Introduction to the CANopen IO-C12

The CANopen IO-C12 module was desined to provide an easy way to access and configure digital and analog IOs remotely, using the standardized CANopen protocol.

The CANopen IO-C12 module features the complete functionality of a CANopen Slave device. The CANopen conformance test issued by the CiA (CAN in Automation e.V.) is in progress.
The present version of the CANopen IO-C12 supports 11-Bit identifier (CAN 2.0B passive). The firmware supports the standard Device Profile according to **CiA DSP401** ,the Communication Profile according to **CiA DS301 V4.01** and the Indicator specification according to CiA DR303-3 V1.0. The CANopen C12 is fully configurable via CAN and features the non-volatile storage of all configuration data.

2.1 Technical Highlights

- CANopen device according to CiA standard DS401
-
- non-volatile storage of configuration data
- Easy configuration of CAN-bus bitrate via LSS[1] or DIP-Switch
- Configuration of CAN node address via easy accessible HEX-encoding switches
- All inputs and outputs feature LED indicators
- 24 digital inputs, 24VDC, seperated in groups of 4 inputs each, each group galvanic isolated to another
- 3 digital inputs, 24VDC, galvanic isolated
- 4 Relay outputs 250VAC/ 2A
- 16 transistor buffered outputs 24VDC/ 0,5A, pluse switching, protected against short-circuit
- 4 analog inputs, 10-bit resolution, 0...10 V
- 2 analog outputs, 8 Bit resolution, 0-10 V

[1] LSS Layer Setting Services according to CiA standard CiA DSP-305

- 2 PWM[1] outputs, 0-24V/ 0,5 A, LowSide Switch, open Collector, short-circuit proof
- galvanic isolated high-speed CAN Bus interface, CAN-bus driver supports up to 64 nodes on one bus
- RS232 interface, used for firmware update
- Internal power supply, 24 VDC/0,5A ±20%

2.2 Device Pinout

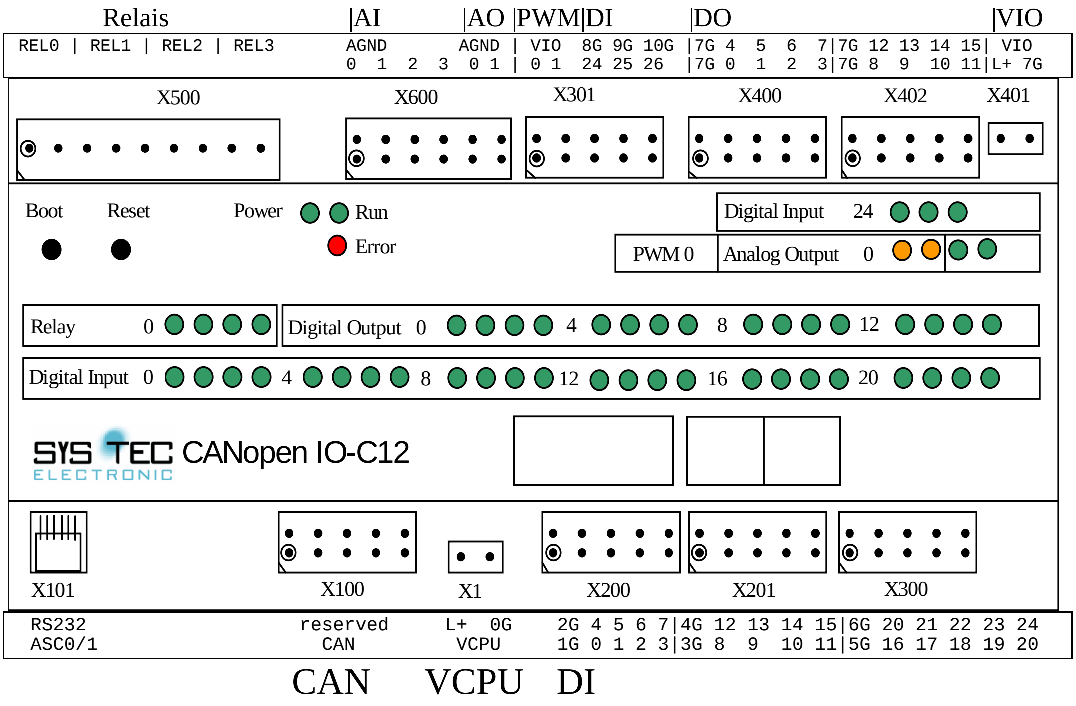

Figure 1: Device pinout

[1] PWM : **P**ulse **W**idth **M**odulation

 © SYS TEC electronic GmbH 2013

2.3 Connector description

On every connector pin 1 is marked with a circle or inclined egde.

For two rowed connectors the pinout is defined as following:

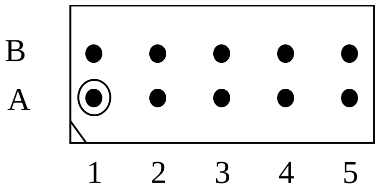

Figure 2: Pinout for two rowed connectors

For the RJ-11 connector the pinout is defined as following:

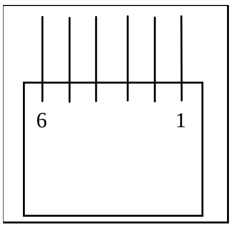

Figure 3: Pinout for the RJ-11 connector

2.4 Pin Assignment and Description

Interface	Process Name	Pin	Label
power supply CPU	VCPU, +24VDC Ground GND_VCPU	X1.1 X1.2	L+ 0G
power supply for IOs	VIO, +24VDC Ground GND_VIO	X401.1 X401.2	L+ 7G
ASC0, RS-232 interface for firmware update	TxD, RS232 level	X101.2	
	GND	X101.3	
	RxD, RS232 level	X101.4	
CAN-bus	reserved	X100.5A	
	CAN_H	X100.4A	
	reserved	X100.3A	
	CAN_L	X100.2A	
	CAN_GND	X100.1A	
	Ground GND_VCPU	X1.2	
Digital Inputs DI0..7	common ground DI0...DI3	X200.1A	1G
	DI0	X200.2A	0

	DI1	X200.3A	1
	DI2	X200.4A	2
	DI3	X200.5A	3
	common ground DI4..DI7	X200.1B	2G
	DI4	X200.2B	4
	DI5	X200.3B	5
	DI6	X200.4B	6
	DI7	X200.5B	7
Digital Inputs DI8..15	Common ground DI8..DI11	X201.1A	3G
	DI8	X201.2A	8
	DI9	X201.3A	9
	DI10	X201.4A	10
	DI11	X201.5A	11
	Common ground DI12..DI15	X201.1B	4G
	DI12	X201.2B	12
	DI13	X201.3B	13
	DI14	X201.4B	14
	DI15	X201.5B	15
Digital Inputs DI16..23	Common ground DI16..DI19	X300.1A	5G
	DI16	X300.2A	16
	DI17	X300.3A	17
	DI18	X300.4A	18
	DI19	X300.5A	19
	Common ground DI20..DI23	X300.1B	6G
	DI20	X300.2B	20
	DI21	X300.3B	21
	DI22	X300.4B	22
	DI23	X300.5B	23
Digital Inputs DI24..26	DI24	X301.3A	24
	Ground DI24	X301.3B	8G
	DI25	X301.4°	25
	Ground DI25	X301.4B	9G
	DI26	X301.5°	26
	Ground DI26	X301.5B	10G
Digital Outputs DO0..7	Common ground GND_VIO	X400.1A	7G
	DO0	X400.2A	0
	DO1	X400.3A	1
	DO2	X400.4A	2
	DO3	X400.5A	3
	Common ground GND_VIO	X400.1B	7G
	DO4	X400.2B	4
	DO5	X400.3B	5

© SYS TEC electronic GmbH 2013 L-1046e_7

	DO6	X400.4B	6
	DO7	X400.5B	7
Digital output DO8..15	Common ground GND_VIO	X402.1A	7G
	DO8	X402.2A	8
	DO9	X402.3A	9
	DO10	X402.4A	10
	DO11	X402.5A	11
	Common ground GND_VIO	X402.1B	7G
	DO12	X402.2B	12
	DO13	X402.3B	13
	DO14	X402.4B	14
	DO15	X402.5B	15
PWM outputs	PWM output 0	X301.1A	P0
	+24VDC IO	X301.1B	VIO
	PWM output 1	X301.2A	P1
	+24VDC IO	X301.2B	VIO
Analog Inputs AI0..3	AI0	X600.1A	0
	Common ground GND_VCPU	X600.1B	AGND
	AI1	X600.2A	1
	Common ground GND_VCPU	X600.2B	AGND
	AI2	X600.3A	2
	Common ground GND_VCPU	X600.3B	AGND
	AI3	X600.4A	3
	Common ground GND_VCPU	X600.4B	AGND
Analog Outputs AO0..1	AO0	X600.5A	0
	Common ground GND_VCPU	X600.5B	AGND
	AO1	X600.6A	1
	Common ground GND_VCPU	X600.6B	AGND
Dry-Contact Outputs (Relay)	REL0 C	X500.1	
	REL0 NO	X500.2	
	REL1 C	X500.3	
	REL1 NO	X500.4	
	REL2 C	X500.5	
	REL2 NO	X500.6	
	REL3 C	X500.7	
	REL3 NO	X500.8	
	REL3 NC	X500.9	

Table 1: Pin assignment for all connectors

2.5 Board Configuration

There are two input units available to configure the CANopen IO-C12 module.

- 8-position DIP-Switch
- HEX-encoding Switches

Their usage for configuring the module is described in the following sections.

2.5.1 DIP-Switch S202

The 8-position DIP-switch (S202) is located on the topside of the CANopen IO-C12 module. Four of these switches enable configuration of the CAN bitrate. The valid CAN bitrate is stored in the module. Any changes made on DIP-switches become valid after reboot of the CANopen C12 module.
It is also possible to set the bitrate via LSS (Layer Setting Services). The bitrate configured via LSS is also non-volatile stored and become available after rebooting the device. If a valid prevoisly stored configuration was found the device ignores the DIP-switch settings.
Find more information on how to reset the configuration in section 2.5.3

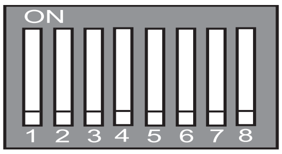

Figure 4: DIP-switch

© SYS TEC electronic GmbH 2013

The following table gives an overview of the possible configurations for bitrate:

DIP-switch								Bitrate kBit/s
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	1000
ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	800
OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	500
ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	250
OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	125
ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	100
OFF	ON	ON	OFF	OFF	OFF	OFF	OFF	50
ON	ON	ON	OFF	OFF	OFF	OFF	OFF	20
OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	10
OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	1000

Table 2: Configuration of CAN Bit Rate

All bitrates shown in *Table 2* are defined in the CiA standard DSP-305 V1.01. Invalid DIP-switch configurations are indicated by the ERROR LED (see section 7.13). DIP8 = ON is reserved for production only, must not use.

2.5.2 HEX-encoding Switch

The CANopen IO-C12 module is equipped with two HEX-encoding switches marked S200 (MSB) and S201 (LSB). The two HEX-encoding switches are intended for configuration purposes prior to operation.
These switches are used for setting up the node address when integrating the CANopen IO-C12 into a CANopen network. This network type requires a unique node number for each individual control unit connected to the CANopen system. Assigning the same node number to two different devices will result in functional problems. Please note that the node numbers 00H and ≥80H are reserved and must not be used. The node address is stored –on the device at power-on after the settings were changed. The ERROR Led indicates configuration errors (see section 7.13). The node address

can also be set via LSS[1]. When using LSS the node address is stored on the device.
If an valid perviously stored configuration is available the HEX-encoding switches are ignored at power-on.
Figure 5 shows the assignment of MSB and LSB to the single switches.

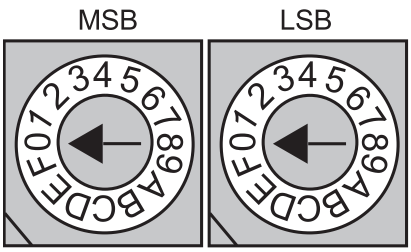

Figure 5: *HEX-encoding Switch S200 and S201*

Figure 6 shows the positions of the single HEX-encoding switches for assigning node address 62H or 98_{dez}.

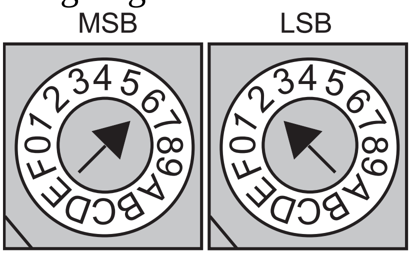

Figure 6: *Example for node address 62H*

ATTENTION!
If the number FFH is selected at power-on, the node address and bitrate stored on device get marked invalid.
Select a new node address and bitrate. After the next power-on the selected configuration is stored on the device.

[1] LSS – Layer Setting Service according to CiA standard DS305

 © SYS TEC electronic GmbH 2013

Knoten-nummer dez	Knoten-nummer hex	Knoten-nummer dez	Knoten-nummer hex	Knoten-nummer dez	Knoten-nummer hex	Knoten-nummer dez	Knoten-nummer hex
1	1	33	21	65	41	97	61
2	2	34	22	66	42	98	62
3	3	35	23	67	43	99	63
4	4	36	24	68	44	100	64
5	5	37	25	69	45	101	65
6	6	38	26	70	46	102	66
7	7	39	27	71	47	103	67
8	8	40	28	72	48	103	68
9	9	41	29	73	49	104	69
10	0A	42	2A	74	4A	106	6A
11	0B	43	2B	75	4B	107	6B
12	0C	44	2C	76	4C	108	6C
13	0D	45	2D	77	4D	109	6D
14	0E	46	2E	78	4E	110	6E
15	0F	47	2F	79	4F	111	6F
16	10	48	30	80	50	112	70
17	11	49	31	81	51	113	71
18	12	50	32	82	52	114	72
19	13	51	33	83	53	115	73
20	14	52	34	84	54	116	74
21	15	53	35	85	55	117	75
22	16	54	36	86	56	118	76
23	17	55	37	87	57	119	77
24	18	56	38	88	58	120	78
25	19	57	39	89	59	121	79
26	1A	58	3A	90	5A	122	7A
27	1B	59	3B	91	5B	123	7B
28	1C	60	3C	92	5C	124	7C
29	1D	61	3D	93	5D	125	7D
30	1E	62	3E	94	5E	126	7E
31	1F	63	3F	95	5F	127	7F
32	20	64	40	96	60		

Table 3: list of node addresses in hexadecimal and decimal notation

2.5.3 Restore factory default settings

To restore the factory default settings on the CANopen IO-C12 the number FFH has to be selected on the HEX-encoding switches and power-on or reset has to be performed. All previously stored configuration data (node-address, bitrate) will be deleted. Also all changes made to the Object dictionary are resetted.

2.6 CAN Interface

The CAN Bus transceiver used is galvanic isolated from the CPU. Power for the CAN-bus transceiver is supplied by the on-board DC/DC converter.

CAN Bus Cable

It is recommended to use a twisted pair CAN bus cable, terminated with a resistor of 124 Ohm between CAN_H and CAN_L at both ends. According to CiA recommendation DRP 303-1 the CAN ground line should be included and connected.
Please refer to the corresponding CiA standards for further information.
Recommended cable profiles according to the CiA standard CiA DRP 303-1:

Cable profile	Specific resistance	max. length in m (safety margin 0.2)			max. length in m (safety margin 0.1)		
		n=32	n=64	n=100	n=32	n=64	n=100
0.25 mm²	70 mΩ/m	200	170	150	230	200	170
0.5 mm²	< 40 mΩ/m	360	310	270	420	360	320
0.75 mm²	< 26 mΩ/m	550	470	410	640	550	480

Table 4: Maximum cable length depending on cable profile and number of connected nodes

If the number of nodes grows above 64 or the cable length is longer than 250m, the pecision of the supplied voltage for the CAN tranceiver PCA82C251 need to be better then 5%.

The connector’s contact resistance should be 2.5 .. 10 mΩ [CiA DRP 303-1].

 © SYS TEC electronic GmbH 2013

2.7 Technical Specification

Environmental Parameters		**Typical**	**Maximum**
power supply	V_{CPU}	24VDC	±20%
	V_{IO}	24VDC	±20%
Current consumption (inactive IOs)	I_{CPU}	0.180mA	
	I_{IO}	0.180mA	
Temperature Range	Storage temperature		-20°..+70°C
	Operating temperature		0°..+50°C
Protection class	Housing	IP20	
Weight	without any cable and packing	350g	
Dimensions	Width Height Depth		160mm 75mm 95mm
Connector type	Spring type connector		

Table 5: Environmental Parameters

Communication Interfaces		**Min.**	**Max.**
CAN-Bus			
CAN0	Bitrate	10kBit/s	1Mbit/s
	Max. number of nodes		64
	CAN-H, CAN-L short-circuit-proof towards 24V		
RS-232			
ASC0	Baudrate	1200Baud	38400Baud

Table 6: Communication Interfaces

I/O-configuration			
Digital Output DO0 .. 15			
24VDC Output (High Side Switch)	U_{OH} at I_{OH} = 500 mA	V_{IO}-0,16V < U_{OH} < V_{IO}	
	U_{OL} at I_{OL} = 0 mA		0.5V
	Current limitation I_{OH_max}		625mA
	Max. current		8A
	$I_{OL(off)}$		10µA
	t_{off} at I_{OH} = 500 mA	115µs	190µs
	t_{on} at I_{OH} = 500 mA	75µs	125µs
Digital Outputs REL0 .. 3			
Relay output (changer)	Switching Voltage		250AC

	Switching Current		6A
	Durability (mech.)		$1x10^5$
	t_{on}	5ms	
	t_{off}	2,5ms	
	Isolation		4kV
Digital Inputs DI0 .. 23			
24VDC-Inputs, pulse switching	U_{IH}	15V	30V
	U_{IL}	-3V	5V
	I_{IH}	3mA	8,5mA
Analog Inputs AI0 .. 3			
0 .. +10V	Measurement range U_I	0..+10.107V	± 1.0% ± 1LSB
	Destructive voltage U_{I_max}		>30V
	Input resistance R_I	115.18kΩ ±0.1%	
	Physical Resolution		10Bit
PWM outputs 0..1			
24VDC-PWM-Output (Low Side Switch)	U_{OL} at I_{OL} = -500mA		<1V
	$I_{OH(off)}$		20µA
	I_{OH_max}		0.6A
	t_{on} at I_{OL} = -500mA		2.5µs
	t_{off} at I_{OL} = -500mA		3.5µs
	PWM Frequency max. (CAN bit rate =10kBit)	15Hz	11,5kHz
	PWM Frequency max. (CAN bit rate >10kBit)	15Hz	15kHz
Analog Outputs AO0 .. 1			
0 .. +10V	Voltage Range U_O	0 – +10.35V	± 1.0% ± 1LSB
	Output current I_O		30mA
	Output capacity		10nF
	Physical Resolution		8 Bit

Table 7: IO configuration

© SYS TEC electronic GmbH 2013

3 Commissioning and configuration of the CANopen IO-C12

3.1 Power Supply

The CANopen IO-C12 needs an operating voltage of +24VDC for the CPU core board and the IO board. Power has to be connected to VCPU und VIO (*see Figure 1*).

3.2 CAN Interface

The CANopen IO-C12 module features a MB90F543 microcontroller with integrated FULL CAN interface.
The CAN interface is galvanic decoupled and brought out to connector X100.

The pinout for the X100 connector is assigned as following:

CAN_HIGH → X100.4A
CAN_LOW → X100.2A
CAN_GND → X100.1A

Using a SUB-D9 plug the signals have to be connected as following: (according to CiA)

connector pin	Signal	SUB-D9 pin
X100.4A	→CAN_HIGH	→ PIN 7
X100.2A	→CAN_LOW	→ PIN 2
X100.1A	→CAN_GND	→ PIN 6 and/or PIN 3

(according to CiA standard DS 301 for Communication Profile)

At this point the CANopen IO-C12 is ready for communication.

3.3 Relay contact

The following Figure show the assignment between the symbols and the relay contact.

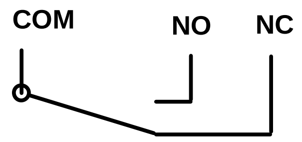

Figure 7: Description of relay contact (NO normally open and change over contact)

3.4 Interface connection digitalen Output

The following Figure show the connection of load at highside Switch for transistor outputs up DO0 to DO15.

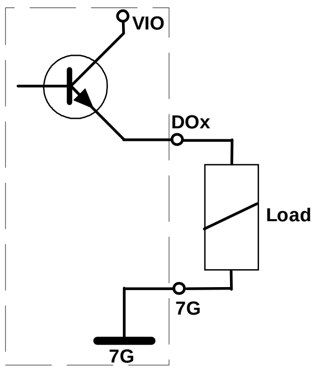

Figure 8: Example connection digitalen outputs

© SYS TEC electronic GmbH 2013

3.5 Interface connection PWM Output

The following Figure show the connection of load at lowside Switch for PWM outputs up P0 to P1.

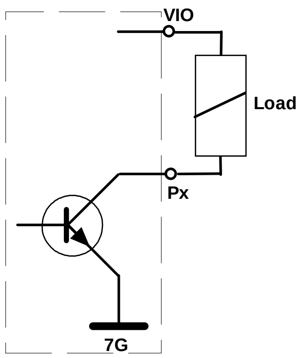

Figure 9: Example connection PWM outputs

 © SYS TEC electronic GmbH 2013

© SYS TEC electronic GmbH 2013

4 QuickStart

This section describes basic start-up of the CANopen IO-C12. It assumes basic knowledge of CANopen networks. It also requires, that the CANopen IO-C12 is properly connected to the CAN bus and power is supplied to the CANopen IO-C12. Please *refer to sections 5* and *6* for basic description of CAN and CANopen.

4.1 Start-Up of the CANopen IO-C12

All values in the Object Dictionary (OD) are pre-configured with default-values. Hence, start-up configuration of the CANopen IO-C12 is not necessary. The CANopen IO-C12 supports the CANopen Minimum Boot-Up. Following reset and internal initialization, the board is in PRE-OPERATIONAL state (*refer to section 7.3 PRE-OPERATIONAL*). Upon receipt of a single message (*Start_Remote_Node*) the board switches to OPERATIONAL state (*refer to section 7.4 OPERATIONAL*).

11-bit CAN Identifier	2 Byte Data	
0	*0x01h*	*Node_ID*

The first data byte contains the Start command, while the second byte contains the node address (*Node_ID*). If you specify the value 00h then all nodes in the network (Broadcast) are started.

4.2 Shut-Down of the CANopen IO-C12

All node activity can be stopped by receipt of the *Enter_Pre_Operational_State* message. This message consists of the following:

11-bit CAN Identifier	2 Byte Status Data	
0	*0x80h*	*Node_ID*

The *Node-ID* identifies the node-addresses. *Node-ID = 00h* addresses all devices on a network (Broadcast).

When in “PRE-OPERATIONAL“ state, SDO-Transfer is still possible. All configuration parameters are then captured and “frozen” as they were in their most recent active state.

Note: CANopen configuration tools, such as the CANopen Configuration Manager (CCM) or CANopen Device Monitor (CDM) always use SDO-Transfer to access the CANopen device. Thus, those tools also can work when the device is in state “PRE-OPERATIONAL“.

4.3 CAN Message and Identifier

According to **CiA Draft Standard 301**, a specific CAN identifier is assigned to each CAN message containing process data (Process-Data-Object - PDO). The CAN identifier for input and output data is derived from the node address.

CAN-Identifier (hex)	Data type
180H + node id	1. Tx PDO
280H + node id	2. Tx PDO
380H + node id	3. Tx PDO
200H + node id	1. Rx PDO
300H + node id	2. Rx PDO
400H + node id	3. Rx PDO

Table 8: CAN ID for Different PDO Types

4.4 PDO Mapping for I/O’s

The PDO mapping of the available I/O's depends on the selected I/O configuration. In the default mapping, the 3rd Tx PDO and the 3rd Rx PDO are invalid. This results in the following arrangement of I/Os and PDOs:

Byte number	1. Tx PDO	2. Tx PDO	1. Rx PDO	2. Rx PDO
0	DI 0..7	AI 0	DO 0..7	AO 0
1	DI 8..15		DO 8..15	AO 1

© SYS TEC electronic GmbH 2013 L-1046e_7

2	DI 16..23	AI 1	REL 0..4	invalid
3	DI 24..26		invalid	invalid
4	invalid	AI 2	invalid	invalid
5	invalid		invalid	invalid
6	invalid	AI 3	invalid	invalid
7	invalid		invalid	invalid

Table 9: PDO Mapping for I/O's

The CANopen IO-C12 also supports variable PDO mapping. This allows for free mapping of inputs to Tx PDOs and Rx PDOs to output lines. Such free mapping settings can be saved in the on-board EEPROM by writing to index 0x1010.

4.5 Board Reset

Following each board reset, the CANopen IO-C12 transmits an Bootup message without data content. Temporary suspension of CANopen IO-C12 activity and subsequent restart can be recognized without Nodeguarding *(refer to section 4.6 Node Guarding)*. The transmitter of this message will be detected by the CAN identifier.

11-bit CAN Identifier	1 Byte Data
700h+ Node_ID	*00h*

4.6 Node-Guarding

Nodeguarding and *Lifeguarding* functions monitor operation of the CANopen network. Distributed peripheral CAN devices are monitored via Nodeguarding, while the Lifeguarding function supervises the guarding Master. To realize Nodeguarding, the Master requests a cyclic status message from the slave nodes. This status request is initiated with a Remote frame message that contains only the status data. The RTR-Bit (Remote Transmit Request Bit) is set for this reason.

11-bit CAN Identifier	1 Byte Data
700h + Node_ID	*Node-Guarding*

Following transmission of the Remote-Frame message, the Slave-nodes responds with a status message consisting of 1 byte of service data.

11-bit CAN Identifier	1 Byte Data
700h + Node_ID	*X*

The data bytes within status message further contain a toggle bit that is supposed to change after each message. Should the status and toggle bits not correspond to the message pattern expected by the Master, or should no response to a message follow, the Master assumes a Slave malfunction. If the Master requests cyclic guard messages, a Slave node can recognize shut-down of the Master. This is the case if the Slave does not receive a message request from the Master within the pre-configured *Life Time*. The Slave then assumes failure of the Master, sets its inputs into Error state, transmits an Emergency message and switches into *Pre-Operational* state.

The *Life Time Factor* is configured within the Object [100D] and is multiplied by the *Guard Time* [100C]. This results in the *Life Time* of the "Nodeguard Protocol". The time base of these cycles is 1 ms. The *Guard Time* specifies how much time must elapse between two *Node-Guarding* messages. The *Life Time Factor* indicates how many times the *Guard Time* can elapse before an error is generated.

Default Values:

Life Time Factor	0
Guard Time	0 ms.
Life Time	0 sec.

Example Values:

Life Time Factor	3
Guard Time	1000 ms.
Life Time	3 sec.

© SYS TEC electronic GmbH 2013

5 Controller Area Network – CAN

5.1 Communication with CANopen

The Controller Area Network (the CAN bus) is a serial data communications bus for real-time applications. CAN was originally developed by the German company Robert Bosch for use in the automotive industry. It is a two-wire bus system that provides a cost-effective communication bus alternative to expensive and cumbersome harness wiring. CAN operates at data rates of up to 1 Mbit per second and has excellent error detection and confinement capabilities. On account of its proven reliability and robustness, CAN is being used in many other automation and industrial applications. CAN is now an international standard and is documented in ISO 11898 (for high-speed applications) and ISO 11519 (for lower-speed applications) documents.

CANopen is a higher-layer network protocol based on the CAN serial bus system, specifically, it is a software-level protocol standard for industrial communication between automated devices. CANopen is authorized by the User and Manufacturers' Group "CAN in Automation e.V." (CiA) and adheres to ISO/OSI standards.

CANopen unleashes the full power of CAN by allowing direct peer to peer data exchange between nodes in an organized, heirarchical manner. The network management functions specified in CANopen simplify project design, implementation and diagnosis by providing standard mechanisms for network start-up and error management.

CANopen supports both cyclic and event-driven communication. This makes it possible to reduce the bus load to a minimum, while still maintaining extremely short reaction times. High communication performance can be achieved at relatively low baud rates, thus reducing EMC problems and minimizing cable costs. CANopen is the ideal networking system for all types of automated machinery. One of the distinguishing features of CANopen is its support for data exchange at the supervisory control level as well as accommodating the integration of very small sensors and actuators on the same physical network. This avoids the unnecessary expense of gateways linking sensor/actuator bus systems with higher communication networks and makes CANopen particularly attractive to original equipment manufacturers.

CANopen Advantages

- Vendor-independent open-source structure
- Universal standards
- Supports inter-operability of different devices
- High speed real-time capability
- Modular - covers simple to complex devices
- User-friendly - wide variety of support tools available
- Real-Time-capable communication for process data without protocol overhead;
- a modular, configurable structure that can be tailored to the needs of the user and his or her networked application
- Interbus-S, Profibus and MMS oriented-profiles

CANopen Features

- Auto configuration of the network
- Easy access to all device parameters
- Device synchronization
- Cyclic and event-driven data transfer
- Synchronous reading or setting of inputs, outputs or parameters

In addition to its designation as a physical CAN layer standard, CANopen is a “layer-7 protocol” implementation of CAL and is defined by the CANopen Communications Profile in CiA DS-301.

CAL, in turn, is based on an existing and proven protocol originally developed by Philips Medical Systems. CAL is an application-independent application layer that has been specified and is also maintained by the CAN in Automation (CiA) user group.

 © SYS TEC electronic GmbH 2013 L-1046e_7

5.2 CAN Application Layer

The CAN Application Layer (CAL) supports various applications and the integration of CAN hardware from different vendors. A CAL implementation consists of four blocks, each of which can operate as network Master and Slave.

CAN Message Specification (CMS)

CMS defines the communication objects, such as multiplexed variables, Events and Domains.

- Variables: serve data exchange of basic messages
- Events: handle the activity of specifically defined events, such as switches and transmission of asynchronous messages
- Domains: support transmission of data packages larger than the maximum eight bytes of a standard CAN message

CMS further regulates the communication structure between the object targets.

Network Management (NMT)

NMT implements network management functions for NMT-Master and NMT-Slave. These functions support start-up and expansion of a network, as well as Error supervision (Lifeguarding) and prevention of bus overload.

Distributor (DBT)

DBT supports the use of CAN nodes from various vendors through its automatic assigning of message identifiers. The DBT-Master/Slave functions enable administration of a global data basis for communication objects (COBs) of varying priority classes.

Layer Management (LMT)

LMT assigns parameters to the lower layers of data communication, such as timing parameters of CAN nodes or management of a manufacturer code by node name designation.

 © SYS TEC electronic GmbH 2013

5.3 CANopen – Open Industrial Communication

The following Special Interest and Working Groups have developed the CAL-based CANopen communication profile:

- SIG Distributed I/O – Chairmanship ***Selectron***
- SIG Motion Control - Chairmanship ***port***

and the Working Group (WG)

- WG Higher Layer Protocols

The CiA DS 301 CANopen standard derived from the results of the ASPIC ESPRIT project. The communication profile describes in detail how data are exchanged over the CAN bus based on the functions provided by CAL. This data can be sorted into two main types:

- Process data
- Service data

Process data is real-time data generated by a networked device. This data is transmitted via a Process Data Object (PDO). The CANopen communication profile determines how a PDO functions within CAL communication objects, as well as which protocol is used for transmission of data. PDOs can be used simultaneously by multiple networked devices, hence enabling broadcast operations.

Service data are used to configure and establish parameters for networked devices. Service data directly communicate to the Object Dictionary of each device and are transmitted using Service Data Objects (SDO).

The CANopen communication profile also determines how these objects are connected and which CAL functions and services can be used. An SDO can only be used between two networked devices, typically a configuration Master and another device that is to be

© SYS TEC electronic GmbH 2013

configured. The SDO-Transfer is also capable of confirmation of message receipt.

Each individual networked device provides several PDOs and SDOs. This enables configuration of multi-master networks, in addition to typical single Master / multiple Slave networks.

In addition to data classes, CANopen defines the communication classes that describe:

- Synchronized communication
- Event processing
- Communication initialization

CANopen also defines device profiles that describe the basic functions of networked devices. These device profiles consist of the following two primary components:

- Functional Description
- Operational Description

The ***Functional Description*** of a device is represented by functional blocks and data flows. Descriptive parameters are stored in the Object Dictionary. Each Object Dictionary has a pre-defined structure. Hence, parameters for networked devices of a certain type (for instance I/O modules or drives) are always located in the same place within an Object Dictionary. Parameters can be classified as mandatory, optional and manufacturer-specific.

The ***Operational Description*** of a device is described by state flow diagram (*refer to Figure 10 in Section 6.9*).

Device Profiles are standardized for:

- Generic I/O Modules CiA DS 401
  digital I/O's
  analog I/O's
- Drives and Motion Control CiA DSP 402

 © SYS TEC electronic GmbH 2013

Servo drivers,
Step motors and
Frequency transformers

- Measurement Devices and Closed Loop Controllers CiA DSP 404
- IEC61131-3 Programmable Devices CiA DSP 405
- Encoder CiA DSP 406
- Inclinometer CiA DSP 410

Please refer to the CAN in Automation homepage www.can-cia.org for up-to-date information of available device profiles. All device profiles correspond to the DRIVECOM Profile with CAN-specific modifications to enable multi-master capability.

Software for CANopen Slave functions is based on services for data exchange and network management as defined in CAL standards. In particular only certain parts of CAL have been implemented in CANopen, such as standards for Multiplexed-Domain-Transfer for SDOs and Stored Event-Transfer for transmission of PDOs.

 © SYS TEC electronic GmbH 2013

6 CANopen Communication

6.1 CANopen Fundamentals

Open fieldbus systems enable design of distributed network systems by connecting components from multiple vendors while minimizing the effort required for interfacing. To achieve an open networking system, it is necessary to standardize the various layers of communication used.

CANopen uses the international CAN standard, ISO 11898 as the basis for communication. This standard covers the lower two layers of communication specified by the OSI model. Based on this, the CANopen profile family specifies standardized communication mechanisms and device functionality for CAN-based systems. The profile family, which is available and maintained by CAN in Automation e.V. (CiA) consists of the Application layer and communication profile (DS 301), various frameworks and recommendations (CiA DS-30x) and various device profiles (CiA DS-40x).

The network management functions specified in CANopen simplify project design, implementation and diagnosis by providing standard mechanisms for network start-up and error management.

CANopen is the ideal networking system for all types of automated machinery. One of the distinguishing features of CANopen is its support for data exchange at the supervisory control level as well as accommodating the integration of very small sensors and actuators on the same physical network. This avoids the unnecessary expense of gateways linking sensor/actuator bus systems with higher communication networks and makes CANopen particularly attractive to original equipment manufacturers.

 © SYS TEC electronic GmbH 2013

6.2 CANopen Device Profiles

CANopen profiles are defined for communication in CiA Draft Standard 301, for I/O Modules in CiA Draft Standard 401, for Drives and Motion Control in CiA Draft Standard 402 and for Encoder in CiA Draft Standard 406. Other profiles are in preparation.

The profiles of a CANopen device are stored in the Object Dictionary (OD) in a defined manner. The Object Dictionary manages the objects using a 16-bit index. This index can be further subdivided with an 8-bit sub-index. All entries are summarized within groups.

For example, the Communication profile is located at index 1000h to 1FFFh.

Certain types of object entries are mandatory; others are optional or manufacturer-specific. The following types of objects are available:

- Domain — a variable number of data
- Deftyp — a definition entry, such as unsigned16
- Defstruct — record type, such as PDO mapping
- Var — an individual variable
- Array — a multiple data field, whereby each individual data field is a simple variable of the same type
- Record — a multiple data field, whereby the data fields are any combination of simple variables

With structured entries, subindex 0 indicates the number of following subindices.

6.3 Communication Profile

The interface between application and CANopen device is clearly defined by a uniform communication profile based on CAL. The CANopen communication protocol defines several methods for transmission and receipt of messages over the CAN bus, including transfer of synchronous and asynchronous messages. Coordinated data exchange across an entire network is possible by means of

© SYS TEC electronic GmbH 2013 L-1046e_7

synchronous message transmission. Synchronous data transfer allows network wide coordinated data exchange. Pre-defined communication objects, i.e. SYNC Objects transmitted on a cyclic time period and Time Stamp objects support synchronous transfers. Asynchronous or event messages may be transmitted at any time and allow a device to immediately notify another device without having to wait for the next synchronous data transfer cycle.

6.4 Service Data Objects

Network management controls communication and device profiles of all networked devices. For this type of access service data objects (SDO) are used. In CANopen devices, all parameters and variables that are accessible via CAN are clearly arranged in the Object Dictionary.

All objects in the Object Dictionary can be read and/or written via SDOs. SDO represent a peer-to-peer communication between networked nodes. This access occurs according to the Multiplexed Domain protocol, whereby the index and subindex of the addressed objects are used as a multiplexor. This protocol is based on handshaking.

Individual parameters are addressed using a 16-bit index and an 8-bit subindex addressing mechanism. In this mode data packages may be larger than 8 bytes using multiple CAN messages. Messages smaller than 5 bytes can be transferred with a transmission acknowledgement. The owner of the Object Dictionary is the server of the domain. Read and write accesses via SDOs are supervised by the CANopen server and are checked for validity.

A variety of access restrictions must be taken into account, such as; *Read only, Write only* and *No PDO mapping.* Error messages provide detailed information on any access conflicts. Service Data Objects (SDOs) are normally used for device configuration such as setting device parameters. They are also used to define the type and format of information communicated using the Process Data Objects.

6.5 Process Data Objects

A Process Data Object (PDO) is a CAN message whose data contents, identifier, inhibit time, transmission type and CMS priority are configurable via entries in the Object Dictionary. PDO format and data content of the message may be fixed or dynamically configured using SDO data transfers. PDOs do not contain any explicit protocol overhead, hence enabling very fast and flexible exchange of data between applications running on each node. Hence, PDO transfers are typically used for high speed, high priority data exchange. Data size in a PDO message is limited to 8 bytes or less. PDO‘s can be transmitted directly from any device on the network simultaneously to any number of other devices. Data exchange across a CANopen network does not require a bus Master. This multicast capability is one of the unique features of CAN and is fully exploited in CANopen.

PDO entries start at index 1400h for receipt objects and at 1800h for transmission objects. CANopen permits cyclic and event-controlled communication. The type of transfer indicates the manner of the reaction to the SYNC message; while the inhibit time is the minimum time that must elapse between two transmissions of the PDO. PDOs reduce the bus load to a minimum, achieve a high information flow-rate and can be accessed via remote frames.

A simple CANopen device usually supports four PDOs. These are initialized with preset identifiers. Additional PDOs can be designated, yet to avoid message collison they may be set invalid (deactivated). This deactivation is configured by setting the MSB (bit 31) in the identifier of the PDO.

The message identifier can be found in the Object Dictionary under the entry for communication parameter in subindex 1. Bit 30 indicates if remote request for this PDO is enabled (bit 30 = 0) or not. Bit 29 configures the CAN frame format, bit 29 = 0 indicates 11-bit identifier.

 © SYS TEC electronic GmbH 2013

Bit	31	30	29	28 – 11	10 - 0
11-bit-ID	0/1	0/1	0	000000000000 000000	11-bit Identifier
29-bit-ID	0/1	0/1	1	29-bit Identifier	

Table 10: COB-Identifier (Communication Target Object Identifier)

The transmission types in subindex 2 can be configured within a range of 0 to 255. The values 0 to 240 define that the transfer of the PDO is in relation to the SYNC message. The value 0 indicates that current input values are only transmitted upon arrival of a SYNC message and if the requested input value has changed. Values between 1 and 240 indicate that the PDO is transmitted upon arrival of a corresponding number of SYNC messages. The values 241 to 251 are reserved. The values 252 and 253 are intended only for remote objects. For value 252, data is updated but not transmitted upon receipt of the SYNC message. The value 253 updates data upon receipt of the remote request. Values 254 and 255 are used for asynchronous PDOs. The release of these asynchronous PDOs is manufacturer or Device Profile-specific.

The inhibit time is stored in multiples of 100 µs as unsigned16-values at subindex 3.

At subindex 4, the priority group for this particular PDO is defined. The priority group is only effective in case DBT services (communication object identifier distribution services) are executed. Depending on the supported subindices, subindex 0 must be set to the applicable value.

 © SYS TEC electronic GmbH 2013

PDO settings must correspond to the I/O profile rules:

- the first transmit and receipt PDO is used for exchange of ***digital*** data;
- the second transmit and receipt PDO is used for exchange of ***analog*** data.

If a CANopen device does not support digital inputs or outputs, it is recommended that the first transmit and receipt PDO remains unused. If a CANopen device does not support analog signals, it is recommended that the second transmit and receipt PDO remains unused.

6.6 PDO-Mapping

A unique mapping entry exists for each communication parameter entry of a PDO. This mapping entry is located in the Object Dictionary 200h above the corresponding communication parameter entry for this PDO. This mapping table corresponds to PDO data contents. The requirement for PDO mapping is the presence of variables in the Object Dictionary that are capable of mapping. For example, digital outputs at index 6200h and digital inputs at index 6000h can be mapped. These values can also be set and read out via SDO. However, in order to use the benefits of the CAN bus, the variables of a CANopen device are put in PDOs.

The mapping of variables is organized as follows:
All mapping entries are 4 bytes in size. The number of objects to be mapped is written to subindex 0. Each following subindex contains a reference to the index and subindex of variables and their length stored in “Bit“. For example:

60000108h

- reference to index 6000
    - subindex 1
        - length 8 bits

In this example the value of the digital input is indicated by the first byte of a transmit PDO. For most CANopen devices, mapping occurs

© SYS TEC electronic GmbH 2013 L-1046e_7

with a granularity of eight (8). This means that a maximum of eight entries per byte is possible for a mapping table.

In special cases mapping of bit objects can be supported. It is also advisable to sometimes exclude areas from mapping. For example, a CANopen device might evaluate only the fifth byte of a PDO. In this case, 2 unsigned16 dummy objects are inserted in the mapping identity, if supported by the CANopen device. A mapping table can be used to appropriately configure communication parameters to encode a PDO for transmission or to decode a received PDO.

PDO Mapping Example

All network variables can be transferred by PDOs, which can transmit a maximum of 8 bytes of information. The allocation of variables to PDOs is defined by mapping tables. These variables are addressable via the Object Dictionary. Reading and writing of entries to the Object Dictionary occurs by means of Service Data Objects (SDO), which are used to configure the network by means of a special configuration tool.

This process is illustrated below in *Table 11*. Inputs 2 and 3 of device A are to be transferred to the outputs 1 and 3 of device B. Both devices support complete mapping.

 © SYS TEC electronic GmbH 2013

Device A:

1000H	Device Type
......	
6000H,1	Input 1, 8 Bit
6000H,2	Input 2, 8 Bit
6000H,3	Input 3, 8 Bit
....	

Transmit PDO Mapping Parameter

1A00H,0	# of Entries	2
1A00H,1	1.Map Object	60000208H
1A00H,2	2.Map Object	60000308H

Transmit PDO Communication Parameter:

1800H,0	# of Entries	2
1800H,1	COB-ID	501
1800H,2	Trans.Type	255
....		

Resulting PDO:

COB-ID	DATA	
501	Output 1	Output 3

Table 11: PDO Mapping Example

Transmit and receive PDOs utilize the same CAN identifier 501. Thus device B automatically receives the PDO transmitted by device A. The recipient, device B, interprets the data in accordance with its mapping scheme; it passes the first byte at output 1 and the second byte at output 3. These correspond to inputs 2 and 3, respectively of the transmitting device A.

© SYS TEC electronic GmbH 2013

6.7 Error Handling

Each node in the network is able to signal error states as far as they are detected by the hardware and software. Error Handling is enabled by Emergency Objects. Internal fatal error states are encoded in error codes and sent only once to the other nodes. If other errors occur, the node remains in error state and transmits a new Emergency Object. If the error is recovered, the node then transmits an error message with the code *No error*. The Emergency message consists of 8 bytes, whereby the first and second bytes contain additional information that is found in the device profiles. The third byte contains the contents of the error register; while the remaining five bytes contain manufacturer-specific information. The Emergency Error code is stored in object [1003h], the *Pre-Defined Error Field*. This creates an error log that chronologically sorts errors. The oldest error is situated at the highest subindex.

Byte	0	1	2	3	4	5	6	7
Content	Emergency Error Code		Error Register, Object [1001]	Manufacturer Specific Error Field				

Table 12: Emergency-Message Contents

6.8 Network Services

In addition to services enabling configuration and data exchange, various CAN network services also support monitoring of networked devices. NMT (network management) services require a node in the network that assumes the functions of the NMT-Masters. The NMT-Master services include initialization of NMT-Slaves, distribution of the identifiers, the node monitoring and network booting.

6.8.1 Life-Guarding

Optional node monitoring is achieved by "*Life-Guarding*". The NMT-Master periodically transmits a Lifeguard message to the Slave. The Slave responds to the Lifeguard message with a return message indicating its present status and a bit that toggles between two messages. Should the Slave not respond or indicate an unexpected status, the NMT-Master application is informed by means of a status message. Moreover, the Slave can detect failure of the Master. *Life-Guarding* is started with the transmission of the initial message from the Master.

6.8.2 Heartbeat

Similar to Lifeguarding, the Heartbeat function is an additional network supervisory service. But unlike the Lifeguarding, the Heartbeat does not require a NMT-Master. Only CANopen Slaves are able to function as Heartbeat Producer and Consumer because they provide an Object Dictionary in order to store the Heartbeat times.

6.8.3 Heartbeat Producer

The Heartbeat Producer cyclically sends a Heartbeat message. The configured Producer Heartbeat time (16-bit – value in ms), located at index 1017h, will be used as an interval time. If this interval time expires, a message with the following contents will be sent:

Byte	0	1...7
Content	Producer State	reserved

Table 13: Heartbeat Message Structure

The COB-ID that is used is 0700h + the node number.
The Hearbeat Producer gives its status, which can be any of the following values, in the first byte of the message:

00h BOOTUP
04h STOPPED
05h OPERATIONAL
7Fh PRE-OPERATIONAL

The Heartbeat Producer is deactivated when the producer Heartbeat time is set to 0.

© SYS TEC electronic GmbH 2013

6.8.4 Heartbeat Consumer

The Heartbeat Consumer analyzes Heartbeat messages sent from the producer. In order to monitor the Producer, the Consumer requires every producers' node number, as well as the consumer Heartbeat time.

For every monitored Producer, there is a corresponding sub-entry that has the following contents:

	MSB		**LSB**
Bit	31-24	23-16	15-0
Value	00h	Node-ID	Consumer Heartbeat Time

Table 14: Structure of a Consumer Heartbeat Time Entry

The Consumer is activated when a Heartbeat message has been received and a corresponding entry is configured in the OD. If one of the activated Heartbeat times expires during an active Heartbeat consumer without receipt of a corresponding Heartbeat message, then the consumer for this producer is deactivated.

The Heartbeat consumer is completely deactivated when the consumer Heartbeat time is given a value of 0.

6.9 Network Boot-Up

The NMT-Master is responsible for booting of the network. The boot procedure takes place over several steps. According to the type of networked CANopen device, the identifier defaults to pre-defined values (for minimum CANopen devices) or is configured via DBT services. The pre-defined configuration for the identifier values include Emergency Objects, PDOs and SDOs. These are calculated according to node addresses, which can be located between 1 and 128 and are added to a base identifier that determines the function of an object.

Bit	10			7	6						0
COB-Identifier											
	Function Code				Device-ID						

Table 15: Calculation of the COB-Identifier from the Node Addresses

This base identifier is determined as follows:

Object	Resulting COB-ID [hex]	Resulting COB-ID [decimal]	Communication Parameter at Index
EMERGENCY	80h + Device-ID	129 – 255	
PDO1 (tx)	180h + Device-ID	385 – 511	1800h
PDO1 (rx)	200h + Device-ID	513 – 639	1400h
PDO2 (tx)	280h + Device-ID	641 – 767	1801h
PDO2 (rx)	300h + Device-ID	769 – 895	1401h
PDO3 (tx)	380h + Device-ID	896 – 1022	1802h
PDO3 (rx)	400h + Device-ID	1025 – 1151	1402h
SDO (tx)	580h + Device-ID	1409 – 1535	
SDO (rx)	600h + Device-ID	1537 – 1663	
Nodeguard	700h + Device-ID	1793 – 1919	(100Eh)

Table 16: Base Identifier

Configuration data can be loaded on Slave devices via the pre-defined SDO. PDOs can be transmitted after nodes are set from *Pre_Operational* to *Operational* state by the NMT service *Start_Remote_Node*. Minimum CANopen devices also support the *Stop_Remote_Node*, *Enter_Pre-Operational_State*, *Reset_Node*, *Reset_Communication* services. As indicated in Figure 3, networked nodes automatically enter *Pre_Operational* following boot-up and initialization. The *Reset_Node* service completely resets target nodes. *Reset_Communication* resets communication parameters.

© SYS TEC electronic GmbH 2013 L-1046e_7

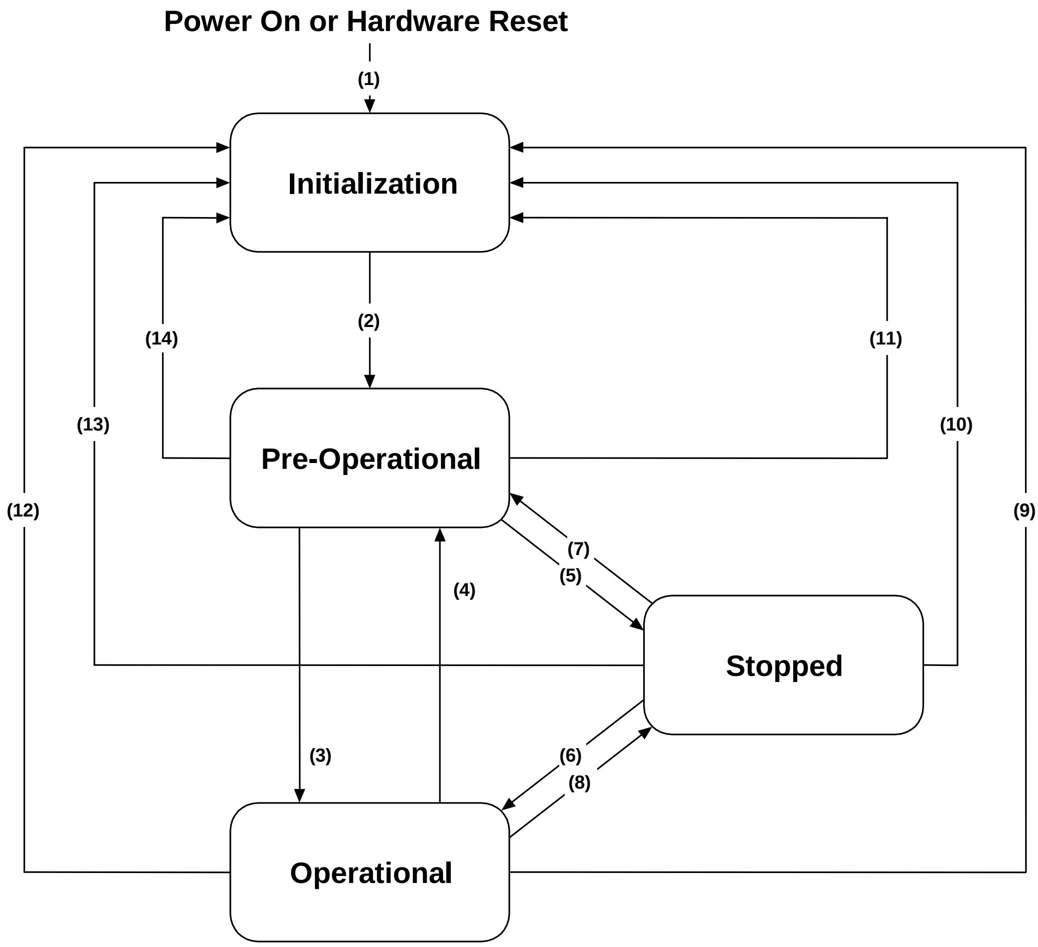

Figure 10: State Diagram of a CANopen Device

State transition	Action required
(1)	following "Power On", automatically switches into "Initialization" state
(2)	"Initialization" finished, automatically switches into "Pre-Operational" state
(3),(6)	NMT service "Start_Remote_Node"
(4),(7)	NMT service "Enter_Pre-Operational_State"
(5),(8)	NMT service "Stop_Remote_Node"
(9),(10),(11)	NMT service "Reset_Node"
(12),(13),(14)	NMT service "Reset_Communication"

Table 17: Description of State Flow Diagram Symbols

For networked devices operating in a network with or without DBT capabilities, it is necessary to reserve the identifier for “minimum devices“ in the database of the DBT-Master.

Extended Boot-up is based on CAL specifications. The device states *Pre-Operational* and *Initializing* have been implemented in addition.

6.10 Object Dictionary Entries

Beside the parameters for the PDOs, a number of additional entries in the Object Dictionary belong to the data that specify a CANopen device. The communication profile contains such information as:

- the device type at index [1000H];
- the error register at index [1001H];
- the Pre-Defined Error Field at [1003H];
- the identifier of the SYNC message at [1005H];
- the device name at [1008H],
- the hardware and software version of the manufacturer at [1009] and [100AH];
- the node address at [100BH];
- the parameter Guard-Time at [100CH] and
- the parameter Life-Time-Factor at [100DH].

In the device type, information about the implemented device profile and the capabilities of the device is encoded. The error register gives information about internal errors of the device; the pre-defined error field provides an error log. In case the Guard-Time and Life-Time-Factor are unequal to Zero, the multipied values result in the Life Time of the CANopen device for the node monitoring protocol.

 © SYS TEC electronic GmbH 2013

6.11 Input/Output Assignment to Object Dictionary Entries

The CANopen IO-C12 allows an easy configuration for a specific CANopen application. The fixed number of inputs and outputs on the CANopen IO-C12 makes easy configuration of Process Data Objects (PDOs) possible. Both digital and analog inputs, as well as the digital and analog outputs, are configured in accordance with CiA standards. Configuration of Object Dictionary Input/Output entries for the CANopen IO-C12**,** according to data type, is shown in *Table 18*.

Data Type	**Index / Subindex**	**Size**
Digital Input		
DI0 ... DI7	6000H / 1	BYTE
DI8 ... DI15	6000H / 2	BYTE
DI16 ... DI23	6000H / 3	BYTE
DI24 ... DI26	6000H / 4	BYTE
Digital Output		
DO0 ... DO7	6200H / 1	BYTE
DO8 ... DO15	6200H / 2	BYTE
REL0 ... REL3	6200H / 3	BYTE
Analog Input		
AI0	6401H / 1	WORD
AI1	6401H / 2	WORD
AI2	6401H / 3	WORD
AI3	6401H / 4	WORD
Analog Output		
AO0	6410H / 1	WORD
AO1	6410H / 2	WORD

Table 18: Object Dictionary Input/Output Entries

Note: After boot-up of the CANopen IO-C12, objects can be accessed via SDOs. If the node is in *Operational* state**,** objects can be accessed via PDOs. The default mapping parameters applies for Object Dictionary Input/Output entries. Any modification of mapping parameters can be done via SDO with the help of a network configuration tool.

7 CANopen IO-C12 Operation

7.1 CANopen State Transitions

The structure of messages that changes the state of a CANopen node is as follows:

11-bit CAN Identifier		2 Byte Data	
0		*cs*	*NODE_ID*

Node_ID Node address; Node_ID = 0 to address all devices (Broadcast)

cs Command

Table 19 summarizes all NMT-Master messages used for status control:

Command (cs)	Description	Function	State after Execution
1 (01h)	Start_Remote_ Node	Starts the CANopen device and PDO transmission, activates outputs	OPERATIONAL
2 (02h)	Stop_Remote_ Node	Stops PDO transmission, renders outputs in error state	STOPPED or PREPARED
128 (80h)	Enter_Pre_ Operational_ State	Stops PDO transmission, SDO remains active	PRE-OPERATIONAL
129 (81h)	Reset_Node	Executes a system Reset; Initial Start-up, resets all settings to default values	PRE-OPERATIONAL
130 (82h)	Reset_ Communication	Resets all communication parameters to deault values	PRE-OPERATIONAL

Table 19: NMT-Master Messages for Status Control

7.2 Power On

After "Power-On", the CANopen IO-C12 executes required initialization routines and switches into *Pre_Operational* state.

7.3 PRE-OPERATIONAL

Process Data Objects (PDOs) are not active in *Pre_Operational* state. The default identifier for Service Data Objects (SDOs) is available and all necessary network configurations can be executed via SDO. At the end of the configuration process, the CANopen device can be rendered into *Operational* state. This can be done by the network Master or by the user with the help of a network configuration tool.

7.4 OPERATIONAL

All Process Data Objects (PDOs) can be exchanged in *Operational* state. Access via SDO is also possible.

7.5 STOPPED

Network communication is suspended in state *STOPPED*. This does not affect the Node-Guarding and the "Heartbeat", if this was enabled before. This state can be used to render the application into a "Safety State". In *STOPPED* state PDO, SDO, SYNC and Emergency communication are **NOT** functioning. Leaving this state is only possible with a NMT message.

7.6 Restart Following Reset / Power-On

Each Reset of the CANopen IO-C12 transmits an Emergency message without data contents. Temporary operational failure of the CANopen IO-C12 and subsequent power-up of the device are detected without Node Guarding (*refer to Section 4.6 Node-Guarding*), as the sending device can be determined by the message identifier.

© SYS TEC electronic GmbH 2013

The CANopen IO-C12 distinguishes between *“Load”*_Start and *“Save”*_ Start. *“Load”*_Start is necessary:

- for initial operation of the CANopen IO-C12 after its delivery
- if the device parameters (Object Dictionary entries in RAM) should be overwritten by default values

With *“Load”*_Start, all default CANopen IO-C12 Object Dictionary entries are copied to RAM after Reset/Power On (manufacturer default values).

The string “save” must be written to object [1010H] at subindex 1 in order to carry out the *“Save”*_ Start routine. With *“Save”*_ Start all Object Dictionary entries are copied from nonvolatile memory to RAM after a Reset/Power-On using the saved user-specific values. If the bus Master or the user, by means of a network configuration tool, modifies Object Dictionary entries, then the modifications are only active as of the next RESTART if *“Save”* is written to object [1010H] in subindex 1. This means that only the stored values are valid after the Reset/Power-On of the CANopen IO-C12. These values are stored then in the nonvolatile memory and do not get lost in the event of power-down. A*“Save”*_ Start can take up some seconds, because of the copy operations.

All device parameters can be stored in the nonvolatile memory using object [1010H] in subindex 1. In order to prevent unintended storage of parameters in the E^2PROM device, a special *“Save”* signature must be written to subindex 1. This 32-bit signature (in hex format) appears as follows:

MSB LSB

‘e’	‘v’	‘a’	‘s’
65h	76h	61h	73h

All device parameters can be reset to manufacturer default values according to DS301 or DS401 standards via the object [1011H] in subindex 1. In order to prevent an unintended reset following a store instruction with the *"Save"* signature, the *"Load"* signature must be written to subindex 1. This 32-bit signature (in hex format) appears as follows:

MSB LSB

'd'	'a'	'o'	'l'
64h	61h	6fh	6ch

In order to set the default values, a Reset/Power-On must be subsequently executed.

7.7 Functions of the digital Inputs

There are multiple trigger conditions selectable for the digital outputs. These are defined in the Object Dictionary at Index 6005H, 6006H, 6007H and 6008H.
It is within the userd responsibility to make sure **only one** trigger condition is activated per input. See *Table 18* for the appropriate OD enties of the digital inputs. The present status of all digital inputs are displayed by the LEDs located on the front-cover.

7.7.1 Global interrupt enabling of digital inputs 6005H

The OD entry 6005H enables PDO transmission for digital inputs in asynchronous transmission type.
The default value is TRUE (1).

7.7.2 Interrupt Mask rising and falling edge - 6006H

This is the default setting for digital outputs. The state of the input is transmitted over CAN at every change.

The default value for all digital inputs is 1.

 © SYS TEC electronic GmbH 2013 L-1046e_7

7.7.3 Interrupt Mask rising edge - 6007H

This is an optional setting for digital inputs. The state of the input is transmitted over CAN at a change from "0" to "1” only.

If an input is configured for this mask no other input must be activated in Index 6006H or 6008H.

The default value for all digital inputs is 0.

7.7.4 Interrupt Mask falling edge - 6008H

This is an optional setting for digital inputs. The state of the input is transmitted over CAN at a change from "1" to "0” only.

If an input is configured for this mask no other input must be activated in Index 6006H or 6007H.

The default value for all digital inputs is 0.

7.8 Functions of the digital outputs

See *Table 18* for the appropriate OD enties of the digital outputs. The present status of all digital outputs as well as the relay REL0..REL3 are displayed by the LEDs located on the front-cover. The error behaviour is desbribed in section *8.1*.

7.9 Analog Input Operation

7.9.1 Handling Analog Values

This section provides general information on data storage of analog values in a CANopen frame.

The CANopen Standard DS401 defines that all analog values have to be stored as 32-bit value aligned left with a sign bit. On the CANopen IO-C12 all A/D-conversion values are stored with 10-bit data. Consequently, for each analog channel, two data bytes must be transmitted.

These data bytes are stored and transmitted on the CAN bus as shown in *Table 20*.

Byte 2								Byte 1							
Sign	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+/-	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0	0	0	0	0

Table 20: Storage of Analog Values

On the CAN bus, first byte 1 and then byte 2, is transmitted.

7.9.2 Formula for Calculating the Analog Input Value

The formula listed below is used to calculate a voltage value of an analog input from the A/D-conversion result:

$$AIn_{/V} = \frac{result\ A/D\ conversion_{/hex} \bullet voltage\ range_{/V}}{2^{resolution\ ADC}}$$

The following example will explain the use of this formula in more detail:

A/D-value in OD: = 41A0H (16800dez)
voltage range = 10.107V
logical resolution of OD entry = 15 Bit (signed)
analog input value (AI) = 5.18V

AI = (0x41A0 • 10.107V) / 2^{15} =5.12V
To get the least quantisation of the ADC the actual used internal resolution is needed.
Result A/D conversion: = 01H
Voltage range: = 10.107V
Internal ADC resolution = 10 Bit
Least possible resolution: = 9.87mV/Digit
This corresponses to the value 20H in Object Dictionary transmitted via CAN.

© SYS TEC electronic GmbH 2013 L-1046e_7

7.9.3 Selecting the Interrupt Trigger

This object entry determines which event can release an interrupt. For this purpose the object [6421] "Interrupt_Trigger_Selection" is available. If the "Global_Interrupt_Enable" [6423] is activated, the release of an interrupt transmits the TX-PDO for analog inputs. A specific subindex is available for each analog input channel. This allows precise configuration of the interrupt event for each channel.

The following values are available:

Bit Number	Interrupt Trigger
0	Upper limiting value exceeded
1	Lower limiting value exceeded
2	Input value fluctuates more than *DELTA [6426]*
3	*Not supported!*
4	*Not supported!*
5 to 7	Reserved

Table 21: Interrupt Trigger Bits

Example:
6421,1 = 04h means: the first analog input must fluctuate by more than *DELTA* in order to send the PDO.

Note:
The default values for all analog inputs are set to 07h.

7.9.4 Interrupt Source

This object entry stores which analog input caused the interrupt. The object [6422] "Analog_Input_Interrupt_Source" is available for this purpose. Every single bit refers to the corresponding analog input channel. These bits will be reset automatically if the entry has been read by a SDO or the object entry was transmitted with a PDO.

The following convention is used:
"1" : Channel caused an interrupt,
"0" : Channel caused no interrupt.

Example:
6422,1 = 01h means: analog input channel 0 caused an interrupt.

7.9.5 Interupt Enable

All interrupts can be enabled or disabled using the object entry [6423] "Analog_Input_Global_Interrupt_Enable". The default value is "0", indicating interrupt execution is disabled. To enable the interrupt execution, the value "1" must be written to the object entry (*also refer to section 7.9.3*).

7.9.6 Interrupt Upper and Lower Limit

An interrupt is released, if the analog input value is higher or lower than the specified limiting value in the applicable subindex. The upper limit is sepcified in object [6424], the lower limit in [6425]. To release an interrupt, the OD entry [6423] must be set to "1".

Each analog input value will be transmitted as long the trigger condition is given. This assumes that no other trigger condition, such as the Delta Function, is enabled. The limit values must be specified as 32-bit value aligned left.

For this purpose, the objects:
- [6424] "Analog_Input_Interrupt_Upper_Limit_Integer" and
- [6425] "Analog_Input_Interrupt_Lower_Limit_Integer"

are available.

Note:
The default value in both entries for all analog inputs is "0".

Example:
6423 = 1h, 6421,1 = 05h and **6424,1 = 2000 0000h**:
The analog input #1 releases an interrupt if the value exceeds the limit of 2000h, and then the value fluctuates by more than specified in the Delta function (see following section).

© SYS TEC electronic GmbH 2013

7.9.7 Delta Function

The delta function allows configuration of the extent to which an analog input value can fluctuate since the most recent transmission. Only if the fluctuation on the analog input exceeds the value specified in the delta function transmission of the corresponding PDO on the CAN bus is initiated. This configuration can be done using the object [6426] *Analog_Input_Interrupt_Delta*. Entries specify the number of digits in the conversion result that are allowed to fluctuate. The default value for all four analog inputs is 5. This means that the A/D-conversion result may change by up to 5 digits before a PDO is transmitted. The value must be specified as aligned left and assumes 10-bit resolution.

7.10 Functions of the Analog Outputs

The CANopen C12 module features two analog outputs delivering a voltage range of 0..10V. The conversion results are represented in the Object Dictionary as SINT (signed integer

The following example will explain the use of this formula in more detail:

A/D-value in OD:	= 500H (1280dez)
voltage range	= 10.35V
logical resolution of OD entry	= 15 Bit (0.316mV/digit)
analog ouput value (AI)	= 0.40V

The internal resolution of the analog outputs is 8-Bit. The smallest possible quantization is 40mV/digit. For conversation the lower 7-Bit of the OD value are ignored. For example: the OD value range 500H until 57FH delivers the same output voltage of 0.40V.
The present status of each analog output is indicated by two LED on the front cover. The intensity of the LED corresponds the output voltage.

 © SYS TEC electronic GmbH 2013

7.11 Functions of the PWM outputs

The CANopen C12 has two intergrated, low-side switching, short-circuit proof PWM[1]-outputs. The maximum load current is 0.5A each output for inductive, capacitive and resistive load. There is an external H-Bridge module available to extend load current to up to 5A.
The PWM outputs are galvanic isolated from CPU core. The outputs hare connected to common ground of VIO. The outputs are protected against wrong polarity.

The OD has two parameters for each PWM output used for configuration.

The first parameter on OD index [6500H] describes the duty-cycle of the PWM signal. The parameter is given in percent (0..100%) and is represented as WORD value in the OD.
That means: 0%=0H, 100%=FFFFH.

The default setting for the duty cycle of each PWM output is 0%/0H.

The second parameter on OD index [6510H] describes the output frequency. The value is given in µs. For example the value of 1000 will set the frequency to 1kHz.

The default setting for frequency is set to 1000 for each PWM output.

7.11.1 Special functionality using phyPS-409-KSM01 modules

The module firmware includes these special OD entry's:
index [6520], subindex 0 to 2
index [6530], subindex 0 to 2

Together with the analog input 0 and 1 these entry's influence the PWM output signals. This functionality can be used to switch off the PWM output by over current.

[1] PWM: **P**ulse **W**idth **M**odulation, **P**ulse**W**eiten**M**odulation

© SYS TEC electronic GmbH 2013 L-1046e_7

The entry Motor Current Upper Limit, subindex 1 stores the upper limit for the analog value. If the value reading on AI0 higher then the upper limit the PWM output 0 duty-cycle is set to 0%. That means the PWM output is switched off. The PWM output can be reactivated by writing a new duty-cycle.

This functionality can be switch off by setting the OD entry to 0H.

This functionality is also available for the second PWM output, therefore the upper limit is stored under index [6520], subindex 2. AI1 is used for the analog input.

The OD entry [6530] Motor ON Delay switch off the current comparison during the power phase on off a DC motor. The value is given in µs. For example the value of 500 will set the ON Delay to 500µs. This delay time always starts by an changing of the OD entry [6500]

7.12 Emergency Message

In the event of an error, the status of the CANopen IO-C12 is transmitted via a high-priority Emergency Message. These messages consist of 8 data bytes and contain error information. The Emergency Message is transferred as soon as one of the specified errors occurs. A specific Error Message is only transmitted once, even if the recent error is not resolved for a longer period of time. If all error causes are eliminated, then an Error Message with contents “0“ (error eliminated) is transmitted. The structure of the 8-byte Emergency Message is depicted below:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Error Code		Error-Register [1001]	Manufacturer-specific Error Code				

Table 22: Emergency Message

7.12.1 Error Code

The Error Code (byte field 0+1, LSB, MSB) indicates whether an error is present or whether the error has already been eliminated (no error). The following error codes are valid:

- 0000h: no error
- 1000h: global error
- 3100H: power supply error, Power Fail
- 5000H: hardware reset occur (watchdog or reset push button)
- 6100H: software reset occur
- 8110H: CAN-messages lost, (busload to high)
- 8120H: Error passiv Mode
- 8130h: Lifeguard or Heartbeat Error
  In case of a Heartbeat Consumer error, the node ID of the failing node is transmitted in the manufacturer-specific error code field.

7.12.2 Error Register

The Error Register (byte field 2) can contain the following values:

- 81h: Occurrence of a manufacturer-specific error
- 11H: CAN communication error
- 01h: Occurrence of a common error
- 00h: Error has been eliminated - error reset

7.13 Status LEDs

The present state of the CANopen IO-C12 module is displayed through the both status LEDs RUN and ERROR at runtime.
The function of the LEDs are defined according to the **CiA standard DR303-3 V1.0**. Please refer to the standard to get further information.

7.13.1 RUN LED

The green RUN LED indicates the general state of the module (according to the CANopen network state-diagram).
Table 23 describes the possible LED modes and their meaning.

© SYS TEC electronic GmbH 2013

RUN Led	State	Description
Alternate blinking with the Error LED	LSS Access	There is a LSS Service running.
Single Flash	STOPPED	The module is in state STOPPED
Blinking	PRE-OPERATIONAL	The module is in state PRE-OPERATIONAL
On	OPERATIONAL	The module is in state Zustand OPERATIONAL
Synchronous fast blinking cycle togeter with the ERROR Led	Configuration error[1]	There is an invalid configuration selected on the DIP-Switch or HEX-encoding switch.

Table 23 States of the RUN LED

7.13.2 Error Led

The red Error LED indicates the error states of the CANopen IO-C12 unit with the following possible modes:

Error Led	State	Description
OFF	No error	The module operates within nominal parameters.
Single Flash	Warning Limit reached	The CAN controller internal warning limit was reached. (to many error-frames on CAN bus).
Alternate blinking with Run Led	LSS Access	There is a LSS Service running.
Double Flash	Error Control Event	An error in Lifeguard, Nodeguard or Heartbeat was detected.
On	Bus Off	The CAN controller is in state "Bus Off".
Synchronous short blinking cycle with Run Led	Configuration error [1]	There is an invalid configuration selected on the DIP-Switch.

Table 24: States of the ERROR LED

[1] This state is a SYS TEC specific Add-On and not defined in the DR303-3 standard

 © SYS TEC electronic GmbH 2013

From software version V1.16 and later there are an extendet error and configuration error indication via LED’s „Digital Input“ 0 to 7. In this case the LED’s show not the state of the digital inputs but the error state, see the following table.

LED	Error
DI 0	Configuration error, bit rate not valid
DI 1	Configuration error, node id not valid
DI 2	serial number not valid, please sent the device to SYS TEC for inspection
DI 3	EEPROM CRC error, please erase the EEPROM-data (see 2.5.3), note, you should configure the device new or sent the device to SYS TEC for inspection
DI 4	product code invalid, sent the device to SYS TEC for inspection
DI 5	reserved
DI 6	RAM- error, sent the device to SYS TEC for inspection
DI 7	CANopen internal error, sent the device to SYS TEC for inspection

Table 25 extendet error indication via LED DI0 to DI7

7.14 Hardware versions

There are 2 different hardware versions of the C12 device.
To define the versions, read the index 1009H “Manufacturer Hardware Version” in the object dictionary.

“4121.1 / 4103.2-1” CAN-transceiver PCA82C251

“4121.1 / 4103.2” or
“4121.1 / 4103.2-0” CAN-transceiver TJA1050 (must not use at CAN-bit rats:
1MBit/s and <100kBit/s)

7.15 Manufacturing data

In the Manufacturer-specific profile area of the object dictionary there are entries for production data, locatet in the index 2500H.
This entries are read only.

 © SYS TEC electronic GmbH 2013 L-1046e_7

Index 2500H, Subindex 2, unsigned32, read only, date of manufacturing
Index 2500H, Subindex 3, unsigned32, read only, date of calibration.

Encoding rules for this date entries:
example:
17.02.2009 is encodes as 017022009H.

© SYS TEC electronic GmbH 2013

8 Operations in the Event of Errors

8.1 State of the CANopen IO-C12 in the Event of Errors

The object dictionary entry "Error Behaviour" at index [1029] can be used to define which state the CANopen IO-C12 should transfer to in case of an error.

The following entries at index 1029 subindex 1 (communication error) are possible:

0: change state to PRE-OPERATIONAL (default)
1: do not change state
2: change state to STOPPED

These settings over all possible error sources described in *sections 7.12.1* . The entries Output Error (subindex 2) and Input Error (subindex 3) are not supported.

8.2 Output Handling in the Event of Errors

The user can determine how each output is supposed to behave in the event of an error.

On digital outputs (DO0..DO16 and REL0,REL1), error handling can be pre-defined via the objects:

[6206H] (*"Error_Mode_Output_8-Bit"*) and
[6207H] (*"Error_Value_Output_8-Bit"*).

On analog outputs (AO0..AO1), error handling can be pre-defined via the objects :

[6443H] *("Analogue Error Output Mode")* and
[6444H] *("Analogue Output Error Value Integer")*

These entries can be configured by means of a network configuration tool. In the default configuration, the outputs do not change their states in the event of an error.

 © SYS TEC electronic GmbH 2013

A "1" at the Bitposition of the corresponding output in objekt [6206H] and [6443H] causes a write operation of the values in object [6207H] and [6444H] to the corresponding outputs.

Example for digital outputs:
Digital outputs REL0..REL3

Index	**Subindex**	**REL3**	**REL2**	**REL1**	**REL0**	**Description**
6206	3	0	0	1	1	Error Mode Output 8-bit
6207	3	X	X	0	1	Error Value Output 8-b

Table 26: Example for Error Handling on relay outputs

In the event of an error, the digital output REL0 is set to 1 while REL1 is set to 0. The status of the outputs REL2 and REL3 remain unchanged.

analog Outputs AO0 and AO1

Index	**Sub-Index**	**AO 0**	
6443H	1	1	Analogue Output Error Mode AO 0
6443H	2	0	Analogue Output Error Mode AO 1
6444H	1	0500H	Analogue Output Error Value AO 0
6444H	2	0H	Analogue Output Error Value AO 1

Table 27: : Example for Error Handling on analog outputs

In the event of an error, the analog output AO0 is set to 0.39V while AO1 remains unchanged.

8.3 Changing from Error State to Normal Operation

In the event of an error, the outputs retain their active values until overwritten (by means of PDO/SDO) by new output values. This requires that the error, such as “Bus Off” or “Life-Guarding” error, is eliminated and the CANopen IO-C12 be switched into *Operational* state by a Master *“Start_Remote_Node”* message.

© SYS TEC electronic GmbH 2013 L-1046e_7

9 CANopen IO-C12 Object Dictionary

Index [hex]	**Objekt**	**Name**	**Data Type**
1000	Var	Device typ	Unsigned32
1001	Var	Error register	Unsigned8
1003	Array	Pre-defined erroro field	Unsigned32
1005	Var	Identifier SYNC	Unsigned32
1007	Var	SYNC window length	Unsigned32
1008	Var	Manufacturer device name	String
1009	Var	Manufacturer Hardware Version	String
100A	Var	Manufacturer Software Version	String
100C	Var	Guard Time	Unsigned16
100D	Var	Life Time Factor	Unsigned8
1010	Array	Store parameter	Unsigned32
1011	Array	Restore parameter	Unsigned32
1014	Var	Identifier Emergency	Unsigned32
1016	Array	Consumer heartbeat time	Unsigned32
1017	Var	Producer heartbeat time	Unsigned16
1018	Record	Identity object	Identity
1029	Array	Error behaviour	Unsigned8
1400	Record	Receive PDO 0 Communication Parameter	PDOComPar
1401	Record	Receive PDO 1 Communication Parameter	PDOComPar
1402	Record	Receive PDO 2 Communication Parameter	PDOComPar
1600	Record	Receive PDO 0 Mapping Parameter	PDOMapping
1601	Record	Receive PDO 1 Mapping Parameter	PDOMapping
1602	Record	Receive PDO 2 Mapping Parameter	PDOMapping
1800	Record	Transmit PDO 0 Communication Parameter	PDOComPar
1801	Record	Transmit PDO 1 Communication Parameter	PDOComPar
1802	Record	Transmit PDO 2 Communication Parameter	PDOComPar
1A00	Record	Transmit PDO 0 Mapping Parameter	PDOMapping
1A01	Record	Transmit PDO 1 Mapping Parameter	PDOMapping
1A02	Record	Transmit PDO 2 Mapping Parameter	PDOMapping
2500	Record	Manufacturing data, for production only	
6000	Array	Read input 8 bit	Unsigned8
6005	Var	Globaler Interrupt Enable Digital 8 Bit	Unsigned8
6006	Array	Interrupt Mask Any Change 8 Bit	Unsigned8
6007	Array	Interrupt Mask Low-to-High 8 Bit	Unsigned8
6008	Array	Interrupt Mask High-to-Low 8 Bit	Unsigned8
6200	Array	Write Output 8 Bit	Unsigned8
6206	Array	Error Mode Output 8 Bit	Unsigned8
6207	Array	Error Value Output 8 Bit	Unsigned8
6401	Array	Read Analogue Input 16 Bit	Integer16
6411	Array	Write Analogue Output 8 Bit	Integer16

 © SYS TEC electronic GmbH 2013

6421	Array	Analogue Input Interrupt Trigger Selection	Unsigned8
6422	Array	Analogue Input Interrupt Source	Unsigned32
6423	Var	Analogue Input Global Interrupt Enable	Unsigned8
6424	Array	Analogue Input Interrupt Upper Limit Integer	Integer32
6425	Array	Analogue Input Interrupt Lower Limit Integer	Integer32
6426	Array	Analogue Input Interrupt Delta Unsigned	Unsigned32
6443	Array	Analogue Output Error Mode	Unsigned8
6444	Array	Analogue Output Error Value Integer	Integer32
6500	Array	PWM Output Pulse	Unsigned16
6510	Array	PWM Output Periode	Unsigned16
6520	Array	Motor Current Upper Limit[1]	Unsigned32
6530	Array	Motor ON Delay[2]	Unsigned32

Table 28 : Object Dictionary of the CANopen IO-C12 module

[1] only available for phyPS-409-KSm01
[2] only available for phyPS-409-KSM01

© SYS TEC electronic GmbH 2013 L-1046e_7

10 Firmware-Update

For firmware-update the freeware tool Fujitsu Flash "MCU Programmer" is used. This program has to be installed on PC, used for doing the firmware-update.

To do a firmware-update on CANopen IO-C12, the following steps have to be processed:

- Connection of CANopen IO-C12 to PC
  Connect the PC by a serial interface (RS232) to the CANopen IO-C12 X101 (RJ11-connector, see 2.2, 2.3 and 2.4).
  A suitable cable can be ordered at SYSTEC electronic GmbH.

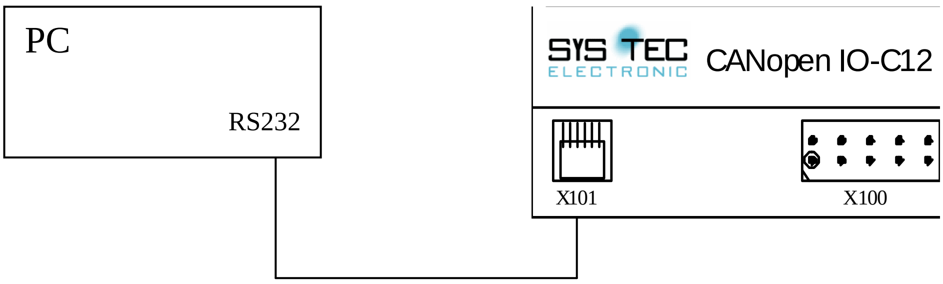

Figure 11: Firmware-Update – Connection to PC

- Configuration of flash-tool „Fujitsu Flash MCU Programmer"
  1.) Configuration of serial port:
  By button "*Set Environment*" in area "*Option*" the COM interface can be selected, what is used for firmware-update:

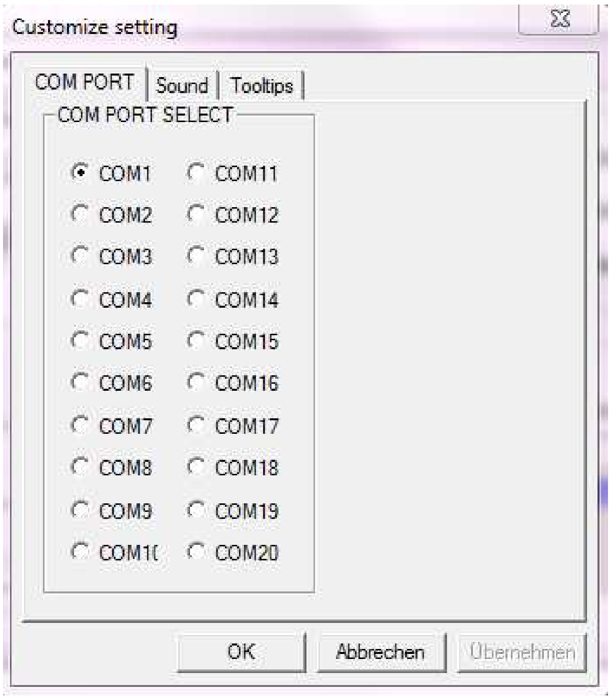

Figure 12: Firmware-Update – Selection of COM-interface

2.) Configuration of target:
Selection of microcontroller *MB90F543/G/GS*
Selection of crystal frequency *4MHz*

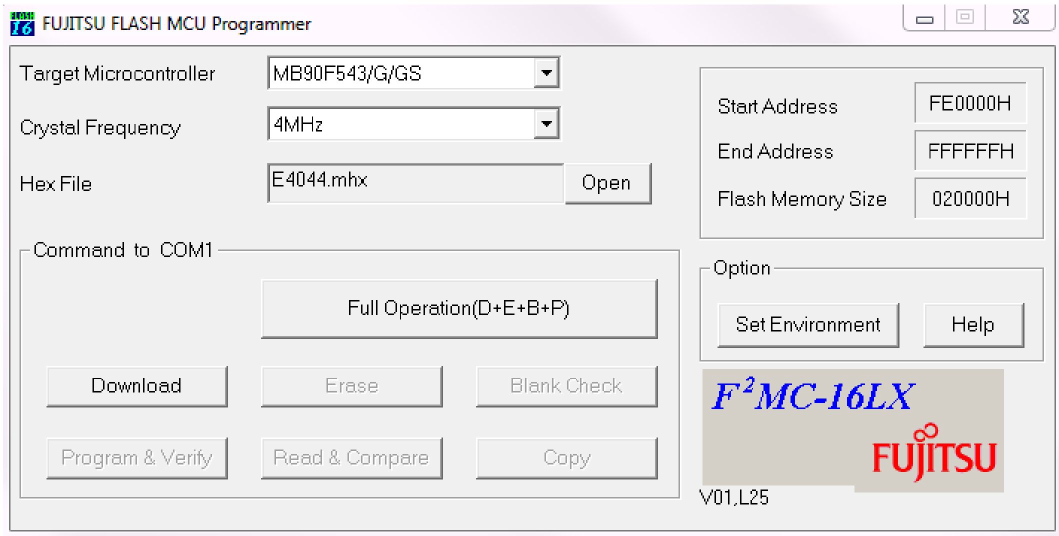

Figure 13: Firmware-Update – Target configuration

3.) Selection of Hex-file to program:
By button „*Open*“ the Hex-file will be selected and loaded.

To prevent a damage of CANopen IO-C12, only a Hex-file suitable for CANopen IO-C12 is allowed to program.

- start of CANopen IO-C12 and set program-mode:
  1.) Connect the power supply to the device
  2.) Set program-mode by pressing the buttons *Boot* and *Reset on CANopen IO-C12* in following order:

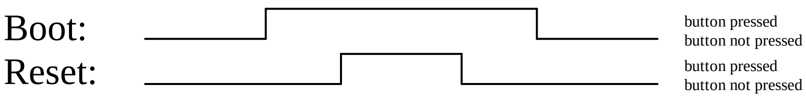

Figure 14: Firmware-Update – Program-mode

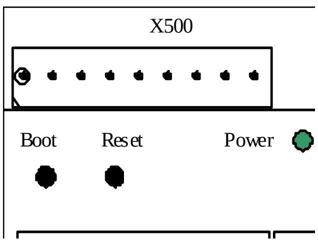

Figure 15: Firmware-Update – Position Boot and Reset

If necessary you will be asked to reset the CANopen IO-C12 during use of flash-tool, what meas to set the program-mode. The press the combination of buttons *Boot* and *Reset* too.

- Erase of previous firmware of CANopen IO-C12:
  Press button „Download“.
  It follows a dialog, that asks you to reset the target, what means to set the program-mode. If it is done, confirm it by OK.

Figure 16: Firmware-Update – Dialog for program-mode

A successful download is shown by following dialog:

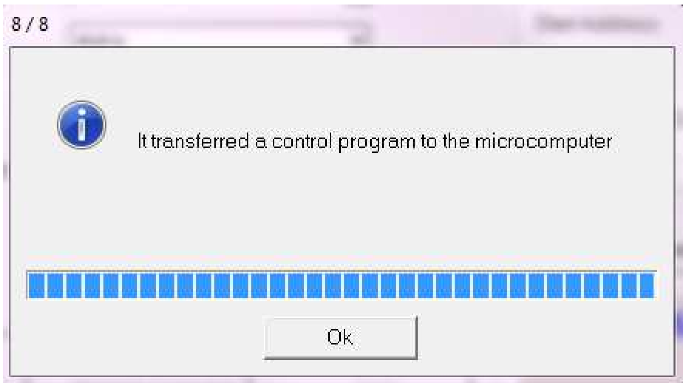

Figure 17: Firmware-Update – Dialog after download

After this press buttons „*Erase*“ for erasing the firmware of CANopen IO-C12.

© SYS TEC electronic GmbH 2013

A successful erasing is shown by following dialog:

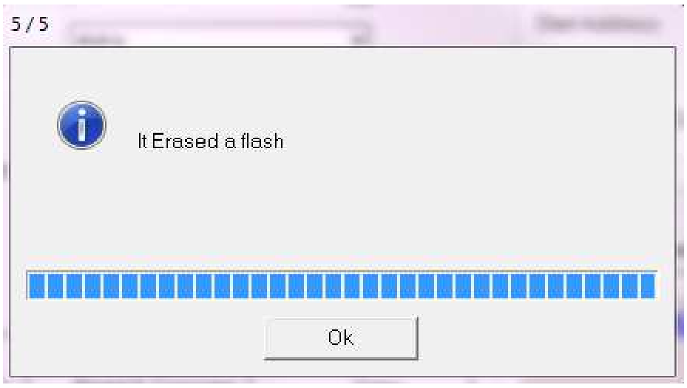

Figure 18: Firmware-Update – Dialog after erase

- Program the new firmware of CANopen IO-C12:
Press button „*Full Operation*“ for program the firmware.
If follows the dialog again, that asks you to set the program-mode. Do this on hardware like it is described above.

A successful programming is shown by following dialog:

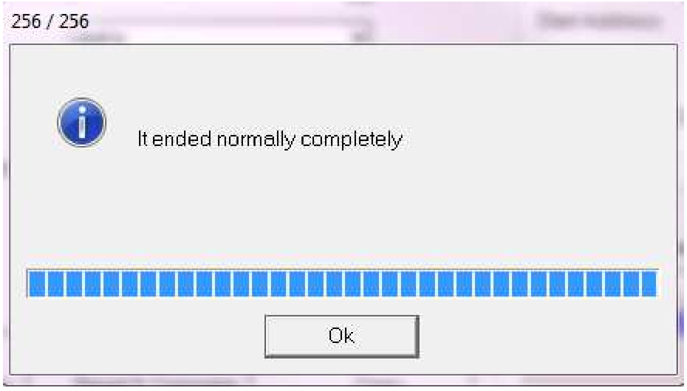

Figure 19: Firmware-Update – Dialog after programming

- Restart of CANopen IO-C12

By button *Reset* on CANopen IO-C12 or by PowerOn the device starts with new firmware.

Revision History of this Document

Date	Manual Version	Changes
04/23/2004	Manual L-1046e_1	Initial translation based on L-1046d_1
01/07/2004	Manual L-1046e_2	Translation based on L-1046d_3
07/07/2004	Manual L-1046e_3	Additional description for customer specific version phyPS-409-KSM01
08/10/2009	Manual L-1046e_5	Correct the calculation of AI and AO and PWM frequency Completion error codes and Connection load at digital Outputs
12/07/2013	Manual L-1046e_6	Fullscale of AO changed from 10.19V to 10.35V Fullscale of AI changed from 10.19V to 10.107V
18.10.2013	Manual L-1046e_7	Add chapter Firmware-Update

 © SYS TEC electronic GmbH 2013

© SYS TEC electronic GmbH 2013

Index

1000H....................................... 42
1001H....................................... 42
1003.. 37
1003H....................................... 42
1005H....................................... 42
1008H....................................... 42
1009.. 42
100AH...................................... 42
100BH...................................... 42
100CH...................................... 42
100DH...................................... 42
1029.. 61
1400h.. 32
1800h.. 32
6000h.. 34
6200h.. 34
6206.. 61
6207.. 61
6421.. 51
6422.. 51
6423.. 52
6424.. 52
6425.. 52
6426.. 53
Array 30
Base Identifier........................... 40
Board Configuration................. 8
Bus Off...................................... 62
CAL.. 25
CAN Application Layer............ 25
CAN Bitrate 8
CAN Bus Cable 12
CAN Identifier 20
CAN in Automation e.V. 23, 29
CAN Interface 12, 15
CAN Message 20
CANopen Advantages 24
CANopen Features.................... 24
CiA DS 401............................... 27
CiA DSP 402............................. 27
CiA DSP 404............................. 28
CiA DSP 405............................. 28
CiA DSP 406............................. 28
CiA DSP 410............................. 28
CMS 25
COB 25
Communication Profile........... 30
Connector Description.............. 5
DBT.. 25
DBT-Master 42
Defstruct................................... 30
Deftyp....................................... 30
Delta Function........................... 53
Device Name............................. 42
Device Pinout 4
Device Profiles 30
Device Type 42
DIP-Switch................................. 8
Domain..................................... 30
Domains 25
Dummy16................................. 35
E^2PROM................................... 47
EMC 1
Emergency................................ 37
Emergency Message.................. 55
Emergency Object..................... 37
Enter_Pre_Operational_State.... 19
Error Code 56
Error Handling......................... 37
Error Led 57
Error Message 37
Error Register............................ 56
Events....................................... 25
Guard-Time............................... 42
Handshaking.............................. 31
Hardware Version 42
Heartbeat 38
Heartbeat Consumer............... 39

 © SYS TEC electronic GmbH 2013

Heartbeat Message 38
Heartbeat Producer 38
HEX-encoding Switch 9
Index 34
Inhibit Time 32, 33
Interrupt Enable 52
Interrupt Trigger 51
Interrupt_Lower_Limit 52
Interrupt_Source 51
Interrupt_Upper_Limit 52
Introduction 3
ISO 11898 29
Lifeguarding 21
Life-Guarding 37, 62
Life-Time-Factor 42
LMT 25
Load_Start 47
Lower Limit 52
Mapping Entry 34
Message Identifier 32
Minimum Boot-Up 19
Multiplexed Domain Protocol .. 31
Network Management 37
NMT 25, 37
NMT-Master 25, 37
NMT-Slave 25
Node Address 9, 20, 39, 42
Nodeguarding 21
Node-Guarding 46
Object Dictionary 26, 42
OD 26
Open Networking System 29
Operational 40, 46
PDO 26, 32, 34
PDO Mapping 20
PDO Mapping Tables 35
PDO-Mapping 34
Pin assignment 5
Pin description 5
Power Supply 15
Power-On 46, 48
Pre_Operational 40, 46
Pre-Defined Error Field 42
Pre-defined Identifier 39
Priority Group 33
Process Data Object 26
Process Data Objects 32, 43
PWM outputs 54
QuickStart 19
Receipt Objects 32
Record 30
Reset 21
Reset_Communication 40
RESTART 47
RTR-Bit 21
RUN Led 56
S1 8
Save_ Start 47
SDO 26, 31, 34
Service Data Objects 26, 31
Shut-Down 19
Software Version 42
Start_Remote_Node 19, 40
Start-Up 19
Status LEDs 56
Stop_Remote_Node 40
STOPPED 46
Subindex 34
Technical Highlights 3
Technical Specification 13
Transmission Objects 32
Transmission Types 33
Upper Limit 52
Var 30
Variables 25

 © SYS TEC electronic GmbH 2013

Document:	CANopen IO-C12
Document number:	L-1046e_7, Edition October 2013

How would you improve this manual?

Did you find any mistakes in this manual? page

Submitted by:

Customer number:

Name:

Company:

Address:

Return to:

SYS TEC electronic GmbH
Am Windrad 2
D-08468 Heinsdorfergrund
GERMANY
Fax : +49 (0) 3765 / 38600-4100

 © SYS TEC electronic GmbH 2013

Published by

Ordering No. L-1046e_7
Printed in Germany

© SYS TEC electronic GmbH 2013