CANopen DemoSet

User's Manual

Contents

1 Disclaimer Statement

The manufacturer specifically disclaims all warranties, either express or implied, including but not limited to implied warranties of fitness for a particular purpose, with respect to the software, the product manuals and written materials and any other accompanying hardware. The manufacturer reserves the right to revise or make improvements to its product at any time and without obligation to notify any person of such revisions or improvements.
In no event shall the manufacturer be liable for any consequential or incidental damages, including any loss of business profits or any other commercial damages, arising out of the use of its product.

2 Parts of the CANopen DemoSet

The delivered CANopen DemoSet should contain the following items:

- NetPorty II
- Cable for NetPorty II serial port connection
- NiPC2 Input / Output board with piggy back C166 controller board, the CANopen firmware is delivered within the FLASH on the piggy back.
- CD- ROM with CANopen Demo Software

3 Getting Started

3.1 CANopen DemoSet Software Intallation

Start your Win 9x or NT computer, quit any application programs before you install the software package. Insert the delivered CD- ROM into your CD- drive.
Installation of Version 1.0:
Start the program “setup.exe” on the CD. Please follow the instructions on the screen. The installation will be done in three steps. The first is the installation of a Tcl/Tk package. If you have already Tcl/Tk on your PC (Version 8.01 or higher) then it is not necessary to install it again. In the second step some additional files will be copied in the Tcl directory. If you use your own Tcl put the correct path into the list box.-The third step is the installation of the CANopen files. The files for the CANopen DemoSet will be installed automatically. After this set up process you have to do the following things (beginning with “Important”) to make the installation complete.

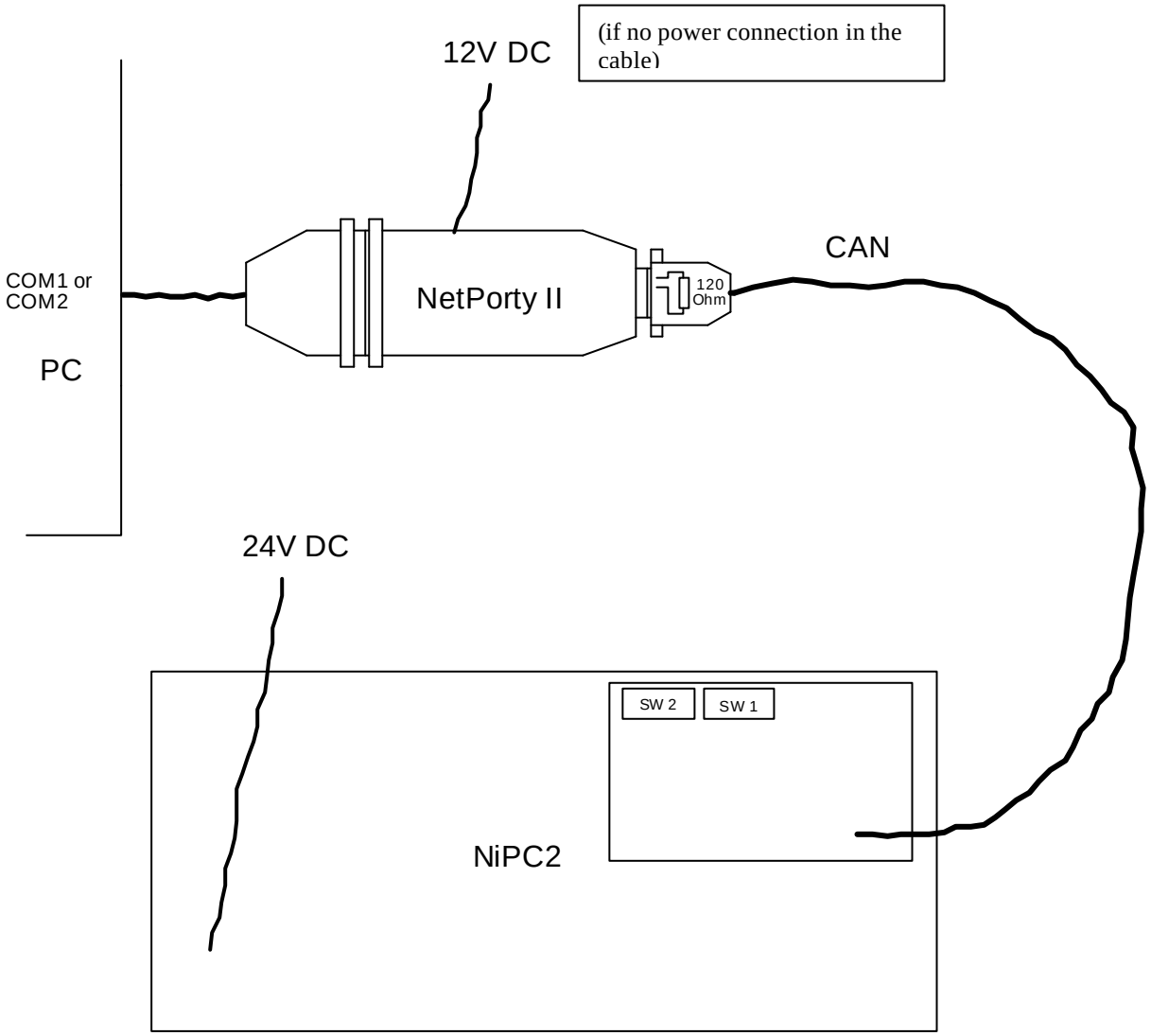
Installation of Version 1.1:
Start the program “canopen_starterkit.exe”. Please follow the instructions on the screen. After finishing of this installation procedure start the Tcl installation by start of the program C:\Programme\I+ME ACTIA\CANopen Demo\TCLSetup\tcl803.exe. After this set up process you have to do the following things to make the installation complete.

IMPORTANT: If you chose another directory than “C:\Programme\TCL” then you have to change the path statement in the file “start.tcl” line 4!! And if you use demo- tcl-scripts from this package you have to check on correct path statements!!

1) The software for a Microsoft TCP/IP protocol must be installed on your PC. You can check this by opening the control panel and double click on the network icon. If TCP/IP is not installed, use the “ADD” button to install it from your Windows NT or 9x CD. A TCP/IP address must be specified (see TCP/IP features). If no address is specified you can use the following settings, which are allowed for internal subnets. The settings of the IP address are 192,168,1,5 and the subnet mask are 255,255,255,0. If you want to use an other IP address then you have to change the address within the file “CDM.RC” line 17.

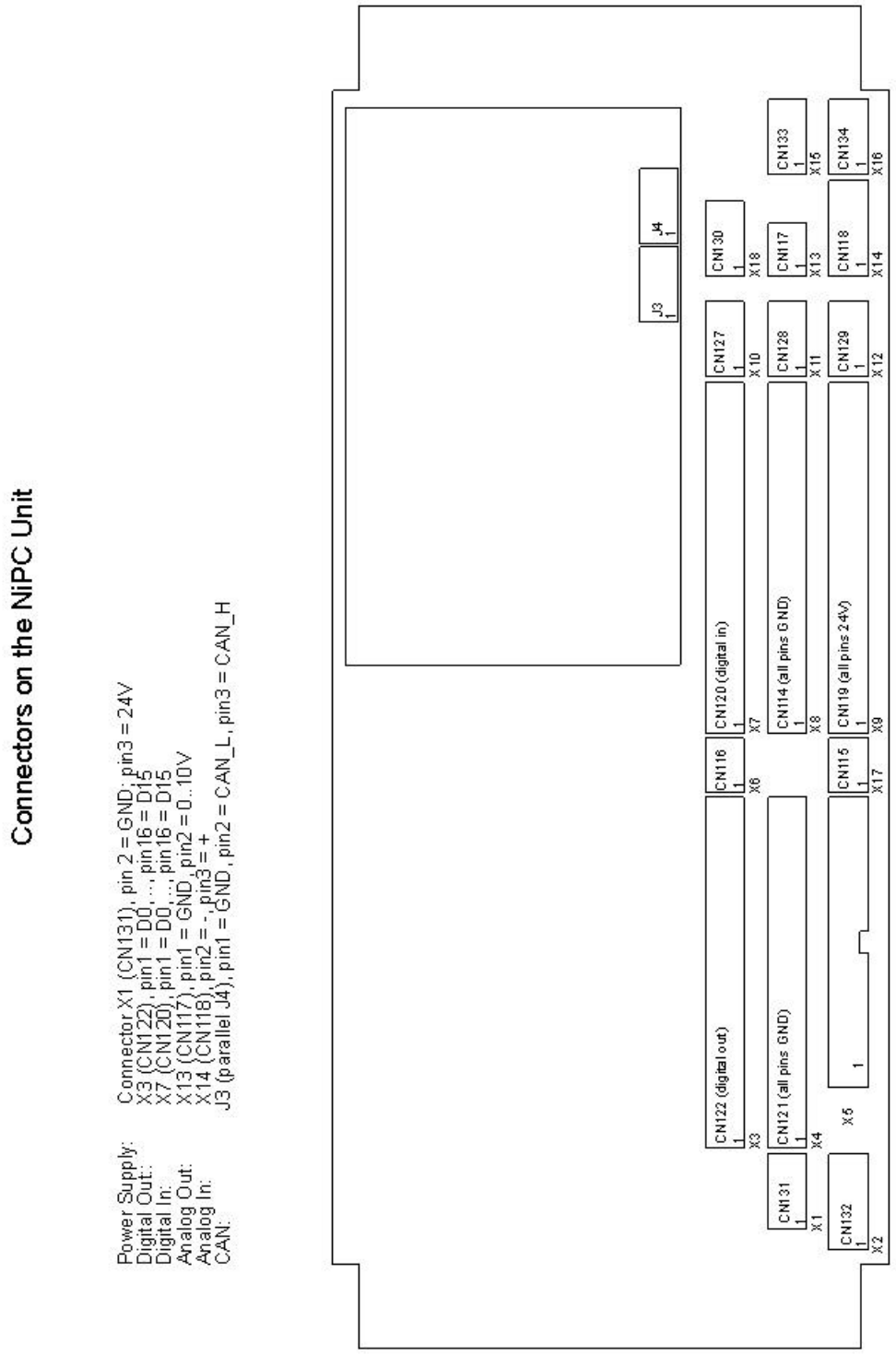
2) Then you should reboot your computer.

3.2 CANopen DemoSet Hardware Intallation



For the CAN connection a “twisted pair”- cable must be used. The CAN connector at the NetPorty II must be equipped with a 120 Ohm termination resistor if a network like shown above is used. The resistor is connected at Pin 2 (CAN L) and Pin 7 (CAN H).

3.3 Connector Assignment on the NiPC2 Board



3.3 CANopen Device Monitor

If you want to work with the CANopen Device Monitor then you have at first to start the CANopen server. For this click to Start/Programme/CANopen DemoSet/CANopen Start (Version 1.0) or click to Start/Programme/CANopen DemoSet/CANopen PC Interface Driver (Version 1.1). The server start window will pop up. Select the connected hardware in the corresponding list box. In this case it could be “Porty2-COM1” or “Porty2-COM2” depending on the serial port you use. Select the desired CAN bit rate in kBit/s. Remark: if the default value of 125kBit/s is selected the switch 4 of SW1 on the NiPC2 piggy back must be in on position. Start the server by pressing the button with the start flag. Do not terminate the server. If the message “Waiting for connections on port xxxx” appears, you can start the Device Monitor: click to Start/Programs/CANopen DemoSet/CANopen Device Monitor or start the program “cdm.tcl” on the explorer. The device monitor reads an electronic data sheet of the NiPC2. The data sheet is stored in the file NiPC2.eds. Do not modify this file. The complete features of the Device Monitor are described in the paper “User Manual CANopen Device Monitor”. Some demo files are located in the directory ..\CANopen Demo\Device_Monitor. You may load a tcl- file by using the console function of the device monitor. If you load the file “digout.tcl” then you can start a test program for the digital outputs of the NiPC2 by writing “application (enter)” (without inverted comas) in the console window. The state of the digital output lines of the NiPC2 is shown by the red LED at the output connector. Every time if you quit the pop up message box of the output test the state of the output will be changed.

3.4 C- Demo

On the delivered CD- ROM are two projects stored additionally in zipped files. They are placed in the directory called C_Demo. After extraction of the file M1.zip a project folder with sample master CANopen application written in C will appear. S1.zip contains a sample slave application also written in C.

4 Technical Data of the NiPC2 CANopen Hard- and Software

4.1 CPU

CPU type	SAB 80C166
CPU frequency	40 MHz (internal divided by 2)

4.2 Timer 0 – used for Analog Output

Timer prescaler	Output of prescaler	Output period	Desired period	Necessary timer count	Timer start value
16	2.5 MHz	400 ns	102400 ns	256	FEFFh

T0CON register bits	Meaning	Bit value (binary)
T0I	Timer frequency = f_{CPU} / 16	000
T0M	Mode: Timer	0
T0R	Timer stopped during init	0

4.3 Timer 4 – used for CANopen

Timer prescaler	Output of prescaler	Output period	Desired period	Necessary timer count	Timer start value
64	0.625 MHz	1600 ns	1 ms	625	FD8Eh

T4CON register bits	Bedeutung	Bit value (binary)
T4I	Timer frequency = f_{CPU} / 64	010
T4M	Mode: Timer	000
T4R	Timer stopped during init	0
T4UD	Count up	0

4.4 Interrupts

IRQ Control Register	IRQ Level	IRQ Group Level	symbol. Address	Address	Short address	TRAP Number	Vector address	Source
GPT1 Timer 4	11	0	T41C	FF64h	B2h	24h	90h	CANopen Timer
CAPCOM Register 15	10	0	CC14IC	FF94h	CAh	1Eh	78h	CAN0
CAPCOM Register 15	10	0	CC15IC	FF96h	CBh	1Fh	7Ch	CAN1*

* not implemented

4.5 CAN

CAN Controller	CAN0: 82C200, CAN1: 82C200
Clock frequency at the CAN controller	16MHz (internal divided by 2)
Value of the Output Control Registers	0xFA
Used CAN Controller for CANopen	CAN1

CAN Controller	Anschluß	Chip	Chip Adresse
CAN0	Connector J4 and J3 on the piggy pack	U18 on the piggy back	1D000h
CAN1 (not used for CANopen)	CN134 on the base board	U19 on the piggy back	1D200h

Settings for the CAN-IRQ:

CAN Controller	Capture-Compare-Mode-Register	Address	Short address	Capture-Compare-Mode-Register		
				Name	Bits	Bit value
CAN0	CCM3	FF58h	ACh	CCMOD14	10-8	%010 = falling edge
CAN1	CCM3	FF58h	ACh	CCMOD15	14-12	%010 = falling edge

4.6 Digital Inputs

Register	Register bits	Digital input	Bit value	Voltage
1D801h	0 - 7	DI_0 – DI_7	0	0 < U < 6,6V
			1	19V < U < 24V
1DA01h	0 - 7	DI_8 – DI_15	0	0 < U < 6,6V
			1	19V < U < 24V

4.7 Digital Outputs

Register	Register bits	Digital Output	Bit value	Output signal
1D800h	0 - 7	DO_0 – DO_7	0	High- Z (GND)
			1	24 Volts
1DA00h	0 - 7	DO_8 – DO_15	0	High- Z (GND)
			1	24 Volts

The digital Outputs of the NiPC2 are driven by “high side-switches” type BTS412B. Each of these switches are equipped with a fault detection circuit. The micro controller can read back the state of these fault detection circuits by reading a state register.

State register	Register bits	Digital Output	Bit value	Meaning
1DC00h	0 – 7	DO_0 – DO_7	0	Fault
			1	No Fault
1DE00h	0 - 7	DO_8 – DO_15	0	Fault

State register	Register bits	Digital Output	Bit value	Meaning
			1	No fault

It is possible to switch off the fault control of unused digital outputs by using the CANopen object *man_config_state_8*.

4.8 Analog Input (10 Bit Unipolar)

Register	Analog Input	Register value	Input Voltage
Internal ADC Channel 0	AI	0 – 3FFh	0 – 10 Volts

4.9 Analog Output (8 Bit Unipolar)

The analog output is implemented by using the PWM- stage of the micro controller C166. The stage is programmed as follows:

Analog Output	Port	Capture-Compare-Register	Capture-Compare-Mode-Register					
			Register	address	Short address	Name	Bits	Bit value
AO	P2.12	CC12	CCM3	FF58h	ACh	CCMOD12	2-0	%111

Values for output voltage generation:

Byte value	p401_write_anout16 (= Byte value << 7)	CC12 register value	Voltage
0 = 0000h	0x0000	0xFFFF	0 Volts
127 = 007Fh	0x3F80	0xFF7F	5 Volts
255 = 00FFh	0x7F80	0xFEFE	10 Volts

4.10 Counter

The counter units of the NiPC hardware are not supported by the CANopen software.

4.11 DIP- Switches

The eightfold DIP- switch SW1 is intended for the selection of the CAN bit rate. If more then one switch is in the ON- position the information from the switch is ignored and the default bit rate of 20 kBit/s is selected.

Eightfold DIP- switch SW1				
Switch No.	Switch position	Port	Value at the port	CAN bit rate in kBit/s
1	ON	P3.0	1	10
	OFF		0	
2	ON	P3.1	1	20
	OFF		0	

Eightfold DIP- switch SW1				
Switch No.	**Switch position**	**Port**	**Value at the port**	**CAN bit rate in kBit/s**
3	ON	P3.3	1	50
	OFF		0	
4	ON	P3.4	1	125
	OFF		0	
5	ON	P3.5	1	250
	OFF		0	
6	ON	P3.8	1	500
	OFF		0	
7	ON	P3.9	1	1000
	OFF		0	
8	ON	P3.12	1	-
	OFF		0	

The second eightfold DIP- switch SW2 is intended for the node ID selection. All values from 1 to 127 are allowed. Every other value is ignored and the ID 127 is used.

Eightfold DIP- switch SW2					
Switch No.	**Switch position**	**Register**	**Register bit**	**Bit value**	**Node-ID**
1 – 8	ON	1D600h	0 - 7	1	Equal to the register value if in the range 1 .. 127.
	OFF			0	

4.12 LED on Piggy Back

LED		
Colour	**Bit value**	**State**
Green	0	Off
	1	On
Red	0	Off
	1	On

5 CANopen Implementation

5.1 Basic Standards and Versions

The software of the NiPC2 CANopen DemoSet is based on the following CiA (CAN in automation) standards:

- DS-301, Version 4.0
- DS-401, Version 1.4

The implementation was made by using the CANopen library from *port* version 4.0.

5.2 Catalog of Implemented Objects

Index (hex)	Sub-index	Object	Data-type	Range	Default value
1000	0	p301_device_type	u32	-	000F0191h
1001	0	p301_error_register	u8	-	0
1003		p301_pre_defined_error_field		-	-
	0	Number of errorsl	u32	0, 1	0
	1	Buffer for the last error	u32	0 - FFFFh	0
1004		p301_number_of_pdos			
	0	total number of PDOs	u32	-	00020002h
	1	number of synchronous PDOs	u32	-	00000000h
	2	number of asynchronous PDOs	u32	-	00020002h
1008	0	p301_manufacturer_device_name	v	-	I+ME – NiPC2
1009	0	p301_manufacturer_hardware_version	v	-	1.03
100A	0	p301_manufacturer_software_version	v	-	1.0
100B	0	p301_node_id	u32	1 - 127	127
100C	0	p301_guard_time	u16	0 - 65535 ms	0
100D	0	p301_life_time_factor	u8	0 - 255	0
100E	0	p301_node_guarding_identifier	u32	1 - FFFFFFFFh	77Fh
1014	0	p301_emergency_identifier	u32	1 - FFFFFFFFh	FFh
1018		p301_identity_object			
	0	number of sub indices	u8	-	1
	1	vendor-ID	u32	-	41h
1400		P301_n1_receive_pdo_parameter			
	0	maximum sub index	u8	-	3
	1	COB-Id of RPDO1	u32	1 - FFFFFFFFh	27Fh
	2	Transmission_type of RPDO1	u8	-	255
	3	Inhibit_time of RPDO1	u16	0 – 255	0
1401		P301_n2_receive_pdo_parameter			
	0	maximum sub index	u8	-	3
	1	COB-Id of RPDO2	u32	1 – FFFFFFFFh	37Fh
	2	Transmission_type of RPDO2	u8	-	255
	3	Inhibit_time of RPDO2	u16	0 – 255	0
1600		P301_n1_receive_pdo_mapping			
	0	Number of mapped objects	u8	-	2
	1	1st mapped value of RPDO1	u32	-	62000108h
	2	2nd mapped value of RPDO1	u32	-	62000208h
1601		P301_n2_receive_pdo_mapping			
	0	Number of mapped objects	u8	-	1
	1	1st mapped value of RPDO2	u32	-	64110110h
1800		P301_n1_transmit_pdo_parameter			
	0	maximum sub index	u8	-	3
	1	COB-Id of TPDO1	u32	1 – FFFFFFFFh	1FFh
	2	Transmission_type of TPDO1	u8	-	255
	3	Inhibit_time of TPDO1	u16	0 – 255	0

Index (hex)	Sub-index	Object	Data-type	Range	Default value
1801		P301_n2_transmit_pdo_parameter			
	0	maximum sub index	u8	-	3
	1	COB-Id of TPDO2	u32	1 – FFFFFFFFh	2FFh
	2	Transmission_type of TPDO2	u8	-	255
	3	inhibit_time of TPDO2	u16	0 – 255	0
1A00		p301_n1_transmit_pdo_mapping –			
	0	number of mapped objects	u8	-	2
	1	1st mapped value of TPDO1	u32	-	60000108h
	2	2nd mapped value of TPDO1	u32	-	60000208h
1A01		p301_n2_transmit_pdo_mapping –			
	0	number of mapped objects	u8	-	1
	1	1st mapped value of TPDO2	u32	-	64010110h
2000		man_error **			
	0	number of the sub indices	u8	-	2
	1	register	u32	0 ... FFFFFFFFh	0
	2	channel state of DO_0 – DO_15	u16	0 ... FFFFh (bit value: 0 = channel faultless, 1 = channel faulty)	0
2001		man_config_state_8 **			
	0	number of 8--bit DO-register	u8	-	2
	1	register for DO_0 .. DO_7	u8	0 ... FFh	0
	2	register for DO_8 .. DO_15	u8	1 ... FFh	0
2002		Man_status_state_8 **			
	0	number of 8- bit DO-register	u8	-	2
	1	register for DO_0 .. DO_7	u8	-	*
	2	register for DO_8 .. DO_15	u8	-	*
6000		p401_read_state_8			
	0	number of 8- bit DI-register	u8	-	2
	1	register for DI_0 .. DI_7	u8	0 ... FFh	*
	2	register for DI_8 .. DI_15	u8	0 ... FFh	*
6002		p401_polarity_read8			
	0	number of 8-bit DI-register	u8	-	2
	1	polarity of DI_0 .. DI_7	u8	0 ... FFh	0
	2	polarity of DI_8 ..DI_15	u8	0 ... FFh (bit value 0 = signal not inverted; 1 = signal inverted	0
6005	0	p401_global_enable_int	u8	0 – disable TPDOs 1 – enable TPDOs	1

Index (hex)	Sub-index	Object	Data-type	Range	Default value
6006		p401_int_any_8			
	0	number of 8- bit DI-register	u8		2
	1	TPDO at signal change of DI_0 .. DI_7	u8	0 ... FFh	FFh
	2	TPDO at signal change of DI_8 .. DI_15	u8	0 ... FFh (bit value: 0 = no TPDO transmission, 1 = TPDO transmission)	FFh
6007		p401_int_lowhigh_8			
	0	number of 8- bit DI-register	u8		2
	1	TPDO at low → high transition DI_0 .. DI_7	u8	0 ... FFh	0
	2	TPDO at low → high transition DI_8 .. DI_15	u8	0 ... FFh (bit value: 0 = no TPDO transmission, 1 = TPDO transmission)	0
6008		p401_int_highlow_8			
	0	number of 8- bit DI-register	u8	-	2
	1	TPDO at high → low transition DI_0 .. DI_7	u8	0 ... FFh	0
	2	TPDO at high → low transition DI_8 .. DI_15	u8	0 ... FFh (bit value: 0 = no TPDO transmission, 1 = TPDO transmission)	0
6020		p401_read_state_bl1_1			
	0	number of the bits	u8	-	16
	1-16	DI_0 .. DI_15	u8	0, 1	*
6200		p401_write_state_8			
	0	number of 8- bit DO-register	u8	-	2
	1	register for DO_0 .. DO_7	u8	0 ... FFh	0
	2	register for DO_8 .. DO_15	u8	0 ... FFh	0
6202		p401_polarity_write_8			
	0	number of 8- bit DO-register	u8	-	2
	1	polarity of DO_0 .. DO_7	u8	0 ... FFh	0
	2	polarity of DO_8 .. DO_15	u8	0 ... FFh (bit value: 0 = signal not inverted 1 = signal	0

Index (hex)	Sub-index	Object	Data-type	Range	Default value
				inverted)	
6206		p401_fault_mode_8			
	0	number of 8- bit DO- register	u8	-	2
	1	DO_0 .. DO_7 set to fault state in case of fault	u8	0 ... FFh	FFh
	2	DO_8 .. DO_15 set to fault state in case of fault	u8	0 ... FFh (bit value: 0 = DO unchanged 1 = DO set to *p401_fault_state_8*)	FFh
6207		p401_fault_state_8			
	0	number of 8- bit DO register	u8	-	2
	1	DO_0 ..DO_7 fault state information	u8	0 ... FFh	0
	2	DO_8 .. DO_15 fault state information	u8	0 ... FFh	0
6220		p401_write_state_bl1_1			
	0	number of the bits	u8	-	16
	1-16	DO_0 .. DO_15	u8	0,1	0
6401		p401_read_anin_16			
	0	number of the analog inputs	i16	-	1
	1	register of analog input	i16	0 ... 7FE0h*** (ADC: 0 ... 3FFh, positive values only)	*
6411		p401_write_anout_16			
	0	number of analog outputs	i16	-	1
	1	register analog output	i16	0 ... 7F80h*** (DAC: 0 ... FFh, positive values only)	0
6421		p401_anin_trigger_sel			
	0	number of analog inputs	u8	-	1
	1	event for PDO transmission	u8	bit number: 0 = cross the upper limit 1 = cross the lower limit 2 = difference between two measurements to large 3-4 = not supported	0

Index (hex)	Sub-index	Object	Data-type	Range	Default value
				5-7 = reseved bit value: 0 = no PDO transmission, 1 = PDO transmission	
6423	0	p401_anin_int_enable	u8	0 = no PDO transmission 1 = PDO transmission	1
6424	 0 1	p401_anin_int_upper number of the analog inputs upper limit of the analog input	 i32 i32	 - 1 ... 7FE0h	 1 7FE0h
6425	 0 1	p401_anin_int_lower number of the analog inputs lower limit of the analog input	 i32 i32	 - 0 ... 7FDFh	 1 0
6426	 0 1	p401_anin_int_delta number of the analog inputs maximum permissible difference between two measurements	 u32 u32	 - 0 ... 7FE0h	 1 7FE0h
6443	 0 1	p401_anout_fault_mode number of the analog outputs analog output set to fault state in case of fault	 u8 u8	 - 0 = analog out unchanged in case of fault 1 = analog out set to *p401_anout_fault_ value* in case of fault	 1 1
6444	 0 1	p401_anout_fault_value number of the analog outputs analog output in case of fault	 u32 u32	 - 0 ... 7F80h*** (DAC: 0 .. FFh, positive values only)	 1 0

* ... the actual input value
** ... see chapter “Error handling“
*** ... according to the standard DS-401 analog values should be stored with the sign in bit 15 and the value in the bits 14..0

6 CAN- Communication

6.1 Communication Objects

Service Data Objects (SDO)

SDO	Data flow	COB-ID (Default)
1	From the application to NiPC2 from NiPC2 to the application	600h + *p301_node_id* 580h + *p301_node_id*

Remark: SYNC messages are not supported.

Process Data Objects (PDO)

NiPC2 PDO	Transmission type	Mapping Object			
		Name	Index	Sub index	Length in Bits
RPDO1	Asynchronous	*p401_write_state_8*	6200h 6200h	01h 02h	08h 08h
RPDO2	Asynchronous	*p401_write_anout_8*	6411h	01h	10h
TPDO1	Asynchronous	*p401_read_state_8*	6000h 6000h	01h 02h	08h 08h
TPDO2	Asynchronous	*p401_read_anin_16*	6401h	01h	10h

The PDO mapping is static. After power on and the change to the communication state "OPERATIONAL" the NiPC2 reports all changes at its inputs by TPDO.

Emergency objects (Emergency Message)
See chapter "Error handling".

Node Guarding
The NiPC2 software supports "Node Guarding".

Heartbeat
The heartbeat is not supported by the software.

7 Reset Communication

All communication parameter are set to the default values. The DIP- switch SW2 set the actual "Node- ID" of the NiPC2. ID's in the range 1 .. 127 are allowed.
The DIP- switch SW1 selects the CAN bit rate. The CAN- controller is set to the selected transmission rate.
All outputs will be set to their fault values if fault values are defined.

8 Reset Application

A "reset application" initiate a completely new start of the NiPC2 firmware

9 Transmit and Receive Buffer

Buffer	Size
Transmit Buffer	20 CAN- Messages
Receive Buffer	20 CAN- Messages

10 Error Handling

10.1 LED-Information

LED		Meaning
Red	**Green**	
Off	On	The system initialization after power on was done successfully.
On	Off	The system initialization after power on was not done successfully
On	Off	CAN communication not possible.

10.2 Emergency Messages

Fault	Emergency Message					LED 1 (red)	Error state for the outputs
	p301_error_code (hex)	p301_error_register	add1	add2	add3		
CAN bus-off	-	Bit 4	-	-	-	On	x
CAN error passive	8120	Bit 4	0	0	0	Off	-
Initialization fault	-	Bit 7	-	-	-	On	-
CAN overflow	8110	Bit 4	0	0	0	On	x
Buffer overflow	81F0	Bit 4	0	0	0	On	x
I/O Hardware	FF01	Bit 7	k	0	0	Off	-
Node Guarding	8130	Bit 4	0	0	0	On	x
Internal data fault	-	Bit 7	-	-	-	On	-

k ... channel number, x ... is supported, - ... Outputs unchanged

Output	Output channel	
	Set fault signal active	Fault signal
DO_0 ... DO_15	*p401_fault_mode_8* = 0	DO_x unchanged
	p401_fault_mode_8 = 1	*p401_fault_state_8*
AO	*p401_anout_fault_mode* = 0	AO unchanged
	p401_anout_fault_mode = 1	*p401_anout_fault_value*

10.3 Error Register

This object is for the internal control only and supports the error diagnosis additional.

Index	2000h
Name	*man_error*
Object code	RECORD
Access	r
Number of elements	2

Value description:

Sub index	1	
Description	Error register (*reg*)	
Object class	M: -	O: all
PDO mapping	-	
Unit	-	
Data range	See tab below	
Default	0	
Data type	UNSIGNED32	

Sub index	2	
Description	DO Error channel (*channel*) at I/O hardware fault	
Object class	M: -	O: all
PDO mapping	-	
Unit	-	
Data range	Bit 15 .. 0 correspond to DO channel 15 .. 0 Bit value 0: channel is ok, bit value 1: channel is faulty (example: 0x0140 = channel DO_8 and DO_6 are disturbed)	
Default	0	
Data type	UNSIGNED16	

Error codes:

Bit in *man_error.reg*	**Meaning (Bit value = 1)**
0 – 1	Initialization of the CAN-Controller faulty (see tab below)
2	CAN bus-off
3	CAN Controller error passive
4	CAN Controller overflow
5	CAN Buffer overflow
6	Node Guarding
7	one or more DO channel faulty
8 – 25	Reserved
26 – 31	internal data error

Bit in man_error.reg		**Meaning**
0	**1**	
1	0	Hardware reset at CAN Controller active
0	1	CAN bit rate unknown

Bit in man_error.reg		Meaning
0	**1**	
1	1	Unknown error

10.4 DO Status

Index	2002h
Name	*man_status_state_8*
Object code	ARRAY
Access	r
Number of elements	2

Value description:

Sub index	1	
Description	State of DO_0 .. DO_7	
Object class	M: -	O: all
PDO mapping	-	
Unit	-	
Data range	0 ... FFh	
Default	actual state	
Data type	UNSIGNED8	

Sub index	2	
Descrition	State of DO_8 .. DO_15	
Object class	M: -	O: all
PDO mapping	-	
Unit	-	
Data range	0 ... FFh	
Default	actual State	
Data type	UNSIGNED8	

10.5 DO Error Detection Enable

	2001h
Name	*man_config_state_8*
Object code	ARRAY
Access	R
Number of elements	2

Value description:

Sub index	1	
Description	Enable / disable error detection DO_0 .. DO_7	
Object class	M: -	O: all
PDO mapping	-	
Unit	-	
Data range	0 ... FFh	

	bit value 0: error detection disabled bit value 1: error detection enabled
Default	0
Data type	UNSIGNED8

Sub index	2	
Description	Enable / disable error detection DO_8 .. DO_15	
Object class	M: -	O: all
PDO mapping	-	
Unit	-	
Data range	0 ... FFh bit value 0: error detection disabled bit value 1: error detection enabled	
Default	0	
Data type	UNSIGNED8	