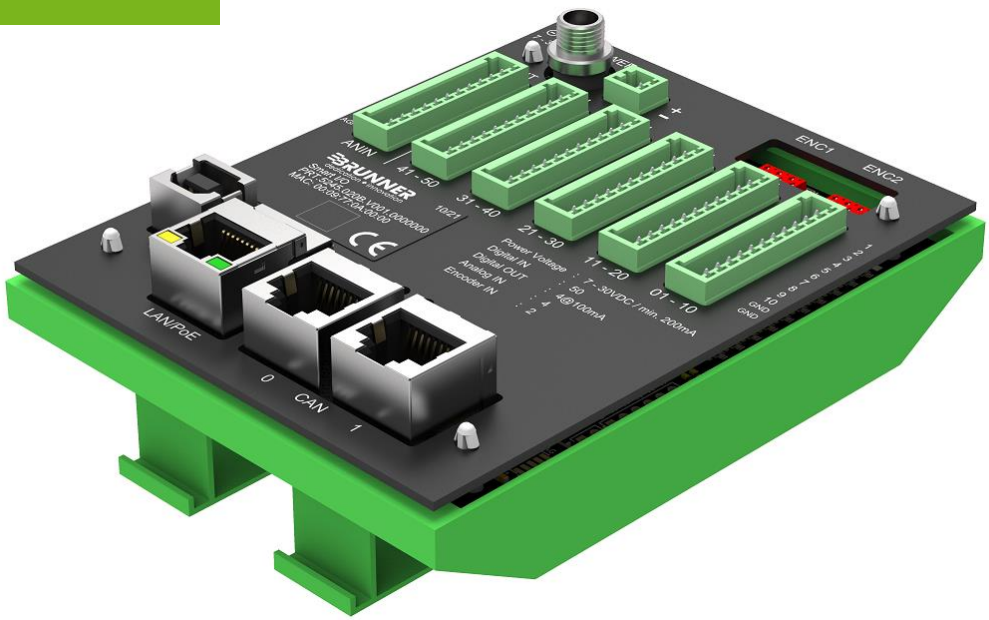
BRUNNER
dedication + innovation

USER MANUAL

Smart I/O

ID	**PRT.5245**
Revision	**1.0**
Date	**17.09.2021**

BRUNNER Elektronik AG
Industriestrasse 27
8335 Hittnau
Switzerland
T +41 (0)44 953 1010
F +41 (0)44 953 1019
www.brunner-innovation.swiss
info@brunner-innovation.swiss



Index

1. Warning and Safety Instructions

The flawless and safe operation of the device requires proper and professional transportation, storage, assembly, project planning as well as careful operation and maintenance. Only trained and qualified personnel should handle electrical installations.

2. Introduction

2.1 Overview

The *Smart I/O* is a CAN gateway device with I/O capabilities.

2.2 Product Features

- CAN Gateway
  - Ethernet to CAN
  - USB to CAN
  - Large Rx/Tx buffers
  - Comprehensive status, error and performance evaluation capabilities
- I/Os
  - 50 Digital Inputs
  - 4 Digital Outputs (up to 100mA)
  - 4 Analog Inputs (12 bit, 3.3V)
  - 2 Incremental Encoders
- CAN
  - ISO-11898, CAN 2.0A/2.0B
  - Galvanically isolated
  - Activatable integrated CAN bus terminating resistor
  - PDO filtering
- Ethernet
  - Auto negotiation: 10BASE-T Half/Full-Duplex, 100BASE-TX Half/Full-Duplex
  - Auto crossover detection
  - DHCP, Static IP or Auto-IP
  - Sustained 100% CAN bus load over Ethernet
  - *horch* Protocol over TCP
  - Multi-socket server
  - Web server for configuration and monitoring via web browser
- USB
  - USB 2.0
  - Gateway: Up to 6000 CAN Packets/s in each direction over USB
  - I/Os: HID Game Controller
- Power options
  - External 24V DC power supply (7V ... 30V)
  - Power over Ethernet (PoE), IEEE 802.3af
  - USB powered (no digital outputs)
- LEDs: Device Status, CAN Indication, Ethernet Link/Activity
- Top hat rail mounting: IEC/EN 60715 – 35mm x 7.5mm / 35mm x 15mm

3. Specifications

3.1 Operating Conditions

Parameter		Remarks	Min.	Typ.	Max.	Unit
Ambient temperature	T_{amb}		-20		75	°C
Relative humidity	φ	Non-condensing	15		85	%

3.2 Electrical Specifications

3.2.1 Power Supply

Parameter		Conditions	Min.	Typ.	Max.	Unit
Supply Voltage	V_{CC}		7	24	30 (*)	VDC
Nominal Current	I_{CC}	V_{CC} = 24V V_{CC} = 24V, CAN & Ethernet V_{CC} = 7V, CAN & Ethernet		45 55 175		mA
Power Consumption	P	Full range, Ethernet connected		1.2	1.5	W

(*) These are the recommended operating conditions. Stressing the device beyond 36V may cause permanent damage.

3.2.2 USB

Parameter		Remarks	Min.	Typ.	Max.	Unit
Current Consumption	I_{USB}	No external power, Ethernet connected		170	300	mA

3.2.3 CAN

Parameter		Remarks	Min.	Typ.	Max.	Unit
CANH Voltage	V_{CANH}	With respect to CAN/GND	-12		12	V
CANL Voltage	V_{CANL}	With respect to CAN/GND	-12		12	V
Isolation Voltage	V_{iso}	Single protection, t = 60 sec			2.5	kV
Electrostatic Discharge	V_{ESD}	Human Body Model Charged Device Model Machine Model			±4 ±1.5 ±0.2	kV

3.2.4 Ethernet/PoE

Parameter		Remarks	Min.	Typ.	Max.	Unit
PoE Input Voltage	V_{PoE}	Rectified DC Voltage	0		57	V
PoE Current Range	I_{PoE}		0		350	mA

3.2.5 Digital Inputs

Parameter		Remarks	Min.	Typ.	Max.	Unit
Input Voltage	V_{IN}		-0.3		3.6	V
Active State	V_{IL}		-0.3	0	0.7	V
Passive State	V_{IH}	Open Drain	1.9	HiZ	3.6	V
Pull-Up Resistor	$V_{I,RPU}$			47		kΩ
ESD	V_{ESD}	Human Body Model Charged Device Model			±2 ±0.5	kV

3.2.6 Digital Outputs

Parameter		Remarks	Min.	Typ.	Max.	Unit
Output Voltage Supply	$V_{OUT,DD}$			V_{IN} – 0.5		V
Low Output Voltage	V_{OL}			0		V
High Output Voltage	V_{OH}			$V_{OUT,DD}$		V
Output Current	I_{OUT}				100	mA
On-State Resistance	R_{ON}	$V_{OUT,DD}$ to OUT		110	280	mΩ

3.2.7 Analog Inputs

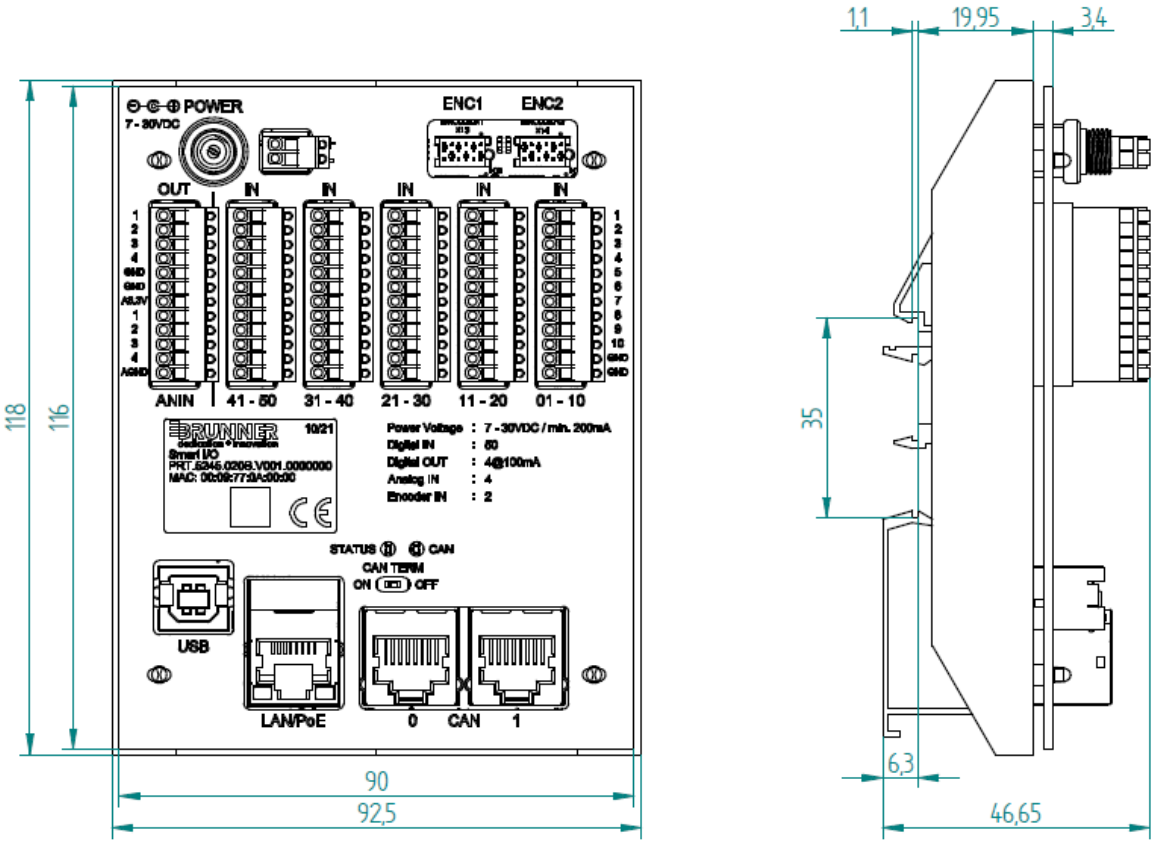
Parameter		Remarks	Min.	Typ.	Max.	Unit
Analog Input Range	V_{ANIN}		0		3.3	V
Resolution	N			12		bits
Total unadjusted error	E_T	Over full input range		±20	±50	LSB
Electrostatic Discharge	V_{ESD}	Human Body Model			±2.5	kV

3.2.8 Incremental Encoders

Parameter		Remarks	Min.	Typ.	Max.	Unit
A/B Input Voltage Range	$V_{A/B}$		0		5.5	V
A/B Active State	$V_{A/B,L}$		0	0	0.8	V
A/B Passive State	$V_{A/B,H}$	Open Drain	2.3	HiZ	5.5	V
A/B Pull-Up Resistor	$V_{A/B,RPU}$			2.2		kΩ
A/B ESD	$V_{A/B,ESD}$	Human Body Model Charged Device Model			±2 ±1	kV
PB Input Voltage Range	V_{PB}		2.15		3.6	V
PB Active State	$V_{PB,L}$		0	0	1.15	V
PB Passive State	$V_{PB,H}$		2.15	HiZ	3.6	V
PB Pull-Up Resistor	$V_{PB,RPU}$			4.7		kΩ
PB ESD	$V_{PB,ESD}$	Human Body Model Charged Device Model			±2 ±0.5	kV

3.3 Mechanical Data

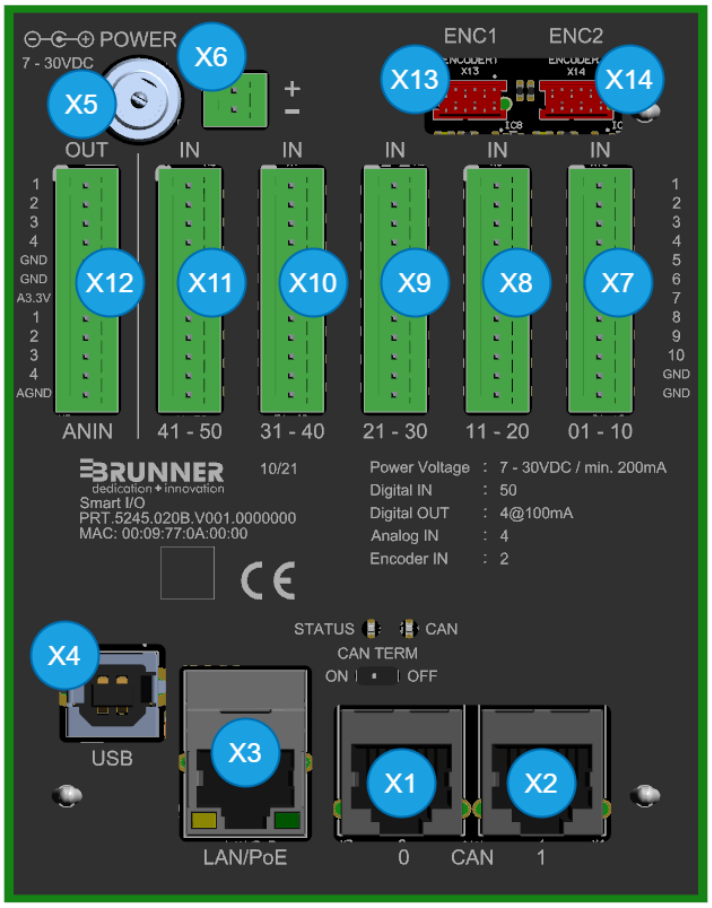
Parameter	Typ.	Unit
Length	118	mm
Width	92.5	mm
Height	46.7	mm



3.4 Mounting

This device is compatible with the IEC/EN 60715 top hat rail 35mm x 7.5mm and 35mm x 15mm for snap-on mounting.

3.5 Connectors



Connector	Description
X1	CAN0 Connector
X2	CAN1 Connector
X3	Ethernet Connector (LAN/PoE)
X4	USB Connector
X5	Power Jack Connector
X6	Power Connector
X7	Digital Inputs 1-10
X8	Digital Inputs 11-20
X9	Digital Inputs 21-30
X10	Digital Inputs 31-40
X11	Digital Inputs 41-50
X12	Digital Outputs 1-4 + Analog Inputs 1-4
X13	Incremental Encoder 1
X14	Incremental Encoder 2

3.5.1 Connector X1 – CAN0

Connector: RJ45

Pin	Name	Description
1	CANH	CAN High Level
2	CAHL	CAN Low Level
3	CAN/GND	CAN Ground
4	-	
5	-	
6	-	
7	CAN/GND	CAN Ground
8	-	
G	Shield	

3.5.2 Connector X2 – CAN1

Connector: RJ45

Pin	Name	Description
1	CANH	CAN High Bus Line
2	CAHL	CAN Low Bus Line
3	CAN/GND	CAN Ground
4	-	
5	-	
6	-	
7	CAN/GND	CAN Ground
8	-	
G	Shield	

3.5.3 Connector X3 – LAN/PoE

Connector: RJ45

10BASE-T/100BASE-TX Mode A

Pin	Name	Description
1	RX+/DC+	Receive Data+ / PoE Power Supply
2	RX-/DC+	Receive Data- / PoE Power Supply
3	TX+/DC-	Transmit Data+ / PoE Return
4	-	
5	-	
6	TX-/DC-	Transmit Data- / PoE Return
7	-	
8	-	
G	Shield	

10BASE-T/100BASE-TX Mode B

Pin	Name	Description
1	RX+	Receive Data+
2	RX-	Receive Data-
3	TX+	Transmit Data+
4	DC+	PoE Power Supply
5	DC+	PoE Power Supply
6	TX-	Transmit Data-
7	DC-	PoE Return
8	DC-	PoE Return
G	Shield	

3.5.4 Connector X4 – USB

Connector: USB Type-B

Pin	Name	Description
1	V_{USB}	+5V Supply Voltage
2	D-	USB Data-
3	D+	USB Data+
4	USB/GND	Ground

3.5.5 Connector X5 – Power Jack

Socket: Ø6.3mm, Ø2.0mm

Mating Connector: Ø5.5mm, Ø2.0mm, 9.5mm (e.g. Switchcraft S760)

Pin	Name	Description
Sleeve	GND	Ground
Center Pin	V_{IN}	Positive Supply Voltage (7 VDC ... 30 VDC, typ. 24 VDC)

3.5.6 Connector X6 – Power

Connector: Phoenix-Contact PCB Connector 1881558, MCV 0,5/2-G-2,5

Pin	Name	Description
1	V_{IN}	Positive Supply Voltage (7 VDC ... 30 VDC, typ. 24 VDC), max 500mA
2	GND	Ground

3.5.7 Connector X7 – Digital Inputs 1-10

Connector: Phoenix-Contact PCB Connector 1881655, MCV 0,5/12-G-2,5

Pin	Name	Description
1	DIG_IN1	Digital Input 1, Open Drain
2	DIG_IN2	Digital Input 2, Open Drain
3	DIG_IN3	Digital Input 3, Open Drain
4	DIG_IN4	Digital Input 4, Open Drain
5	DIG_IN5	Digital Input 5, Open Drain
6	DIG_IN6	Digital Input 6, Open Drain
7	DIG_IN7	Digital Input 7, Open Drain
8	DIG_IN8	Digital Input 8, Open Drain
9	DIG_IN9	Digital Input 9, Open Drain
10	DIG_IN10	Digital Input 10, Open Drain
11	GND	Digital Ground
12	GND	Digital Ground

3.5.8 Connector X8 – Digital Inputs 11-20

Connector: Phoenix-Contact PCB Connector 1881655, MCV 0,5/12-G-2,5

Pin	Name	Description
1	DIG_IN11	Digital Input 11, Open Drain
2	DIG_IN12	Digital Input 12, Open Drain
3	DIG_IN13	Digital Input 13, Open Drain
4	DIG_IN14	Digital Input 14, Open Drain
5	DIG_IN15	Digital Input 15, Open Drain
6	DIG_IN16	Digital Input 16, Open Drain
7	DIG_IN17	Digital Input 17, Open Drain
8	DIG_IN18	Digital Input 18, Open Drain
9	DIG_IN19	Digital Input 19, Open Drain
10	DIG_IN20	Digital Input 20, Open Drain
11	GND	Digital Ground
12	GND	Digital Ground

3.5.9 Connector X9 – Digital Inputs 21-30

Connector: Phoenix-Contact PCB Connector 1881655, MCV 0,5/12-G-2,5

Pin	Name	Description
1	DIG_IN21	Digital Input 21, Open Drain
2	DIG_IN22	Digital Input 22, Open Drain
3	DIG_IN23	Digital Input 23, Open Drain
4	DIG_IN24	Digital Input 24, Open Drain
5	DIG_IN25	Digital Input 25, Open Drain
6	DIG_IN26	Digital Input 26, Open Drain
7	DIG_IN27	Digital Input 27, Open Drain
8	DIG_IN28	Digital Input 28, Open Drain
9	DIG_IN29	Digital Input 29, Open Drain
10	DIG_IN30	Digital Input 30, Open Drain
11	GND	Digital Ground
12	GND	Digital Ground

3.5.10 Connector X10 – Digital Inputs 31-40

Connector: Phoenix-Contact PCB Connector 1881655, MCV 0,5/12-G-2,5

Pin	Name	Description
1	DIG_IN31	Digital Input 31, Open Drain
2	DIG_IN32	Digital Input 32, Open Drain
3	DIG_IN33	Digital Input 33, Open Drain
4	DIG_IN34	Digital Input 34, Open Drain
5	DIG_IN35	Digital Input 35, Open Drain
6	DIG_IN36	Digital Input 36, Open Drain
7	DIG_IN37	Digital Input 37, Open Drain
8	DIG_IN38	Digital Input 38, Open Drain
9	DIG_IN39	Digital Input 39, Open Drain
10	DIG_IN40	Digital Input 40, Open Drain
11	GND	Digital Ground
12	GND	Digital Ground

3.5.11 Connector X11 – Digital Inputs 41-50

Connector: Phoenix-Contact PCB Connector 1881655, MCV 0,5/12-G-2,5

Pin	Name	Description
1	DIG_IN41	Digital Input 41, Open Drain
2	DIG_IN42	Digital Input 42, Open Drain
3	DIG_IN43	Digital Input 43, Open Drain
4	DIG_IN44	Digital Input 44, Open Drain
5	DIG_IN45	Digital Input 45, Open Drain
6	DIG_IN46	Digital Input 46, Open Drain
7	DIG_IN47	Digital Input 47, Open Drain
8	DIG_IN48	Digital Input 48, Open Drain
9	DIG_IN49	Digital Input 49, Open Drain
10	DIG_IN50	Digital Input 50, Open Drain
11	GND	Digital Ground
12	GND	Digital Ground

3.5.12 Connector X12 – Digital Outputs 1-4 + Analog Inputs 1-4

Connector: Phoenix-Contact PCB Connector 1881655, MCV 0,5/12-G-2,5

Pin	Name	Description
1	DIG_OUT1	Digital Output 1
2	DIG_OUT2	Digital Output 2
3	DIG_OUT3	Digital Output 3
4	DIG_OUT4	Digital Output 4
5	GND	Digital Ground
6	GND	Digital Ground
7	+3.3VADC_EXT	+3.3V Analog Power Supply
8	ANIN1	Analog Input 1
9	ANIN2	Analog Input 2
10	ANIN3	Analog Input 3
11	ANIN4	Analog Input 4
12	AGND	Analog Ground

3.5.13 Connector X13 – Incremental Encoder 1

Connector: Micro-Match 7-215079-6

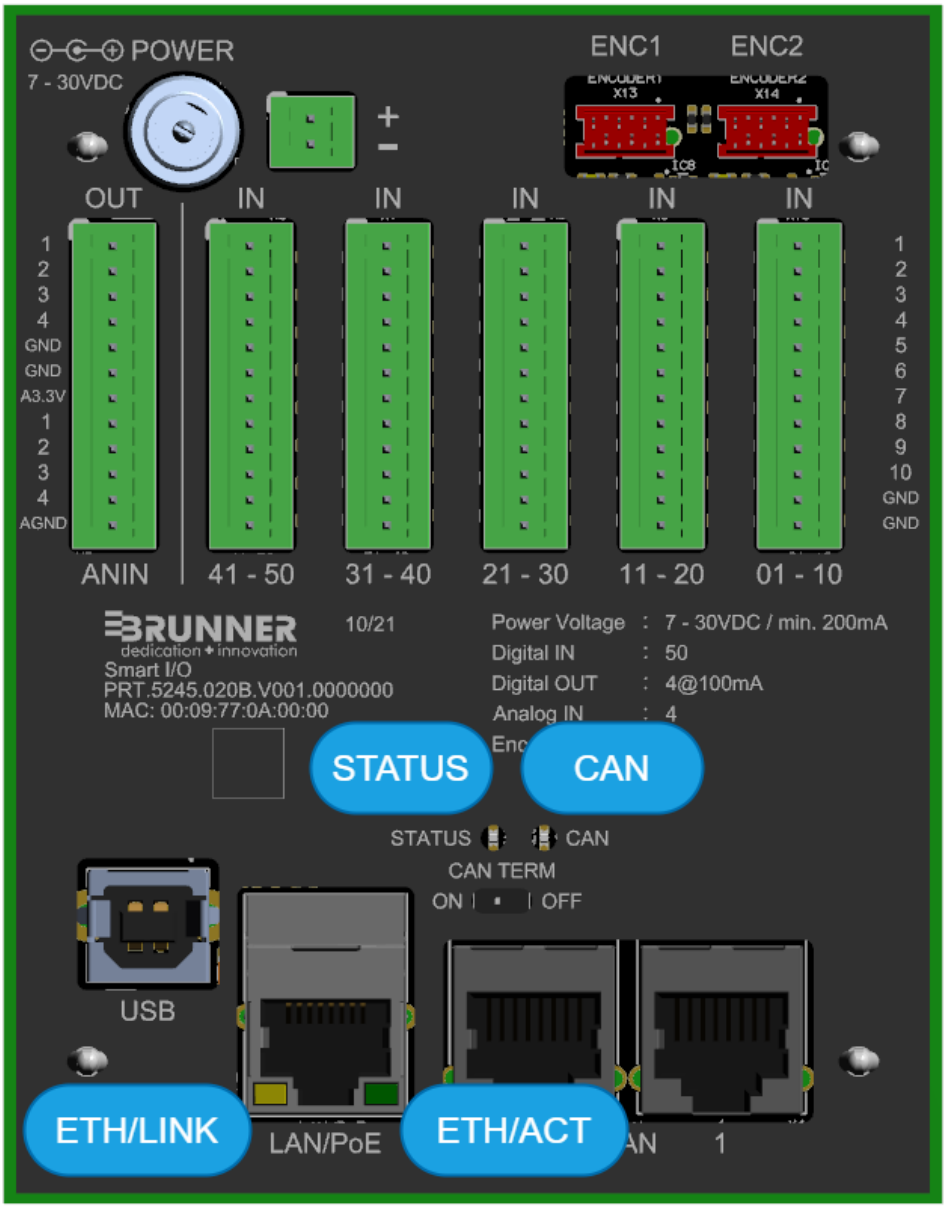
Pin	Name	Description
1	GND	Digital Ground
2	GND	Digital Ground
3	ENC1_PB	Pushbutton Input, Open Drain
4	ENC1_A	Encoder Signal A, Open Drain
5	ENC1_B	Encoder Signal B, Open Drain
6	+5V	+5V Power Supply

3.5.14 Connector X14 – Incremental Encoder 2

Connector: Micro-Match 7-215079-6

Pin	Name	Description
1	GND	Digital Ground
2	GND	Digital Ground
3	ENC1_PB	Pushbutton Input, Open Drain
4	ENC1_A	Encoder Signal A, Open Drain
5	ENC1_B	Encoder Signal B, Open Drain
6	+5V	+5V Power Supply

3.6 LEDs



3.6.1 Device Status LED

LED	LED State	Description
STATUS	OFF Green, blinking (1Hz) Green, blinking (5Hz)	Device not powered or in Update Mode Device is operational, Gateway Mode Device is operational, Sniffer Mode

3.6.2 CAN LED

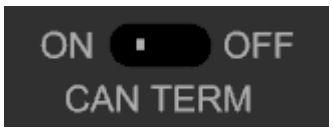
LED	LED State	Description
CAN	OFF Flashing Green	No CAN Activity CAN Rx/Tx Activity

3.6.3 Ethernet LEDs

LED	LED State	Description
ETH/LINK	OFF Amber	No Ethernet Link Ethernet Link is up
ETH/ACT	OFF Flashing Green	No Ethernet Activity Ethernet Rx/Tx Activity

3.7 CAN Terminating Resistor

This device has an integrated 124Ω CAN terminating resistor.
It can be enabled or disabled with a mechanical switch. This switch is located right next to the CAN connectors.



In the ON position the CAN terminating resistor is enabled, and in the OFF position it is disabled.

4. Communication

The *Smart I/O* allows to interface CAN networks to a PC over Ethernet or USB.

Supported CAN Interface Features:

- Supported Bitrates: 125 kbps, 250 kbps, 500 kbps, 1 Mbps
- All standard CAN packet types
- Object Identifiers: 11 bit (CAN 2.0A), 29 bit (CAN 2.0B)
- RTR

4.1 Ethernet Protocol

A TCP connection to the gateway can be established on either port 7235 (control channel) or 7236 (diagnostics channel). The interrupts on port 7235 are serviced with a higher priority, therefore it's recommended as the primary Ethernet communication channel. Port 7236 is suited as a sniffer or diagnostics channel. The priorities are only relevant when both channels are used simultaneously.

The protocol that is used for Ethernet communication is a subset of the *horch* protocol (http://www.oertel-halle.de/horch/index.html), which is ASCII based. The TCP streams to and from the gateway have different formats and are documented in the following section.

4.1.1 TCP stream from PC to gateway

All protocol formats are specified using Regex.

Commands for packet transmission

A CAN Tx packet consists of a standard or extended identifier followed by 0 to 8 data bytes.

A standard identifier (11 bit) is initiated by "w ", while an extended identifier (29 bit) is initiated by "W ".

All values are in hexadecimal format. Hexadecimal numbers must be prefixed by "0x" and should be written in lower case, but upper case hexadecimal characters are tolerated.

The CAN remote transmit request (RTR) flag is marked with "r " and consists of the identifier with up to 8 dummy data bytes, used to encode the length of the requested data.

Format of CAN packets with a standard identifier (11 bit):

```
w (r )?0x[a-f0-9]{1,3} (0x[a-f0-9]{1,2} ){0,8}\n
```

Format of CAN packets with an extended identifier (29 bit):

```
W (r )?0x[a-f0-9]{1,8} (0x[a-f0-9]{1,2} ){0,8}\n
```

Examples

Outgoing CAN data packet with standard identifier and 8 data bytes:

```
"w 0x60b 0x80 0xff 0x60 0x0 0x0 0x0 0x4 0x5 \n"
```

Outgoing CAN data packet with extended identifier and 1 data byte:

```
"W 0x8a 0x0 \n"
```

Outgoing remote transmit request for 8 data bytes with standard identifier:

```
"w r 0x480 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 \n"
```

Outgoing CAN data packet with extended identifier (with maximum value) and 8 data bytes:

```
"W 0x1fffffff 0xff 0xff 0xa0 0xb 0xc 0x40 0x4 0x5 \n"
```

Command to set the CAN Bitrate

The following command allows to set the CAN Bitrate in kBit/s:

```
b (125|250|500|1000) \n
```

Example for setting CAN Bitrate to 500kBit/s:

```
"b 500 \n"
```

4.1.2 TCP stream from gateway to PC

Format for CAN packet reception

A CAN Rx packet consists of an optional timestamp, the CAN identifier, two description flags and 0 to 8 data bytes or a data length field in case of a remote transmit request. Decimal numbers may have leading spaces, but no leading zeroes.

Format overview:

```
[Timestamp] <decimalIdentifer>/<hexIdentifier> : <flags> : <dataBytes>
```

Fields:

- `[Timestamp]` is in seconds, has 12 digits before the decimal point (with spaces as placeholder) and 6 digits after the decimal point. The timestamp is added for compatibility purpose only and is always set to 0.000000.
- `<decimalIdentifer>` is a string of at least 4 chars and a maximum length of 9 chars. If the numeric value is below 1000, the value is padded with preceding spaces.
- `<hexIdentifier>` is the identifier in hexadecimal format (upper case). It has at least 3 digits "0x000" to "0xFFF" and a maximum of 8 digits with a maximum value of "0x1FFFFFFF".
- `<flags>`:
  - s|x: standard (11 bit) or extended (29 bit) frame format.
  - D|R: data or remote frame
- `<dataBytes>` are always hexadecimal (upper case), without the "0x" prefix, and space separated.

Format of CAN data packets:

```
\<[0-9]{1,9}/0x[A-F0-9]{3,8} : (s|x)D : ([A-F0-9]{2} ){0,8}
\r\n
```

Format of CAN remote transmit requests:

```
\<[0-9]{1,9}/0x[A-F0-9]{3,8} : (s|x)R : \( [0-9]{1} \) \r\n
```

There is a legacy format extension for Timestamp. The following prefix can be added to every packet:

```
\<[0-9]{1,12}.[0-9]{6}(  )
```

Examples

Incoming CAN packet with 8 data bytes:

```
"1434/0x59A : sD : 4F 01 10 00 00 00 00 00 \r\n"
```

Incoming CAN packet with extended identifier and 2 data bytes:

```
"8272/0x2050 : xD : FF 00 \r\n"
```

Incoming CAN packet with 0 data bytes; take note of the 3 leading spaces before the decimal identifier 0:

```
"   0/0x000 : sD : \r\n"
```

Incoming remote transmit request:

```
" 520/0x208 : sR : (length=8) \r\n"
```

Incoming CAN packet with extended header and 2 data bytes:

```
"536870911/0x1FFFFFFF : xD : FF 00 \r\n"
```

4.2 USB protocol

The gateway connects to USB as a HID device and can transmit USB reports with 64 data bytes every 1ms.

The format of the USB report:

Byte #	Description
0	Report ID (always 0x3F)
1	Bit 0 … 3: Number of packets in this report Bit 4 … 7: Packet type
2 … 3	Sequence number of the USB report (little endian)
4 … 63	Payload for 0 … n packets All packets in this payload must be of the same packet type The length of each packet is determined by the packet type

Packet type:

Packet Type	Description	Length in Byte
1	CAN packet with standard 11 bit identifier	10
2	CAN packet with standard 11 bit identifier, with timestamp	14
3	CAN packet with extended 29 bit identifier	14
4	CAN packet with extended 29 bit identifier, with timestamp	18
5 … 15	reserved	-

Format for CAN packets with 11 bit standard identifier:

Byte #	Description
0 … 1	Bit 0 … 10: Standard CAN identifier Bit 11: RTR flag Bit 12 … 15: Data Length
2 … 9	Data (8 byte)
10 … 13	Timestamp (optional, 4 byte)

Format for CAN packets with 29 bit extended identifier:

Byte #	Description
0	Data Length
1	Flags Bit 0: Extended ID Bit 1: RTR
2 … 5	Identifier
6 … 13	Data (8 byte)
14 … 17	Timestamp (optional, 4 byte)

4.3 CANopen communication via *horch*

The *Smart I/O* can be accessed as CANopen device via node ID 126 (default node ID, can be changed). Only SDOs are used for configuration.

4.3.1 Device Configuration

Parameters like IP Address Settings, CAN Bitrate, and CAN Filters are stored immediately on change.

Parameters can be reset to defaults by a command.

A firmware update won't reset the existing parameters.

4.3.2 CANopen SDO protocol

Knowledge of the CiA CANopen specifications is necessary. All CANopen devices have an Object Dictionary with data entries called Objects. These objects are addressed by a 16 bit Index and an 8 bit Subindex. There are rules and conventions for index/subindex usage. One way to access the values (objects) of the object dictionary is the SDO protocol, which will be explained in this chapter.

SDO packets have a constant data length of 8 data bytes.

SDO packet format (from/to a device):

Field	Description
Identifier	SDO request: 0x600 + Node ID SDO response: 0x580 + Node ID
Byte 0	SDO header byte
Byte 1 … 2	Object Index
Byte 3	Object Subindex
Byte 4 … 7	Data value of 1, 2 or 4 bytes length Error code of 4 bytes length

Values with multiple bytes are little endian (least significant byte of the data is placed in the first position).

SDO header byte for write requests/responses:

Header Byte	Description
0x23	write request with 4 byte value
0x2b	write request with 2 byte value
0x2f	write request with 1 byte value
0x60	write response to a successful write (any value type)
0x80	response to a failed write e.g. a read-only or non-existing value

SDO header byte for read requests/responses:

Header Byte	Description
0x40	read request (for any value type)
0x43	read response with a 4 byte value
0x4b	read response with a 2 byte value
0x4f	read response with a 1 byte value
0x80	response to a failed read e.g. a non-existing value

The response to a failed read/write is called an SDO Abort. It has an index and subindex. However the data value is replaced by an error code. The error codes are defined in the CiA 301 specification, however manufacturer specific codes are possible as well.

4.4 Write/Read an object over *horch* protocol

Example 1: Write

Field	Value	Description
Node ID	1	Device 1
Object Index	0x6040	Controlword
Object Subindex	0	Controlword
Data Length	2	INTEGER16
Data Value	0x76	Shutdown

TCP/IP *horch* Request String:

```
"w 0x601 0x2b 0x40 0x60 0x00 0x76 0x00 0x00 0x00 \n"
```

Value	Description
w	*horch* command sending a standard CAN message
0x601	COB-ID: 0x600 + Node ID
0x2b	SDO Protocol header: write 2 Byte
0x40	Object Index (low byte)
0x60	Object Index (high byte)
0x00	Object Subindex
0x76	Data Byte 0 (LSB)
0x00	Data Byte 1
0x00	Data Byte 2
0x00	Data Byte 3 (MSB)
\n	End of telegram

Response string:

```
"1409/0x581 : sD : 60 40 60 00 00 00 00 00 \r\n"
```

Value	Description
1409/0x581	COBID decimal/hexadecimal: 0x580 + Node ID
sD	CAN data packet with a standard 11 bit Identifier
60	SDO Protocol header: write successful
40	Object Index (low byte)
60	Object Index (high byte)
00	Object Subindex
00	Data Byte 0 (LSB)
00	Data Byte 1
00	Data Byte 2
00	Data Byte 3 (MSB)
\r\n	End of telegram

Example 2: Read

Field	Value	Description
Node ID	1	Device 1
Object Index	0x6041	Statusword
Object Subindex	0	Statusword
Data Length	2	INTEGER16
Data Value	0	No data

TCP/IP *horch* Request String:

```
"w 0x601 0x40 0x41 0x60 0x00 0x00 0x00 0x00 0x00 \n"
```

Value	Description
w	*horch* command sending a standard CAN message
0x601	COB-ID: 0x600 + Node ID
0x40	SDO Protocol header: read object
0x41	Object Index (low byte)
0x60	Object Index (high byte)
0x00	Object Subindex
0x00	Data Byte 0 (LSB)
0x00	Data Byte 1
0x00	Data Byte 2
0x00	Data Byte 3 (MSB)
\n	End of telegram

Response string:

```
"1409/0x581 : sD : 4b 41 60 00 37 10 00 00 \r\n"
```

Value	Description
1409/0x581	COBID decimal/hexadecimal: 0x580 + Node ID
sD	CAN Message with a standard 11 bit Identifier
4b	SDO Protocol header: read 2 byte successful
41	Object Index (low byte)
60	Object Index (high byte)
00	Object Subindex
37	Data Byte 0 (LSB)
10	Data Byte 1
00	Data Byte 2
00	Data Byte 3 (MSB)
\r\n	End of telegram

5. Ethernet Discovery & Configuration Protocol

A UDP based protocol allows discovering and configuration of *Smart I/O* in a local area network. The requests and responses are ASCII strings, sent as UDP broadcasts. The devices are identified by their MAC address. Server Port (Gateway) is 15000. Client Port (PC) is 15001.

All hexadecimal values must be sent in lower case.

5.1 Search Devices

The following command discovers all *Smart I/O* in the LAN (in the same broadcast domain).

Request:

```
BEH identify request\r\n
```

Response contains the MAC address, the firmware version and optionally a description (up to 16 characters). The firmware version consists of one digit for the major version and 2 digits for the minor version.

```
BEH identify request\r\nMAC: ([0-9a-f]{2}:){5}([0-9a-f]{2}) S:
[0-9]{3}( D: [\x20-\x7E]{1,16})?[ ]*\r\n
```

Example

Request:

```
BEH identify request\r\n
```

Response:

```
BEH identify request\r\nMAC: 00:09:77:0A:00:01 S: 101\r\n
```

MAC address: 00:09:77:0A:00:01

Firmware version: v1.01

5.2 Configure Device

The following command configures the IP settings.

The IP settings are:

- IP address
- Subnet mask
- Default Gateway
- DHCP enable/disable

The target device is addressed by its MAC address.

Format Overview:

```
MAC:<MAC_ADDR>:SN:<IP_ADDR>:<SN_MASK>:<DEF_GW>:<D>\r\n
```

Field	Description
<MAC_ADDR>	MAC Address (hexadecimal, lower case, no special characters)
<IP_ADDR>	IP Address (hexadecimal)
<SN_MASK>	Subnet Mask (hexadecimal)
<DEF_GW>	Default Gateway Address (hexadecimal)
<D>	DHCP enable (0: DHCP disabled; 1: DHCP enabled)

Regex format description:

```
MAC:[0-9a-f]{12}:SN:[0-9A-Fa-f]{8}:[0-9A-Fa-f]{8}:[0-9A-F]{8}:[01]
\r\n
```

The Parameter check is limited to a single parameter (no check if the whole parameter set is valid). The parameters are denied individually (the response may contain some old values).

The IP address is valid if it is in the range 1.0.0.0 (0x01000000) to 254.255.255.255 (0xfeffffff).

The Subnet Mask is valid if all set bits are on the left (MSB) and all cleared bits are on the right (LSB). Values 0.0.0.0 and 255.255.255.255 are also valid.

The Default Gateway must be a valid IP address or 0.0.0.0.

DHCP enable must be 0 or 1.

The response message will reflect the actual IP settings.

Example

Request:

```
MAC:0009770a0001:SN:0a64205a:ff000000:0a640001:1\r\n
```

Response:

```
MAC:0009770a0001:SN:0a64205a:ff000000:0a640001:1\r\n
```

6. CANopen Object Dictionary

6.1 Conventions

Data Types:

Type	Description	Bit size
U8	UNSIGNED8	8
S8	SIGNED8	8
U16	UNSIGNED16	16
S16	SIGNED16	16
U32	UNSIGNED32	32
S32	SIGNED32	32

Access Types:

Access Type	Description
RO	Read-only
WO	Write-only
RW	Read and Write

6.2 Object Dictionary Structure

Index	Object Dictionary Area
0x1000 … 0x1FFF	Communication Area
0x2000 … 0x5FFF	Manufacturer Specific Area
0x6000 … 0x9FFF	Profile Specific Area

6.3 Communication Segment

The Communication Segment (Range 0x1000 … 0x1FFF) is not specifically addressed. Reference is made to the specification, particularly ETG.1000.6, ETG.5001.1 and ETG.5001.3.

Index	Subidx	Type	Access	Description
0x1000	0	U32	RO	Device Type (0x00BE0000)
0x1011	0	U8	RO	Restore Default Parameters Object: Number of Entries = 1
0x1011	1	U32	RW	Restore All Default Parameters Note: this won't impact the MAC Address, Serial Number and License Data!
0x1018	0	U8	RO	Identity Object: Number of Entries = 4
0x1018	1	U32	RO	Vendor ID (0x000001A1)
0x1018	2	U32	RO	Product Code (0x0000003B)
0x1018	3	U32	RO	Revision Number (0x00000001)
0x1018	4	U32	RO	Serial Number

6.4 Manufacturer Specific Segment

Index	Subidx	Type	Access	Description
0x2000	0	U16	WO	Command Object • 0x0001: Reset Device • 0x0002: Clear Statistics, Errors and Status Counters • 0x0004: Load Default Parameters • 0x0005: Load Default Parameters and Reset Device • 0x0010: Enable Sniffer Mode (passive role on the CAN bus) • 0x0011: Enable Gateway Mode (active role on the CAN bus) • 0x8000: Disable CAN-to-Gateway Access • 0x8001: Enable CAN-to-Gateway Access
0x2001	0	U32	RO	Status Object • Bit 0: EEPROM Error (0: ok, 1: error) • Bit 1: CAN Rx Error Status (0: REC = 0, 1: REC > 0) • Bit 2: CAN Tx Error Status (0: TEC = 0, 1: TEC > 0) • Bit 3 … 4: reserved • Bit 5: CAN Rx Buffer Status (0: ok, 1: overflow) • Bit 6: reserved • Bit 7: USB Rx Buffer status (0: ok, 1: overflow) • Bit 8: Ethernet Control (Port 7235) Rx Buffer Status (0: ok, 1: overflow) • Bit 9: Ethernet Diagnostics (Port 7236) Rx Buffer Status (0: ok, 1: overflow) • Bit 10: CAN Tx Buffer Status (0: ok, 1: overflow) • Bit 11: reserved • Bit 12: USB Tx Buffer status (0: ok, 1: overflow) • Bit 13: Ethernet Control (Port 7235) Tx Buffer Status (0: ok, 1: overflow) • Bit 14: Ethernet Diagnostics (Port 7236) Tx Buffer Status (0: ok, 1: overflow) • Bit 15 ... 26: reserved • Bit 27: Developer Mode (0: disabled, 1: enabled - not suited for production environments) • Bit 28: CAN-to-Gatway Access Mode (0: disabled, 1: enabled) • Bit 29: Sniffer Mode (0: active role on the CAN bus, 1: passive role on the CAN bus) • Bit 30: Reduced Mode (0: disabled, 1: enabled – Smart I/O Gateway can only be operated in Sniffer Mode) • Bit 31: CS Flag (0: ok, 1: manufacturer configuration is faulty - not suited for production environments)
0x2009	0	U16	RW	CAN SDO Inhibit Time in ms Range: 0 … 100 (0 = disabled) When enabled, the gateway will prevent sending SDO requests to a specific node for the duration of the inhibit time, while a SDO response is still pending. This prevents a faulty CAN master from flooding the CAN bus.

0x2010	0	U32	RO	Reduced Mode Reason Bit 0: EEPROM Error Bit 1 ... 3: reserved Bit 4: Invalid Serial Number Bit 5 ... 7: reserved Bit 8: No MAC Address Bit 9: Invalid MAC Address Domain Bit 10: Invalid MAC Address Bit 11 ... 30: reserved Bit 31: CS disabled Any value other than 0 will force the gateway into Sniffer Mode. This is usually an indication of a severe failure. Please contact Brunner support.
0x2125	0	U32	RO	Actual IP Address (little endian) e.g. 0x7F000001 = “127.0.0.1”
0x2138	0	U8	RO	CANopen PDO Filter: Number of Entries = 4
0x2138	1	U32	RW	CANopen PDO Filter for Nodes 0 … 31 (bit masked) • 0: filter on (PDO will be dropped) • 1: filter off (PDO accepted)
0x2138	2	U32	RW	CANopen PDO Filter for Nodes 32 … 63 (bit masked) • 0: filter on (PDO will be dropped) • 1: filter off (PDO accepted)
0x2138	3	U32	RW	CANopen PDO Filter for Nodes 64 … 95 (bit masked) • 0: filter on (PDO will be dropped) • 1: filter off (PDO accepted)
0x2138	4	U32	RW	CANopen PDO Filter for Nodes 96 … 127 (bit masked) • 0: filter on (PDO will be dropped) • 1: filter off (PDO accepted)
0x3000	0	U8	RO	Hardware Version • 0: Hardware Version A • 1: Hardware Version B • 2: Hardware Version C • ...
0x3010	0	U16	RO	Firmware Version • Byte 0: Minor Version • Byte 1: Major Version e.g. 0x0101: Firmware Version v1.01
0x3011	0	U16	RO	Firmware Build Variant: must always be 1 (B001)
0x3012	0	U32	RO	Firmware Build Timestamp • Bit 0 ... 5: Minute • Bit 6 ... 10: Hour • Bit 11 ... 15: Day • Bit 16 ... 19: Month • Bit 20 ... 31: Year
0x3013	0	U8	RO	Firmware Build Suffix
0x3020	0	U32	RO	Device Uptime in s
0x31F0	0	U8	RO	Discovery Description: Number of Entries = 4
0x31F0	1	U32	RW	Discovery Description Characters 1-4

0x31F0	2	U32	RW	Discovery Description Characters 5-8
0x31F0	3	U32	RW	Discovery Description Characters 9-12
0x31F0	4	U32	RW	Discovery Description Characters 13-16 Optional Discovery Description used by the Discovery Service. Each byte represents an ASCII character. Only printable characters (0x20 ... 0x7E) are permitted. Upon writing 0x31F0.4 the Discovery Description will be evaluated and stored into EEPROM. A device reset is required for the new Discovery Description to take effect. To disable the Discovery Description, write 0x20202020 to all of the 4 Objects.
0x3200	0	U8	RW	CAN Baudrate Index • 0: 1000 kbps • 1: 800 kbps (not yet supported) • 2: 500 kbps • 3: 250 kbps • 4: 125 kbps • 5: 100 kbps (not yet supported) • 6: 50 kbps (not yet supported) • 7: 20 kbps (not yet supported) • 8: 10 kbps (not yet supported)
0x3201	0	U16	RW	CAN Baudrate in kbps Supported Baudrates: • 1000 • 500 • 250 • 125
0x3202	0	U16	RW	CAN Configuration Flags • Bit 0: CAN-to-Gateway Access (0: disabled, 1: enabled) • Bit 1 ... 15: reserved
0x3203	0	U8	RW	CAN Node ID Default: 126 Range: 1 ... 127 A device reset is required for the new Node ID to take effect.
0x3210	0	U8	RO	CAN Interface Status: Number of Entries = 6
0x3210	1	U8	RO	REC: CAN Rx Error Counter This error counter determines the CAN error state of the gateway.
0x3210	2	U8	RO	TEC: CAN Tx Error Counter This error counter determines the CAN error state of the gateway.

0x3210	3	U8	RO	CAN Status Register (CAN_STS) • Bit 0 ... 2: LEC (Last Error Code) • Bit 3: TX_OK • Bit 4: RX_OK • Bit 5: EPASS (1: CAN controller error level has reached error passive level) • Bit 6: EWARN (1: CAN controller error level has reached warning level) • Bit 7: BOFF (1: CAN controller has entered the bus OFF state) LEC Values: • 0 = No Error • 1 = Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed • 2 = Format Error: A fixed format part of the received frame has the wrong format • 3 = ACK Error: The message transmitted was not acknowledged by another node • 4 = BIT1 Error: A Bit 1 Error indicates that the device wanted to send a High level (logical 1) but the monitored bus value was Low (logical 0) • 5 = BIT0 Error: A Bit 0 Error indicates that the device wanted to send a Low level (logical 0), but the monitored bus value was High (logical 1) • 6 = CRC Error: The CRC checksum was incorrect in the received message, indicating that the calculated value received did not match the calculated CRC of the data • 7 = No Event: When the LEC bit shows this value, no CAN bus event was detected since this value was written to the LEC field
0x3210	4	U8	RO	BOFF: CAN Bus-Off Status • 0: The CAN controller is not in Bus-Off state • 1: The CAN controller is in Bus-Off state (TEC > 255) When the CAN Bus-Off state is asserted, the gateway will automatically disable the CAN controller and attempt to re-enable it after the recovery time has passed.
0x3210	5	U8	RO	EPASS: CAN Error Passive Status • 0: The CAN controller is in the Error Active state (TEC ≤ 127 and REC ≤ 127) • 1: The CAN controller is in the Error Passive state (TEC > 127 or REC > 127)
0x3210	6	U8	RO	EWARN: CAN Warning Status • 0: TEC and REC are below the error warning limit of 96 • 1: TEC, REC or both have reached the error warning limit of 96
0x3220	0	U8	RO	CAN Interface Stats: Number of Entries = 16
0x3220	1	U32	RO	BOFF Counter This counter is incremented whenever the BOFF condition is detected.

0x3220	2	U32	RO	EPASS Counter This counter is incremented whenever the EPASS condition is detected.
0x3220	3	U32	RO	EWARN Counter This counter is incremented whenever the EWARN condition is detected.
0x3220	4	U32	RO	Stuff Error Counter This counter is incremented whenever a Bit Stuffing error occurs.
0x3220	5	U32	RO	Format Error Counter This counter is incremented whenever a Format Error occurs.
0x3220	6	U32	RO	ACK Error Counter This counter is incremented whenever an ACK Error occurs.
0x3220	7	U32	RO	BIT1 Error Counter This counter is incremented whenever a BIT1 Error occurs.
0x3220	8	U32	RO	BIT0 Error Counter This counter is incremented whenever a BIT0 Error occurs.
0x3220	9	U32	RO	CRC Error Counter This counter is incremented whenever a CRC Error occurs.
0x3220	10	U32	RO	Write Tx Error Counter This counter is incremented whenever a Tx message couldn't be written to the CAN controller.
0x3220	11	U32	RO	Read Rx Error Counter This counter is incremented whenever a Rx message couldn't be read from the CAN controller.
0x3220	12	U8	RO	Tx FIFO Level Max The maximum number of Tx FIFO buffers in concurrent use.
0x3220	13	U8	RO	Rx FIFO Level Max The maximum number of Rx FIFO buffers in concurrent use.
0x3220	14	U32	RO	Average Bus Load Bit Counter The number of transmitted bits on the CAN bus. This uses the average number of stuffing bits. CAN messages due to retransmissions are not taken into account.
0x3220	15	U32	RO	Average Bus Load Bit Counter per Second The number of transmitted bits on the CAN bus during a period of one second. This uses the average number of stuffing bits. CAN messages due to retransmissions are not taken into account.
0x3220	16	U16	RO	Average Bus Load in % This uses the average number of stuffing bits. CAN messages due to retransmissions are not taken into account.
0x3250	0	U8	RO	CAN Channel State: Number of Entries = 2
0x3250	1	U8	RO	CAN Configured Flag • 0: CAN Channel isn't configured • 1: CAN Channel successfully configured

0x3250	2	U8	RO	CAN Connected Flag • 0: No connection to another CAN node • 1: Gateway is connected to another CAN node As part of the bootup sequence the gateway will send a CANopen bootup message. Whenever another CAN node acknowledges this message, the connected flag will be asserted. Also when a foreign CAN message is received this connected flag will be set.
0x3258	0	U8	RO	CAN Channel Tx Buffer: Number of Entries = 3
0x3258	1	U16	RO	Tx Buffer Size = 32
0x3258	2	U16	RO	Max Tx Buffer usage
0x3258	3	U32	RO	Tx Buffer Overflow Counter This counter is incremented whenever a CAN message is dropped because the Tx Buffer is full.
0x3259	0	U8	RO	CAN Channel Rx Buffer: Number of Entries = 3
0x3259	1	U16	RO	Rx Buffer Size = 32
0x3259	2	U16	RO	Max Rx Buffer usage
0x3259	3	U32	RO	Rx Buffer Overflow Counter This counter is incremented whenever a CAN message is dropped because the Rx Buffer is full.
0x3260	0	U8	RO	CAN Channel Statistics: Number of Entries = 14
0x3260	1	U32	RO	Transmitted CAN Tx Messages
0x3260	2	U32	RO	Transmitted CAN Tx Messages per Second
0x3260	3	U32	RO	Received CAN Rx Messages
0x3260	4	U32	RO	Received CAN Rx Messages per Second
0x3260	5	U32	RO	Ignored CAN Rx Messages This counter is incremented whenever a CAN Rx message is ignored. For instance when receiving a PDO which is dropped due to the PDO Filter setting (see object 0x2009).
0x3260	6	U32	RO	Ignored CAN Rx Messages per Second
0x3260	7	U32	RO	Tx PDO Counter Transmitted PDO Messages
0x3260	8	U32	RO	Tx PDO Counter per Second Transmitted PDO Messages per Second
0x3260	9	U32	RO	Rx PDO Counter Received PDO Messages
0x3260	10	U32	RO	Rx PDO Counter per Second Received PDO Messages per Second
0x3260	11	U32	RO	Tx SDO Counter Transmitted SDO Messages
0x3260	12	U32	RO	Tx SDO Counter per Second Transmitted SDO Messages per Second
0x3260	13	U32	RO	Rx SDO Counter Received SDO Messages
0x3260	14	U32	RO	Rx SDO Counter per Second Received SDO Messages per Second
0x3310	0	U8	RO	USB Interface Status: Number of Entries = 2

0x3310	1	U8	RO	Previous USB Tx Event • 0: None • 5: Tx Complete
0x3310	2	U8	RO	Previous USB Rx Event • 0: USB connected • 1: USB disconnected • 2: Rx Data available • 6: Error • 7: Suspend • 8: Resume • 9: HID Get Report
0x3320	0	U8	RO	USB Interface Statistics: Number of Entries = 16
0x3320	1	U32	RO	USB Connect Event Counter This counter is incremented whenever a USB Connect event occurs.
0x3320	2	U32	RO	USB Disconnect Event Counter This counter is incremented whenever a USB Disconnect event occurs.
0x3320	3	U32	RO	USB Suspend Event Counter This counter is incremented whenever a USB Suspend event occurs.
0x3320	4	U32	RO	USB Resume Event Counter This counter is incremented whenever a USB Resume event occurs.
0x3320	5	U32	RO	USB Tx Complete Event Counter This counter is incremented whenever a USB Tx Complete event occurs.
0x3320	6	U32	RO	USB Rx Available Event Counter This counter is incremented whenever a USB Rx Available event occurs.
0x3320	7	U32	RO	USB HID Get Report Event Counter This counter is incremented whenever a USB HID Get Report event occurs.
0x3320	8	U32	RO	USB Error Event Counter This counter is incremented whenever a USB Error event occurs.
0x3320	9	U32	RO	Ignored USB Tx Event Counter This counter is incremented whenever an unknown USB Tx Event occurs.
0x3320	10	U32	RO	Ignored USB Rx Event Counter This counter is incremented whenever an unknown USB Rx Event occurs.
0x3320	11	U32	RO	USB Tx Frame Counter This counter is incremented whenever a USB Tx Frame is sent.
0x3320	12	U32	RO	USB Tx Error Counter This counter is incremented whenever a HID device report couldn't be scheduled for transmission.
0x3320	13	U32	RO	USB Rx Frame Counter This counter is incremented whenever a USB Rx Frame has been received.
0x3320	14	U32	RO	USB Rx Invalid Length Counter This counter is incremented whenever a received data container has an unexpected length.

0x3320	15	U32	RO	USB Rx Invalid Report ID Counter This counter is incremented whenever a received data container has an unexpected Report ID.
0x3320	16	U32	RO	USB Rx Invalid Dataset Type Counter This counter is incremented whenever a received data container has an unexpected Dataset Type.
0x3350	0	U8	RO	USB Channel State: Number of Entries = 2
0x3350	1	U8	RO	USB Configured Flag • 0: USB Channel isn't configured • 1: USB Channel successfully configured
0x3350	2	U8	RO	USB Connected Flag • 0: No connection to a USB Host • 1: Gateway is connected to a USB Host
0x3358	0	U8	RO	USB Channel Tx Buffer: Number of Entries = 3
0x3358	1	U16	RO	USB Tx Buffer Size = 32
0x3358	2	U16	RO	Max USB Tx Buffer usage
0x3358	3	U32	RO	USB Tx Buffer Overflow Counter This counter is incremented whenever a CAN message is dropped because the Tx Buffer is full.
0x3359	0	U8	RO	USB Channel Rx Buffer: Number of Entries = 3
0x3359	1	U16	RO	USB Rx Buffer Size = 32
0x3359	2	U16	RO	Max USB Rx Buffer usage
0x3359	3	U32	RO	USB Rx Buffer Overflow Counter This counter is incremented whenever a CAN message is dropped because the Rx Buffer is full.
0x3360	0	U8	RO	USB Channel Statistics: Number of Entries = 14
0x3360	1	U32	RO	Transmitted CAN Tx Messages
0x3360	2	U32	RO	Transmitted CAN Tx Messages per Second
0x3360	3	U32	RO	Received CAN Rx Messages
0x3360	4	U32	RO	Received CAN Rx Messages per Second
0x3360	5	U32	RO	Ignored CAN Rx Messages Reserved for future use (always 0).
0x3360	6	U32	RO	Ignored CAN Rx Messages per Second Reserved for future use (always 0).
0x3360	7	U32	RO	Tx PDO Counter Transmitted PDO Messages
0x3360	8	U32	RO	Tx PDO Counter per Second Transmitted PDO Messages per Second
0x3360	9	U32	RO	Rx PDO Counter Received PDO Messages
0x3360	10	U32	RO	Rx PDO Counter per Second Received PDO Messages per Second
0x3360	11	U32	RO	Tx SDO Counter Transmitted SDO Messages
0x3360	12	U32	RO	Tx SDO Counter per Second Transmitted SDO Messages per Second
0x3360	13	U32	RO	Rx SDO Counter Received SDO Messages

0x3360	14	U32	RO	Rx SDO Counter per Second Received SDO Messages per Second
0x3400	0	U8	RO	Ethernet Interface Settings – MAC Address: Number of Entries = 6
0x3400	1	U8	RO	MAC Address Byte 0 (always 0x00)
0x3400	2	U8	RO	MAC Address Byte 1 (always 0x09)
0x3400	3	U8	RO	MAC Address Byte 2 (always 0x77)
0x3400	4	U8	RO	MAC Address Byte 3
0x3400	5	U8	RO	MAC Address Byte 4
0x3400	6	U8	RO	MAC Address Byte 5
0x3401	0	U8	RO	Ethernet Interface Settings – IP Settings: Number of Entries = 4
0x3401	1	U32	RW	Static IP Address (little endian) e.g. 0xC0A80002 = “192.168.0.2”
0x3401	2	U32	RW	Subnet Mask (little endian) e.g. 0xFFFF0000 = “255.255.0.0”
0x3401	3	U32	RW	Default Gateway (little endian) e.g. 0xC0A80001 = “192.168.0.1”
0x3401	4	U8	RW	DHCP Enable • 0: Static IP • 1: DHCP Auto IP will be used if DHCP is enabled but DHCP server isn’t available.
0x3410	0	U8	RO	Ethernet Interface Status: Number of Entries = 2
0x3410	1	U8	RO	Interface Status • 0: Interface down • 1: Interface up
0x3410	2	U8	RO	Link Status • 0: Link down • 1: Link up
0x3411	0	U8	RO	PoE Status • 0: not powered over Ethernet • 1: Power over Ethernet active
0x3420	0	U8	RO	Ethernet Interface Statistics: Number of Entries = 4
0x3420	1	U32	RO	Interface Up Counter This counter is incremented whenever the Ethernet Interface Up event occurs.
0x3420	2	U32	RO	Interface Down Counter This counter is incremented whenever the Ethernet Interface Down event occurs.
0x3420	3	U32	RO	Link Up Counter This counter is incremented whenever the Ethernet Link Up event occurs.
0x3420	4	U32	RO	Link Down Counter This counter is incremented whenever the Ethernet Link Down event occurs.
0x3421	0	U8	RO	Ethernet Link Statistics: Number of Entries = 12
0x3421	1	U32	RO	Ethernet Link: Transmitted Packets
0x3421	2	U32	RO	Ethernet Link: Received Packets
0x3421	3	U32	RO	Ethernet Link: Forwarded Packets

0x3421	4	U32	RO	Ethernet Link: Dropped Packets
0x3421	5	U32	RO	Ethernet Link: Checksum Error Counter
0x3421	6	U32	RO	Ethernet Link: Invalid Length Error Counter
0x3421	7	U32	RO	Ethernet Link: Out-of-Memory Error Counter
0x3421	8	U32	RO	Ethernet Link: Routing Error Counter
0x3421	9	U32	RO	Ethernet Link: Protocol Error Counter
0x3421	10	U32	RO	Ethernet Link: Error-in-Options Error Counter
0x3421	11	U32	RO	Ethernet Link: Misc Error Counter
0x3421	12	U32	RO	Ethernet Link: Cache Hit Counter
0x3422	0	U8	RO	Ethernet ARP Statistics: Number of Entries = 12
0x3422	1	U32	RO	Ethernet ARP: Transmitted Packets
0x3422	2	U32	RO	Ethernet ARP: Received Packets
0x3422	3	U32	RO	Ethernet ARP: Forwarded Packets
0x3422	4	U32	RO	Ethernet ARP: Dropped Packets
0x3422	5	U32	RO	Ethernet ARP: Checksum Error Counter
0x3422	6	U32	RO	Ethernet ARP: Invalid Length Error Counter
0x3422	7	U32	RO	Ethernet ARP: Out-of-Memory Error Counter
0x3422	8	U32	RO	Ethernet ARP: Routing Error Counter
0x3422	9	U32	RO	Ethernet ARP: Protocol Error Counter
0x3422	10	U32	RO	Ethernet ARP: Error-in-Options Error Counter
0x3422	11	U32	RO	Ethernet ARP: Misc Error Counter
0x3422	12	U32	RO	Ethernet ARP: Cache Hit Counter
0x3423	0	U8	RO	Ethernet IP Statistics: Number of Entries = 12
0x3423	1	U32	RO	Ethernet IP: Transmitted Packets
0x3423	2	U32	RO	Ethernet IP: Received Packets
0x3423	3	U32	RO	Ethernet IP: Forwarded Packets
0x3423	4	U32	RO	Ethernet IP: Dropped Packets
0x3423	5	U32	RO	Ethernet IP: Checksum Error Counter
0x3423	6	U32	RO	Ethernet IP: Invalid Length Error Counter
0x3423	7	U32	RO	Ethernet IP: Out-of-Memory Error Counter
0x3423	8	U32	RO	Ethernet IP: Routing Error Counter
0x3423	9	U32	RO	Ethernet IP: Protocol Error Counter
0x3423	10	U32	RO	Ethernet IP: Error-in-Options Error Counter
0x3423	11	U32	RO	Ethernet IP: Misc Error Counter
0x3423	12	U32	RO	Ethernet IP: Cache Hit Counter
0x3424	0	U8	RO	Ethernet ICMP Statistics: Number of Entries = 12
0x3424	1	U32	RO	Ethernet ICMP: Transmitted Packets
0x3424	2	U32	RO	Ethernet ICMP: Received Packets
0x3424	3	U32	RO	Ethernet ICMP: Forwarded Packets
0x3424	4	U32	RO	Ethernet ICMP: Dropped Packets
0x3424	5	U32	RO	Ethernet ICMP: Checksum Error Counter
0x3424	6	U32	RO	Ethernet ICMP: Invalid Length Error Counter

0x3424	7	U32	RO	Ethernet ICMP: Out-of-Memory Error Counter
0x3424	8	U32	RO	Ethernet ICMP: Routing Error Counter
0x3424	9	U32	RO	Ethernet ICMP: Protocol Error Counter
0x3424	10	U32	RO	Ethernet ICMP: Error-in-Options Error Counter
0x3424	11	U32	RO	Ethernet ICMP: Misc Error Counter
0x3424	12	U32	RO	Ethernet ICMP: Cache Hit Counter
0x3425	0	U8	RO	Ethernet UDP Statistics: Number of Entries = 12
0x3425	1	U32	RO	Ethernet UDP: Transmitted Packets
0x3425	2	U32	RO	Ethernet UDP: Received Packets
0x3425	3	U32	RO	Ethernet UDP: Forwarded Packets
0x3425	4	U32	RO	Ethernet UDP: Dropped Packets
0x3425	5	U32	RO	Ethernet UDP: Checksum Error Counter
0x3425	6	U32	RO	Ethernet UDP: Invalid Length Error Counter
0x3425	7	U32	RO	Ethernet UDP: Out-of-Memory Error Counter
0x3425	8	U32	RO	Ethernet UDP: Routing Error Counter
0x3425	9	U32	RO	Ethernet UDP: Protocol Error Counter
0x3425	10	U32	RO	Ethernet UDP: Error-in-Options Error Counter
0x3425	11	U32	RO	Ethernet UDP: Misc Error Counter
0x3425	12	U32	RO	Ethernet UDP: Cache Hit Counter
0x3426	0	U8	RO	Ethernet TCP Statistics: Number of Entries = 12
0x3426	1	U32	RO	Ethernet TCP: Transmitted Packets
0x3426	2	U32	RO	Ethernet TCP: Received Packets
0x3426	3	U32	RO	Ethernet TCP: Forwarded Packets
0x3426	4	U32	RO	Ethernet TCP: Dropped Packets
0x3426	5	U32	RO	Ethernet TCP: Checksum Error Counter
0x3426	6	U32	RO	Ethernet TCP: Invalid Length Error Counter
0x3426	7	U32	RO	Ethernet TCP: Out-of-Memory Error Counter
0x3426	8	U32	RO	Ethernet TCP: Routing Error Counter
0x3426	9	U32	RO	Ethernet TCP: Protocol Error Counter
0x3426	10	U32	RO	Ethernet TCP: Error-in-Options Error Counter
0x3426	11	U32	RO	Ethernet TCP: Misc Error Counter
0x3426	12	U32	RO	Ethernet TCP: Cache Hit Counter
0x3450	0	U8	RO	Ethernet Control Channel State: Number of Entries = 2
0x3450	1	U8	RO	Ethernet Control Channel Configured Flag • 0: Ethernet Control Channel isn't configured • 1: Ethernet Control Channel successfully configured
0x3450	2	U8	RO	Ethernet Control Channel Connected Flag • 0: No connection on port 7235 • 1: TCP connection on port 7235 established
0x3458	0	U8	RO	Ethernet Control Channel Tx Buffer: Number of Entries = 3
0x3458	1	U16	RO	Ethernet Control Channel Tx Buffer Size = 32
0x3458	2	U16	RO	Max Ethernet Control Channel Tx Buffer usage

0x3458	3	U32	RO	Ethernet Control Channel Tx Buffer Overflow Counter This counter is incremented whenever a CAN message is dropped because the Tx Buffer is full.
0x3459	0	U8	RO	Ethernet Control Channel Rx Buffer: Number of Entries = 3
0x3459	1	U16	RO	Ethernet Control Channel Rx Buffer Size = 32
0x3459	2	U16	RO	Max Ethernet Control Channel Rx Buffer usage
0x3459	3	U32	RO	Ethernet Control Channel Rx Buffer Overflow Counter This counter is incremented whenever a CAN message is dropped because the Rx Buffer is full.
0x3460	0	U8	RO	Ethernet Control Channel Statistics: Number of Entries = 14
0x3460	1	U32	RO	Transmitted CAN Tx Messages
0x3460	2	U32	RO	Transmitted CAN Tx Messages per Second
0x3460	3	U32	RO	Received CAN Rx Messages
0x3460	4	U32	RO	Received CAN Rx Messages per Second
0x3460	5	U32	RO	Ignored CAN Rx Messages Reserved for future use (always 0).
0x3460	6	U32	RO	Ignored CAN Rx Messages per Second Reserved for future use (always 0).
0x3460	7	U32	RO	Tx PDO Counter Transmitted PDO Messages
0x3460	8	U32	RO	Tx PDO Counter per Second Transmitted PDO Messages per Second
0x3460	9	U32	RO	Rx PDO Counter Received PDO Messages
0x3460	10	U32	RO	Rx PDO Counter per Second Received PDO Messages per Second
0x3460	11	U32	RO	Tx SDO Counter Transmitted SDO Messages
0x3460	12	U32	RO	Tx SDO Counter per Second Transmitted SDO Messages per Second
0x3460	13	U32	RO	Rx SDO Counter Received SDO Messages
0x3460	14	U32	RO	Rx SDO Counter per Second Received SDO Messages per Second
0x3490	0	U8	RO	Ethernet Diagnostics Channel State: Number of Entries = 2
0x3490	1	U8	RO	Ethernet Diagnostics Channel Configured Flag • 0: Ethernet Diagnostics Channel isn't configured • 1: Ethernet Diagnostics Channel successfully configured
0x3490	2	U8	RO	Ethernet Diagnostics Channel Connected Flag • 0: No connection on port 7236 • 1: TCP connection on port 7236 established
0x3498	0	U8	RO	Ethernet Diagnostics Channel Tx Buffer: Number of Entries = 3
0x3498	1	U16	RO	Ethernet Diagnostics Channel Tx Buffer Size = 32
0x3498	2	U16	RO	Max Ethernet Diagnostics Channel Tx Buffer usage
0x3498	3	U32	RO	Ethernet Diagnostics Channel Tx Buffer Overflow Counter This counter is incremented whenever a CAN message is dropped because the Tx Buffer is full.
0x3499	0	U8	RO	Ethernet Diagnostics Channel Rx Buffer: Number of Entries = 3

0x3499	1	U16	RO	Ethernet Diagnostics Channel Rx Buffer Size = 32
0x3499	2	U16	RO	Max Ethernet Diagnostics Channel Rx Buffer usage
0x3499	3	U32	RO	Ethernet Diagnostics Channel Rx Buffer Overflow Counter This counter is incremented whenever a CAN message is dropped because the Rx Buffer is full.
0x34A0	0	U8	RO	Ethernet Diagnostics Channel Statistics: Number of Entries = 14
0x34A0	1	U32	RO	Transmitted CAN Tx Messages
0x34A0	2	U32	RO	Transmitted CAN Tx Messages per Second
0x34A0	3	U32	RO	Received CAN Rx Messages
0x34A0	4	U32	RO	Received CAN Rx Messages per Second
0x34A0	5	U32	RO	Ignored CAN Rx Messages Reserved for future use (always 0).
0x34A0	6	U32	RO	Ignored CAN Rx Messages per Second Reserved for future use (always 0).
0x34A0	7	U32	RO	Tx PDO Counter Transmitted PDO Messages
0x34A0	8	U32	RO	Tx PDO Counter per Second Transmitted PDO Messages per Second
0x34A0	9	U32	RO	Rx PDO Counter Received PDO Messages
0x34A0	10	U32	RO	Rx PDO Counter per Second Received PDO Messages per Second
0x34A0	11	U32	RO	Tx SDO Counter Transmitted SDO Messages
0x34A0	12	U32	RO	Tx SDO Counter per Second Transmitted SDO Messages per Second
0x34A0	13	U32	RO	Rx SDO Counter Received SDO Messages
0x34A0	14	U32	RO	Rx SDO Counter per Second Received SDO Messages per Second
0x3800	0	U8	RO	Digital Inputs: Number of Entries = 2
0x3800	1	U32	RO	Digital Inputs 1-32 • Bit 0: Digital Input 1 • Bit 1: Digital Input 2 • ... • Bit 31: Digital Input 32 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)

0x3800	2	U32	RO	Digital Inputs 33-50, Misc Inputs • Bit 0: Digital Input 33 • Bit 1: Digital Input 34 • ... • Bit 17: Digital Input 50 • Bit 18: Encoder 1 Count Up • Bit 19: Encoder 1 Count Down • Bit 20: Encoder 1 Pushbutton • Bit 21: Encoder 2 Count Up • Bit 22: Encoder 2 Count Down • Bit 23: Encoder 2 Pushbutton • Bit 24 ... 31: reserved Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x3810	0	U8	RO	Digital Outputs: Number of Entries = 1
0x3810	1	U8	RW	Digital Outputs 1-4 • Bit 0: Digital Output 1 • Bit 1: Digital Output 2 • Bit 2: Digital Output 3 • Bit 3: Digital Output 4 • Bit 4 ... 7: reserved (always 0) Bit Value • 0: Output Low (GND) • 1: Output High (EXT_VCC)
0x3811	0	U8	RO	Digital Output Status Feedback: Number of Entries = 1
0x3811	1	U8	RO	Digital Output Status Feedback • Bit 0: Digital Output 1-2 Status Feedback • Bit 1: Digital Output 3-4 Status Feedback Bit Value • 0: OK • 1: Output Overload
0x3820	0	U8	RO	Raw Analog Input Values: Number of Entries = 4
0x3820	1	U16	RO	Raw Analog Input 1 Raw 12 bit Value • 0: 0V • 4095: 3.3V
0x3820	2	U16	RO	Raw Analog Input 2 Raw 12 bit Value • 0: 0V • 4095: 3.3V
0x3820	3	U16	RO	Raw Analog Input 3 Raw 12 bit Value • 0: 0V • 4095: 3.3V
0x3820	4	U16	RO	Raw Analog Input 4 Raw 12 bit Value • 0: 0V • 4095: 3.3V
0x3821	0	U8	RO	Filtered Analog Input Values: Number of Entries = 4

0x3821	1	U16	RO	Filtered Analog Input 1 Filtered 12 bit Value • 0: 0V • 4095: 3.3V
0x3821	2	U16	RO	Filtered Analog Input 2 Filtered 12 bit Value • 0: 0V • 4095: 3.3V
0x3821	3	U16	RO	Filtered Analog Input 3 Filtered 12 bit Value • 0: 0V • 4095: 3.3V
0x3821	4	U16	RO	Filtered Analog Input 4 Filtered 12 bit Value • 0: 0V • 4095: 3.3V
0x3822	0	U8	RO	User Analog Input Values: Number of Entries = 4
0x3822	1	U16	RO	User Analog Input 1 Value in mV
0x3822	2	U16	RO	User Analog Input 2 Value in mV
0x3822	3	U16	RO	User Analog Input 3 Value in mV
0x3822	4	U16	RO	User Analog Input 4 Value in mV
0x3832	0	U8	RO	User System Analog Input Values: Number of Entries = 2
0x3832	1	U16	RO	User Analog Input Hardware Version Hardware Version in mV
0x3832	2	U16	RO	User Analog Input CPU Temperature CPU Temperature in °C
0x3842	0	U8	RO	Power Analog Input Values: Number of Entries = 3
0x3842	1	U16	RO	+3.3V Power Supply Value in mV
0x3842	2	U16	RO	+5V Power Supply Value in mV
0x3842	3	U16	RO	External Power Supply Value in mV
0x3850	0	U8	RO	Incremental Encoder Position: Number of Entries = 2
0x3850	1	U16	RO	Incremental Encoder 1 Position
0x3850	2	U16	RO	Incremental Encoder 2 Position
0x6000	0	U8	RO	Digital Inputs (8 bit): Number of Entries = 7
0x6000	1	U8	RO	Digital Inputs 1-8 • Bit 0: Digital Input 1 • Bit 1: Digital Input 2 • ... • Bit 7: Digital Input 8 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)

0x6000	2	U8	RO	Digital Inputs 9-16 • Bit 0: Digital Input 9 • Bit 1: Digital Input 10 • ... • Bit 7: Digital Input 16 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x6000	3	U8	RO	Digital Inputs 17-24 • Bit 0: Digital Input 17 • Bit 1: Digital Input 18 • ... • Bit 7: Digital Input 24 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x6000	4	U8	RO	Digital Inputs 25-32 • Bit 0: Digital Input 25 • Bit 1: Digital Input 26 • ... • Bit 7: Digital Input 32 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x6000	5	U8	RO	Digital Inputs 33-40 • Bit 0: Digital Input 33 • Bit 1: Digital Input 34 • ... • Bit 7: Digital Input 40 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x6000	6	U8	RO	Digital Inputs 41-48 • Bit 0: Digital Input 41 • Bit 1: Digital Input 42 • ... • Bit 7: Digital Input 48 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)

0x6000	7	U8	RO	Digital Inputs 49-50, Misc Inputs • Bit 0: Digital Input 49 • Bit 1: Digital Input 50 • Bit 2: Encoder 1 Count Up • Bit 3: Encoder 1 Count Down • Bit 4: Encoder 1 Pushbutton • Bit 5: Encoder 2 Count Up • Bit 6: Encoder 2 Count Down • Bit 7: Encoder 2 Pushbutton Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x6100	0	U8	RO	Digital Inputs (16 bit): Number of Entries = 4
0x6100	1	U16	RO	Digital Inputs 1-16 • Bit 0: Digital Input 1 • Bit 1: Digital Input 2 • ... • Bit 15: Digital Input 16 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x6100	2	U16	RO	Digital Inputs 17-32 • Bit 0: Digital Input 17 • Bit 1: Digital Input 18 • ... • Bit 15: Digital Input 32 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x6100	3	U16	RO	Digital Inputs 33-40 • Bit 0: Digital Input 33 • Bit 1: Digital Input 34 • ... • Bit 15: Digital Input 40 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)

0x6100	4	U16	RO	Digital Inputs 41-50, Misc Inputs • Bit 0: Digital Input 41 • Bit 1: Digital Input 42 • ... • Bit 9: Digital Input 50 • Bit 10: Encoder 1 Count Up • Bit 11: Encoder 1 Count Down • Bit 12: Encoder 1 Pushbutton • Bit 13: Encoder 2 Count Up • Bit 14: Encoder 2 Count Down • Bit 15: Encoder 2 Pushbutton Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x6120	0	U8	RO	Digital Inputs (32 bit): Number of Entries = 4
0x6120	1	U32	RO	Digital Inputs 1-32 • Bit 0: Digital Input 1 • Bit 1: Digital Input 2 • ... • Bit 31: Digital Input 32 Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x6120	2	U32	RO	Digital Inputs 33-50, Misc Inputs • Bit 0: Digital Input 33 • Bit 1: Digital Input 34 • ... • Bit 17: Digital Input 50 • Bit 18: Encoder 1 Count Up • Bit 19: Encoder 1 Count Down • Bit 20: Encoder 1 Pushbutton • Bit 21: Encoder 2 Count Up • Bit 22: Encoder 2 Count Down • Bit 23: Encoder 2 Pushbutton • Bit 24 ... 31: reserved Bit Value • 0: Input open (HiZ) • 1: Input assterted (pulled to GND)
0x6200	0	U8	RO	Digital Outputs (8 bit): Number of Entries = 1
0x6200	1	U8	RW	Digital Outputs 1-4 • Bit 0: Digital Output 1 • Bit 1: Digital Output 2 • Bit 2: Digital Output 3 • Bit 3: Digital Output 4 • Bit 4 ... 7: reserved (always 0) Bit Value • 0: Output Low (GND) • 1: Output High (EXT_VCC)
0x6401	0	U8	RO	Analog Inputs: Number of Entries = 4

0x6401	1	U16	RO	Analog Input 1 Filtered Analog Input 1 (object 0x3821.1), left justified
0x6401	2	U16	RO	Analog Input 2 Filtered Analog Input 2 (object 0x3821.2), left justified
0x6401	3	U16	RO	Analog Input 3 Filtered Analog Input 3 (object 0x3821.3), left justified
0x6401	4	U16	RO	Analog Input 4 Filtered Analog Input 4 (object 0x3821.4), left justified

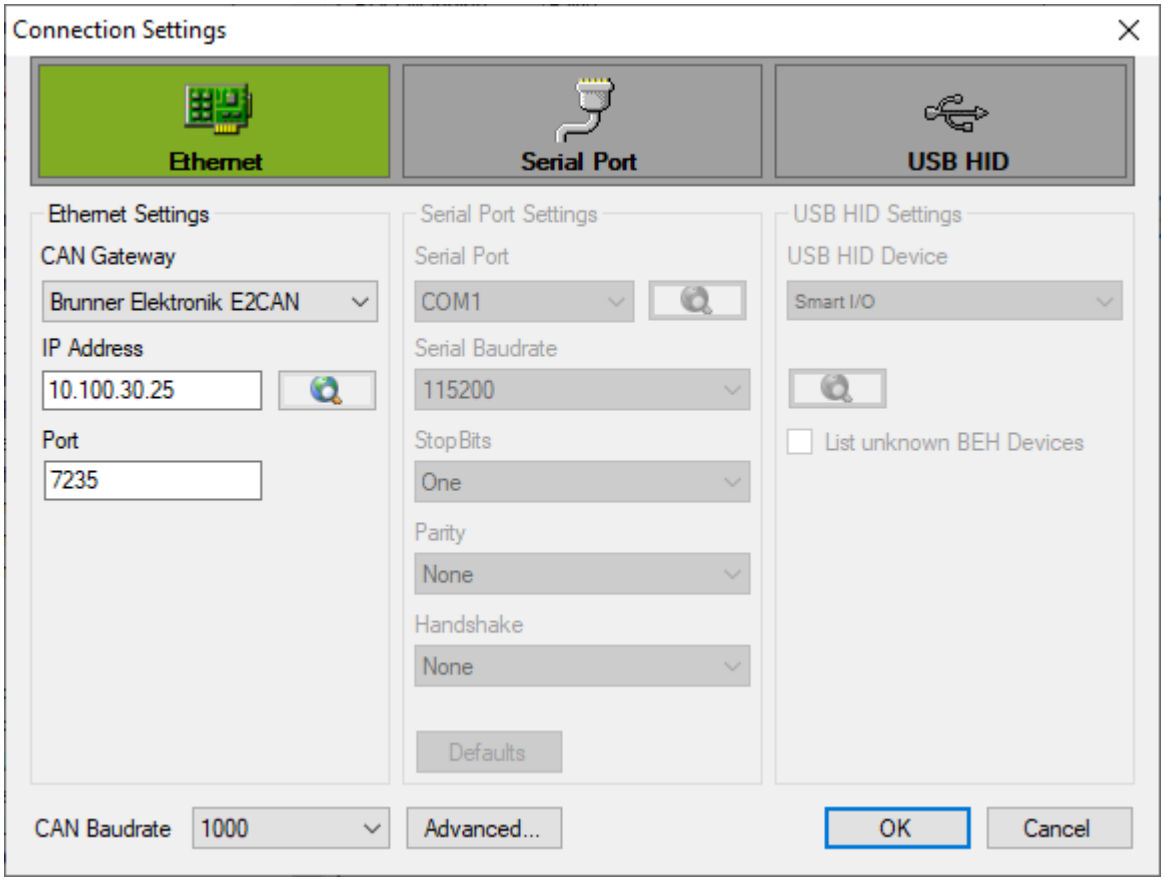
7. CANopen Commander

The CANopen Commander software allows configuration of CANopen devices. Different device-specific plugins are available that enable simple and quick configuration. One particular feature is flexible expandability by adding optional plugin modules.

The *Smart I/O* supports connections over USB and Ethernet.

7.1 Connection Settings

Open connection settings using the menu: File → Connection Settings…



7.1.1 Ethernet

Select "Ethernet" and configure the settings in "Ethernet Settings".

Click "OK": The settings are stored into the configuration file. If "Cancel" is used to exit the dialogue, changed parameters will be lost.

Click "Connect" in the "File" menu to connect to the device.

7.1.2 Serial Port

Not supported for this device.

Virtual COM Ports (like USB CDC) are listed as regular COM Ports.

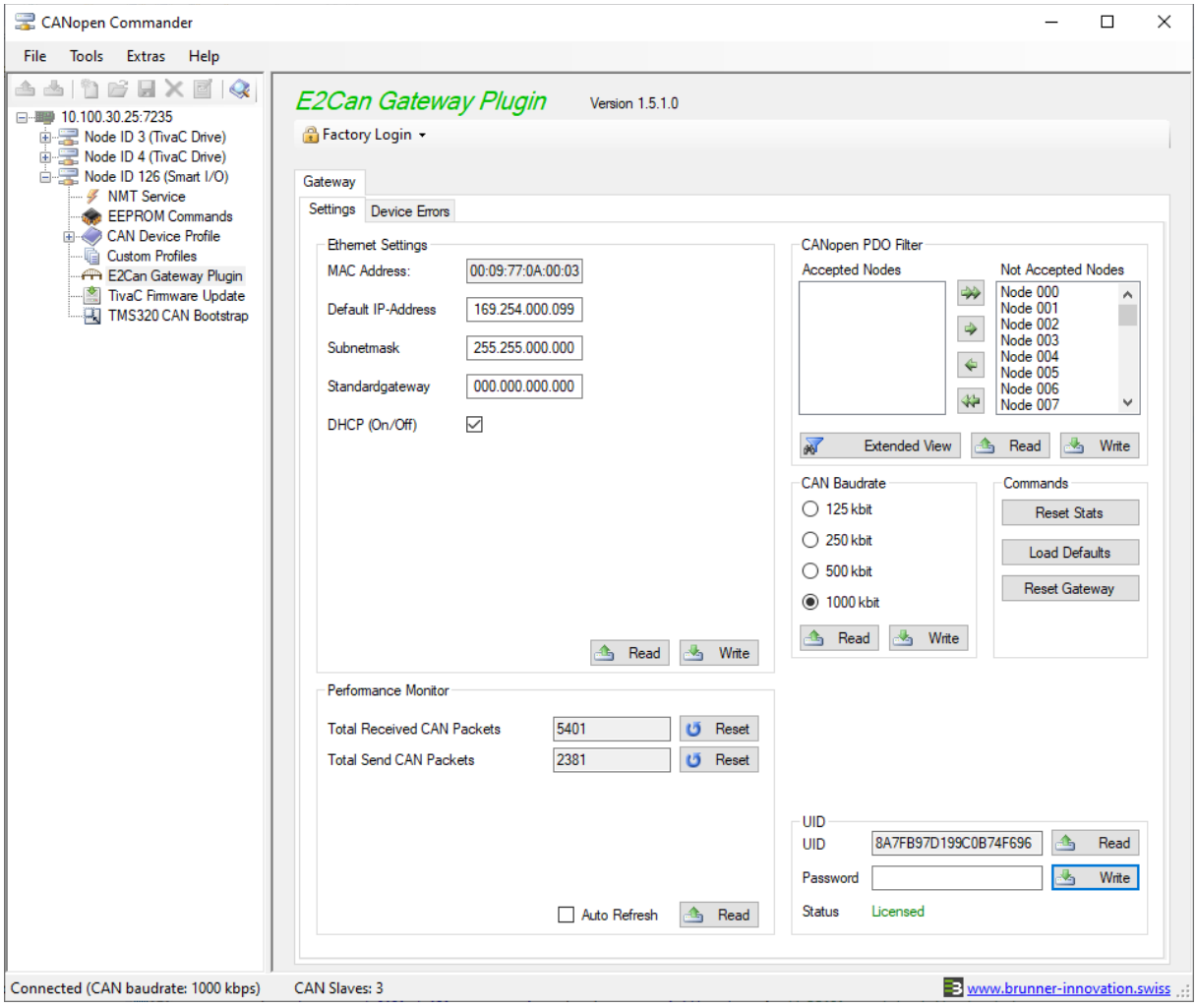
7.1.3 USB HID

Select "USB HID" and configure the settings in "USB Settings". The USB HID Device is listed in the USB Device dropdown list.

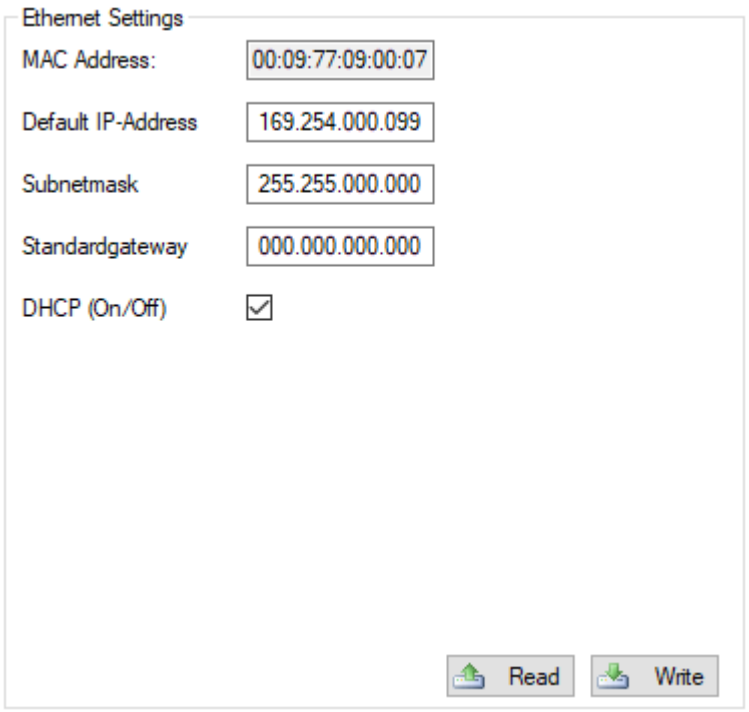
Click "OK": The settings are stored in a file. If "Cancel" is used to exit the dialogue, changed parameters will be lost.

Click "Connect" in the "File" menu to connect to the device.

7.2 E2CAN Gateway Plugin



7.2.1 Ethernet Settings



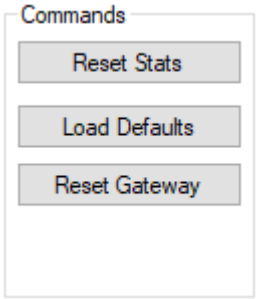
The Read button is used to update the current Ethernet configuration and the write button is used to configure and store the settings.

The IP Address is valid if it is in the range 1.0.0.0 to 254.255.255.255.

The Subnet Mask is valid if all set bits are on the left (MSB) and all cleared bits are on the right (LSB). Values 0.0.0.0 and 255.255.255.255 are also valid.

The Default Gateway must be a valid IP Address or 0.0.0.0.

7.2.2 Commands



7.2.3 Reset Stats

This will reset all communication interface and communication channel statistics, status counters and error counters.

7.2.4 Load Defaults

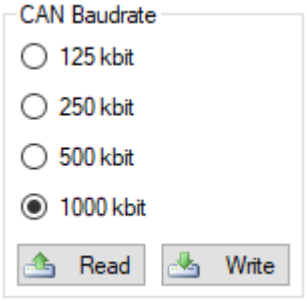
Load manufacturer defaults. Afterwards the connection will have to be re-established.

This will not affect the MAC Address, Serial Number and License Data.

7.2.5 Reset Gateway

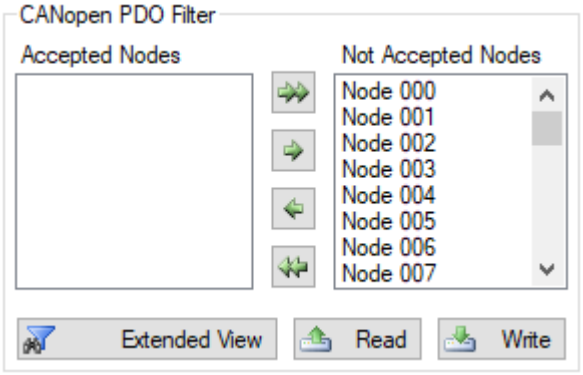
Software reset of the *Smart I/O* device. For an Ethernet connection, a new connection must then be established. Because it can take longer than 20 seconds for the software to detect that there is no longer a connection, it is advisable to manually perform a disconnect.

7.3 CAN Baudrate



Selection of the baudrate in kbit/s.

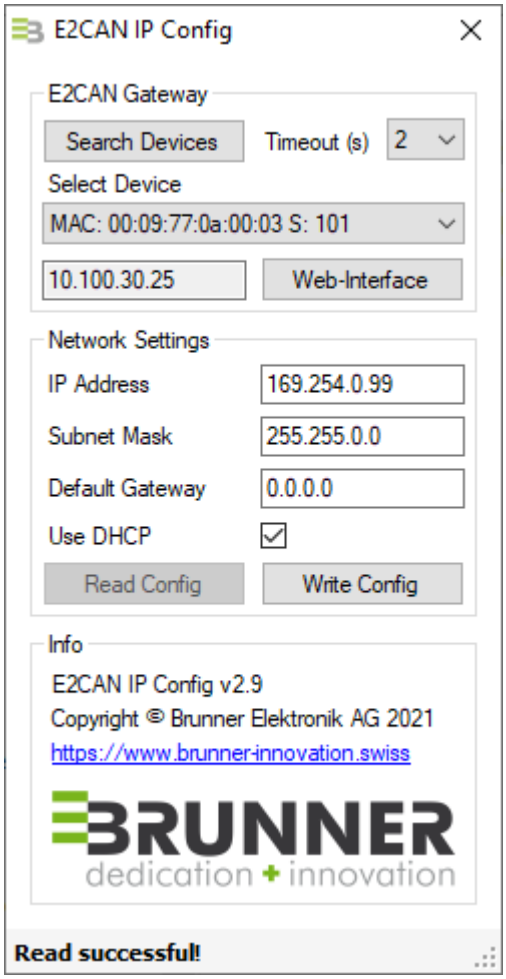
7.4 CANopen PDO Filter



The PDOs of the nodes listed in “Not Accepted Nodes” won’t be routed through the gateway. This is typically used when the CAN traffic is monitored, but certain process data isn’t of interest.

8. E2CAN IP Config Tool

This utility uses the Discovery Service and can be used for easy configuration of the *Smart I/O* IP settings.



Click the “Search Devices” button and select your device from the dropdown list.

Click “Web Interface” to open the *Smart I/O* web interface in your default web browser.

Click “Read Config” to read the current IP Settings.

Configure your settings and click the “Write Config” button.

The IP Address is valid if it is in the range 1.0.0.0 to 254.255.255.255.

The Subnet Mask is valid if all set bits are on the left (MSB) and all cleared bits are on the right (LSB). Values 0.0.0.0 and 255.255.255.255 are also valid.

The Default Gateway must be a valid IP Address or 0.0.0.0.

9. Web Interface

The *Smart I/O* provides a web interface that allows configuration and monitoring of the device using a web browser.

Supported web browsers:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge

Other Chromium based browsers should work as well.

We recommend Google Chrome 93 or newer.

JavaScript has to be enabled.

9.1 Setup

Make sure that your PC is configured to be on the same subnet as the *Smart I/O*.

Open your browser and type the IP address of your gateway device. You can obtain the IP address using *CANopen Commander*, *E2CAN IP Config* tool or your own Discovery service.

9.2 General Info Page

9.2.1 Device Info

General device info (hardware and firmware).

9.2.2 Device Status

Uptime since the last reset.

Gateway Operation Mode: Gateway or Sniffer

Gateway: Smart I/O Gateway is an active CAN node that can read and actively send CAN messages. In most cases this will be the operation mode of choice.

Sniffer: Smart I/O Gateway is a passive CAN node. It can only passively read CAN messages.

CAN-to-GW Access: if enabled the object dictionary of the *Smart I/O* device can be accessed over CAN. Otherwise it can only be accessed over Ethernet or USB.

9.2.3 Manufacturer Status

Developer and Reduced Modes should be disabled for production environments. If they are enabled, it's usually a sign of a hardware or firmware fault. Please contact Brunner support.

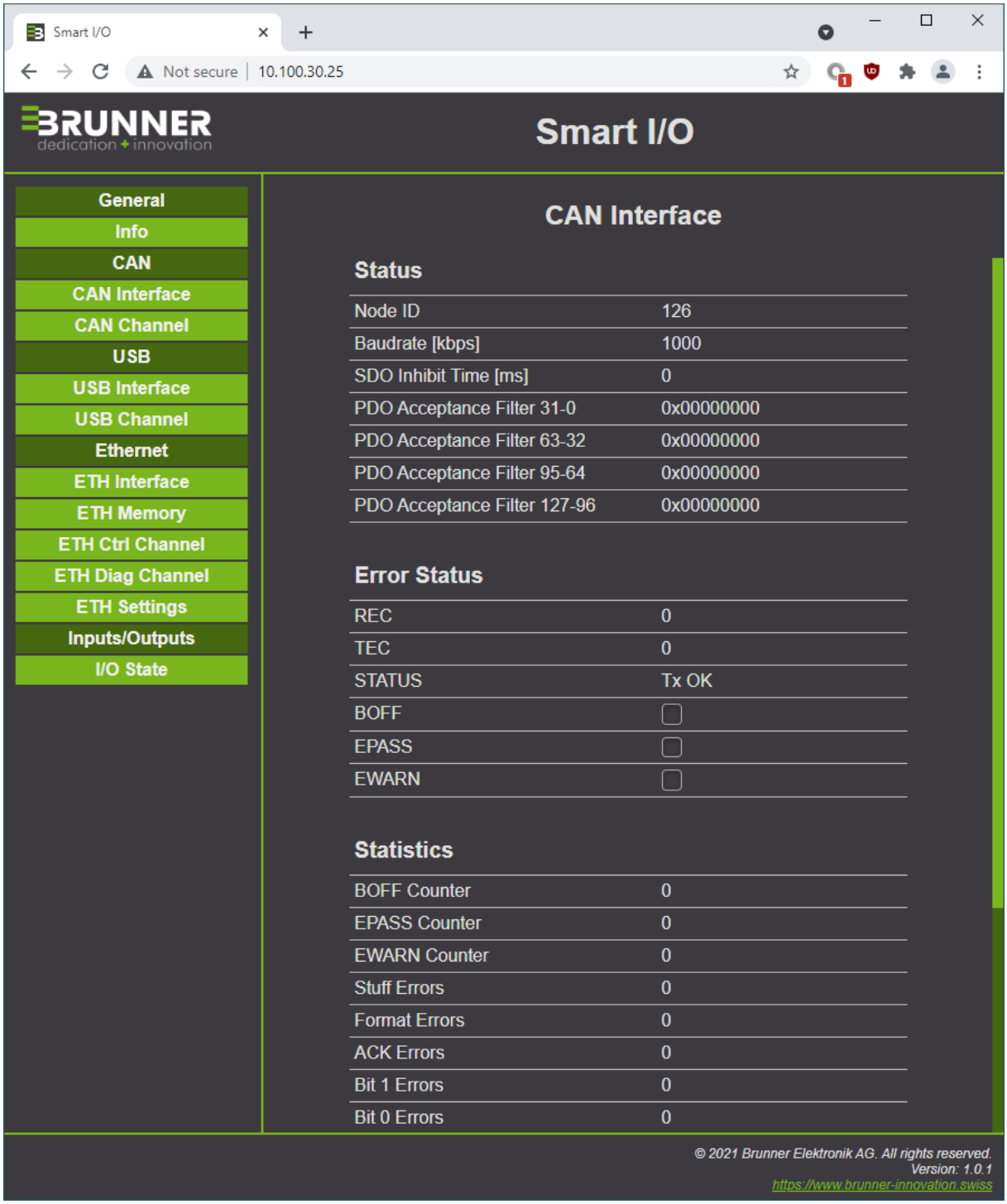
9.2.4 Commands

Reset Stats: This will reset all communication interface and communication channel statistics, status counters and error counters.

Load Defaults & Reset Device: Load factory defaults and reset the Smart I/O. This will not affect the MAC Address, Serial Number and License Data.

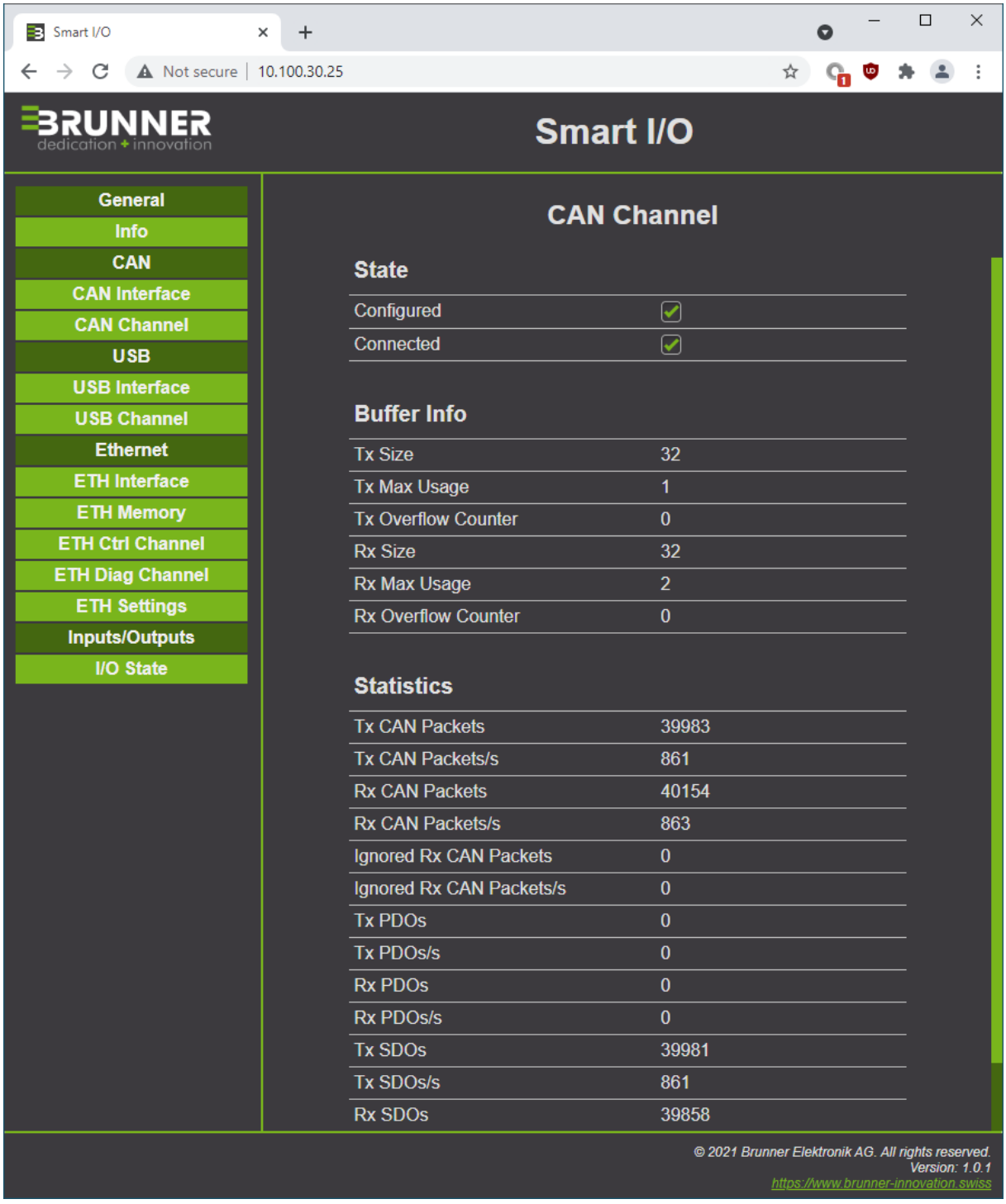
Reset Device: Software reset of the Smart I/O.

9.3 CAN Interface Page



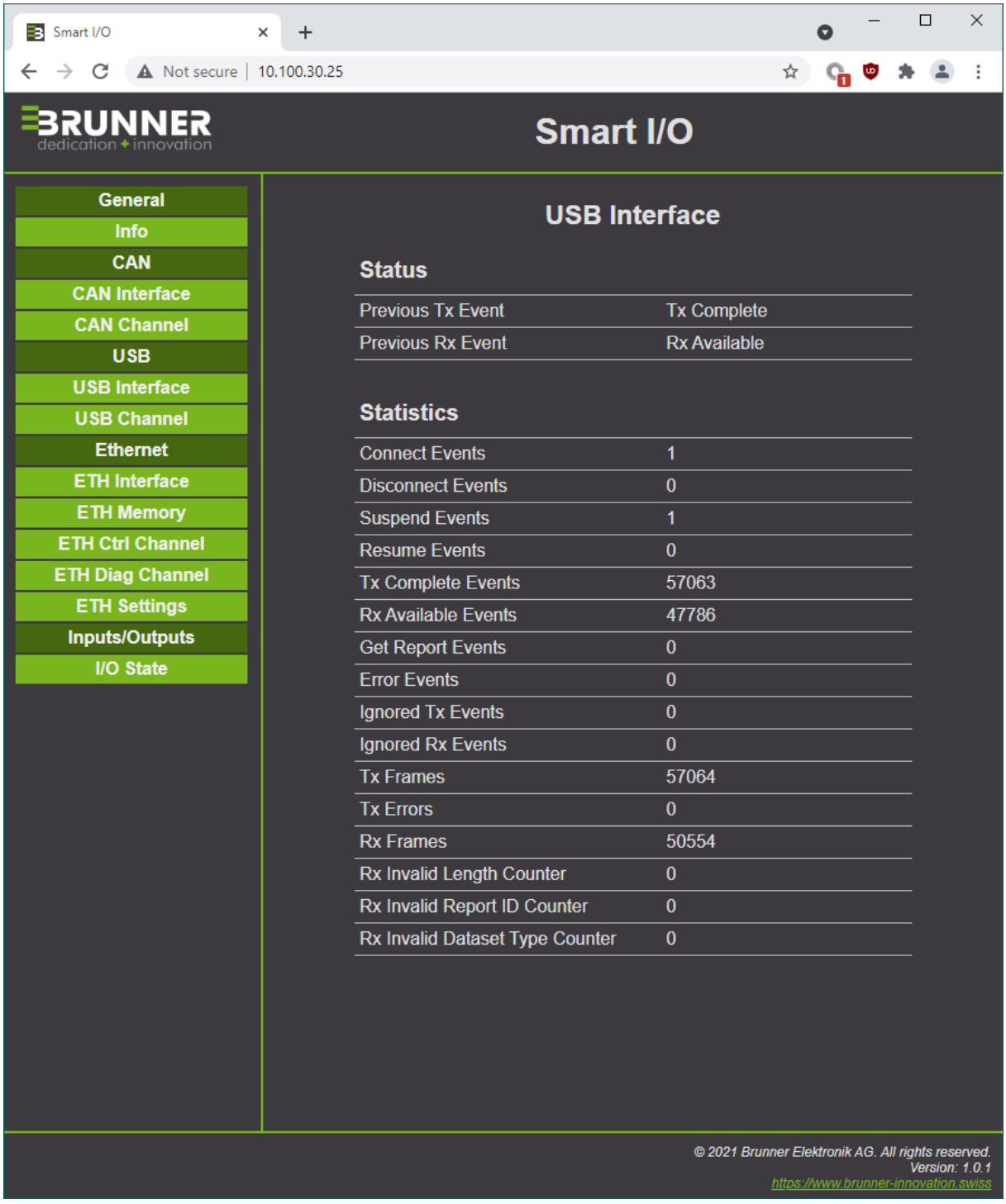
Please refer to the Object Dictionary, Object range 0x3200 ... 0x324F for further information.

9.4 CAN Communication Channel Page



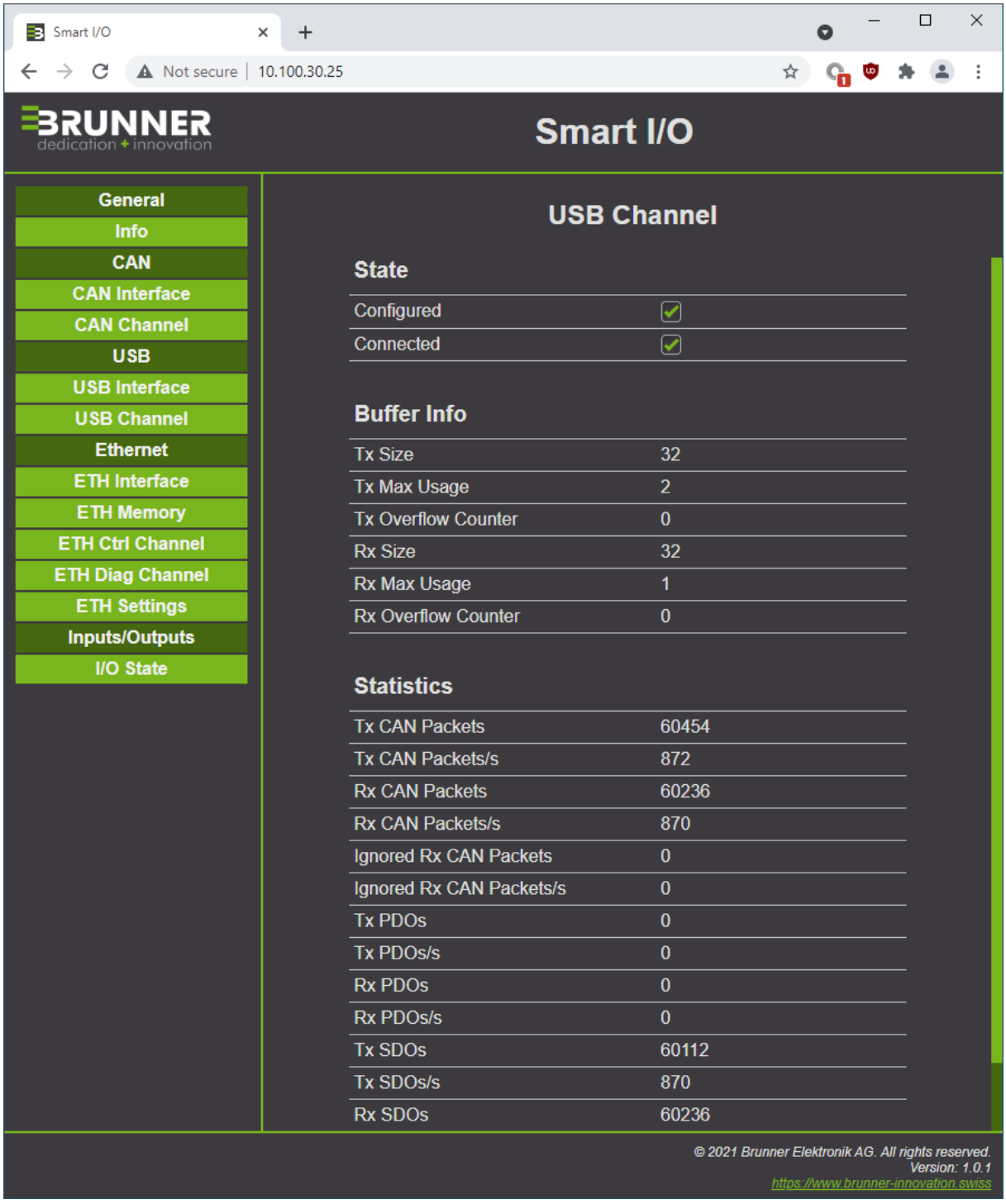
Please refer to the Object Dictionary, Object range 0x3250 ... 0x328F for further information.

9.5 USB Interface Page



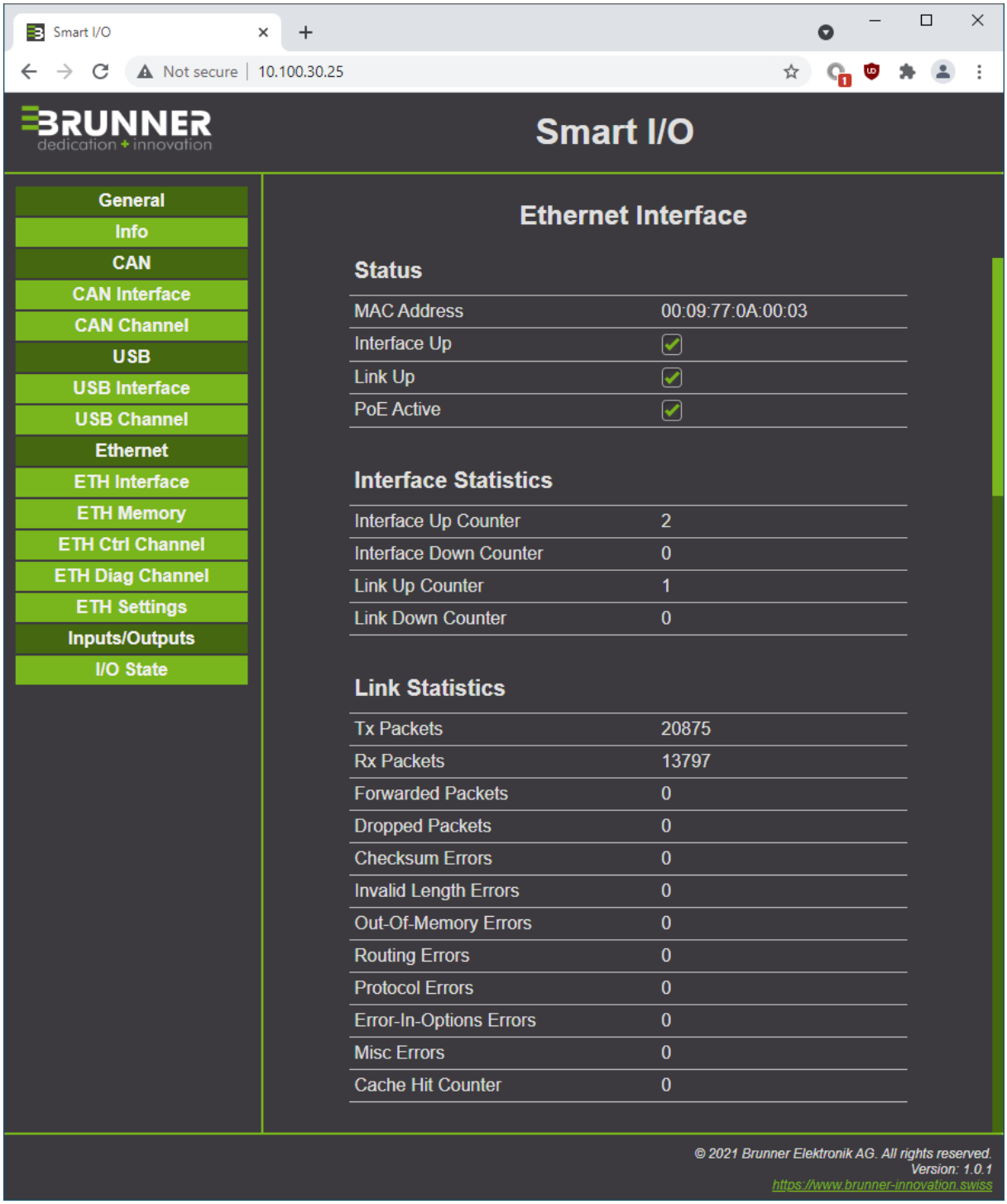
Please refer to the Object Dictionary, Object range 0x3300 ... 0x334F for further information.

9.6 USB Communication Channel Page



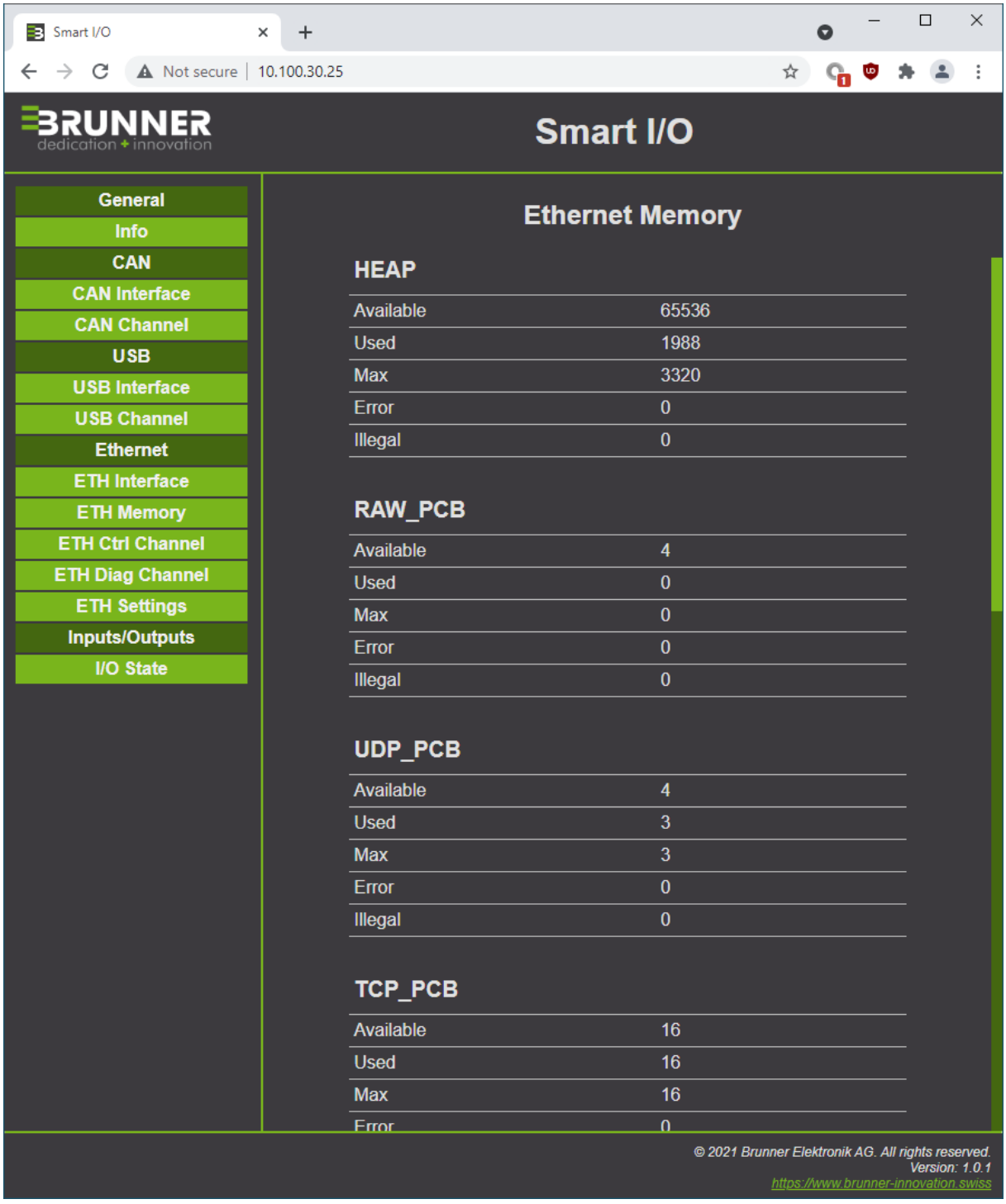
Please refer to the Object Dictionary, Object range 0x3350 ... 0x338F for further information.

9.7 Ethernet Interface Page



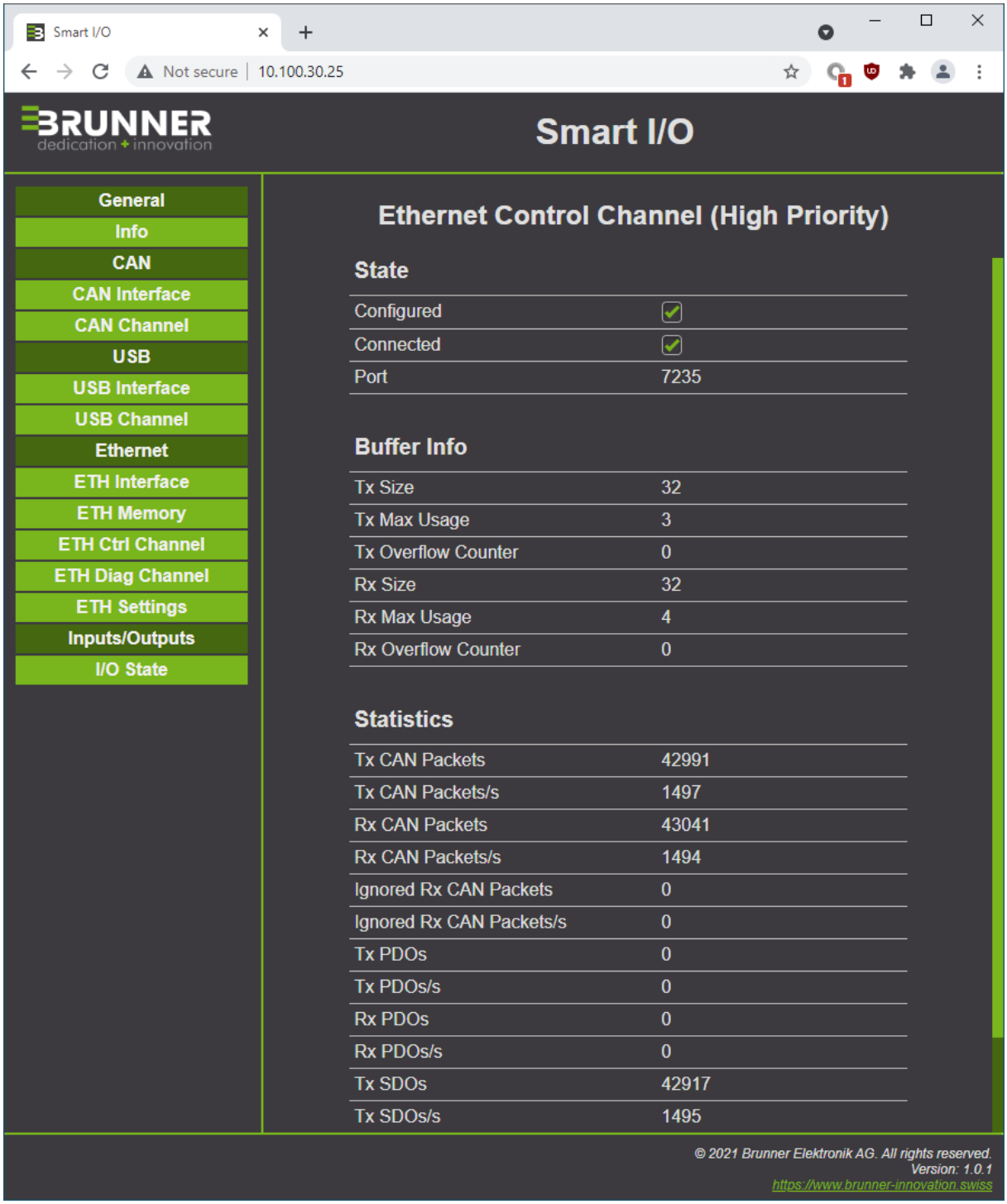
Please refer to the Object Dictionary, Object range 0x3400 ... 0x344F for further information.

9.8 Ethernet Memory Page



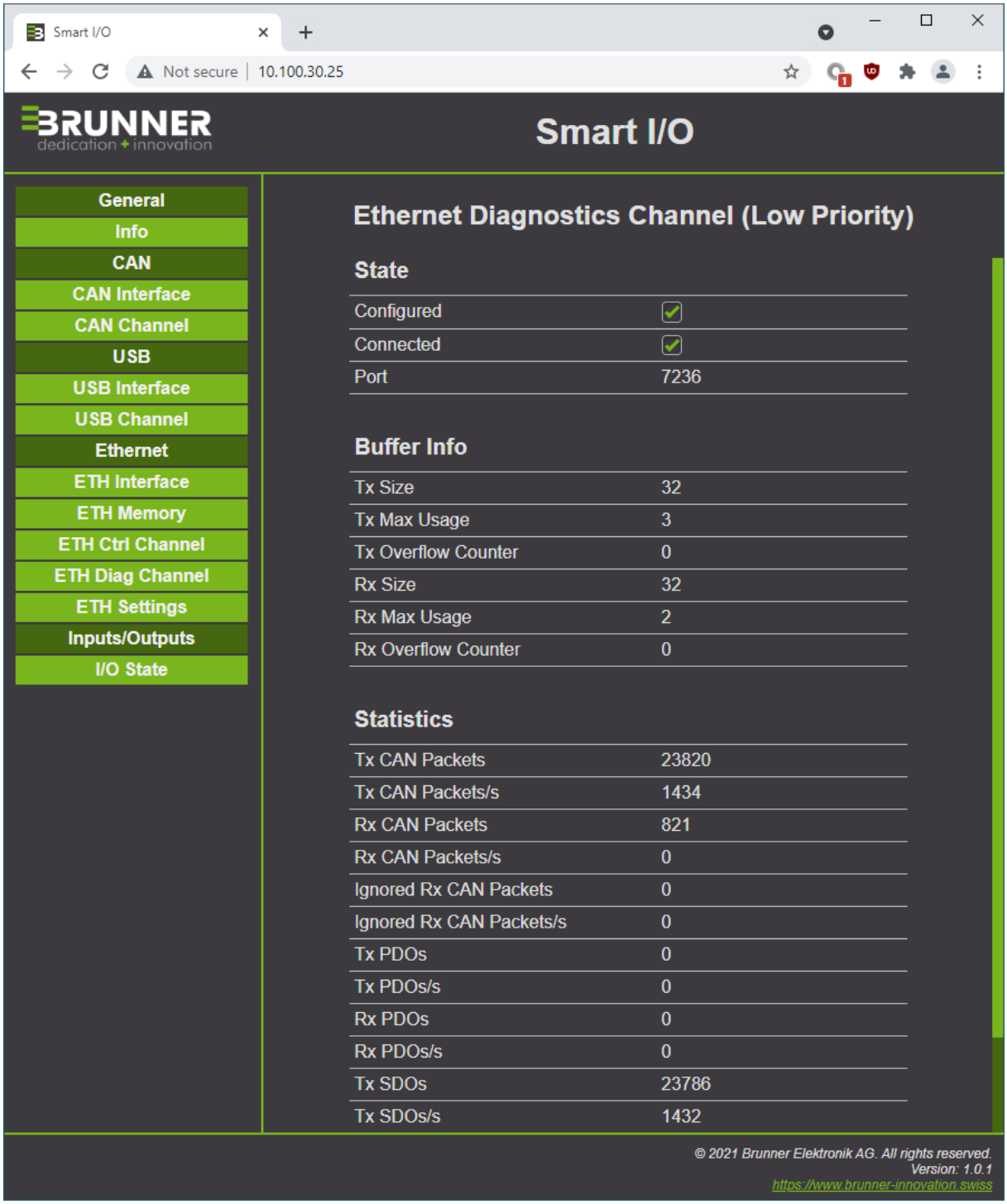
This page is mainly used by the manufacturer for analytical purposes.

9.9 Ethernet Control Communication Channel Page



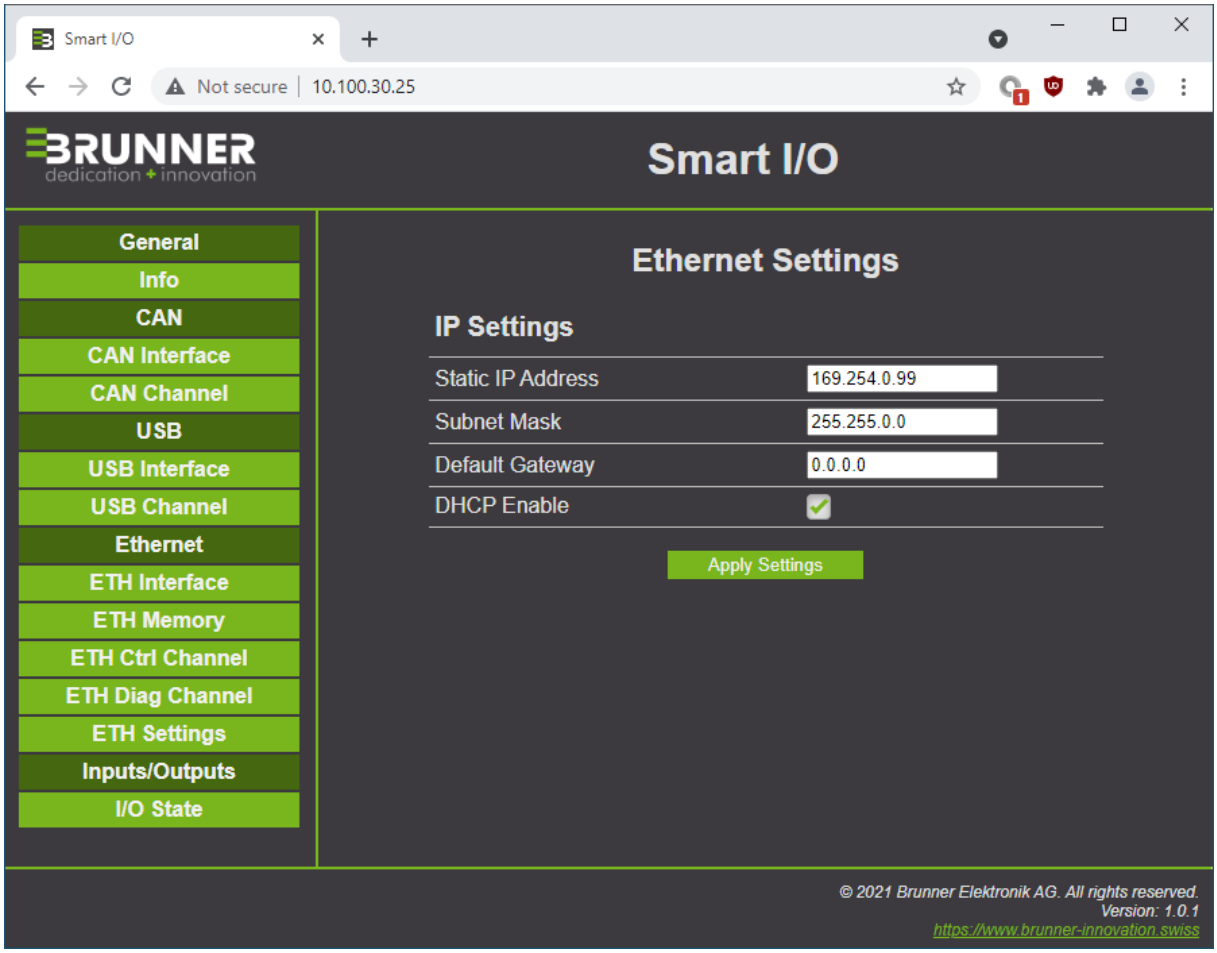
Please refer to the Object Dictionary, Object range 0x3450 ... 0x348F for further information.

9.10 Ethernet Diagnostics Communication Channel Page



Please refer to the Object Dictionary, Object range 0x3450 ... 0x34CF for further information.

9.11 Ethernet Settings Page



Configure your IP settings and click the “Apply Settings” button.

The IP Address is valid if it is in the range 1.0.0.0 to 254.255.255.255.

The Subnet Mask is valid if all set bits are on the left (MSB) and all cleared bits are on the right (LSB). Values 0.0.0.0 and 255.255.255.255 are also valid.

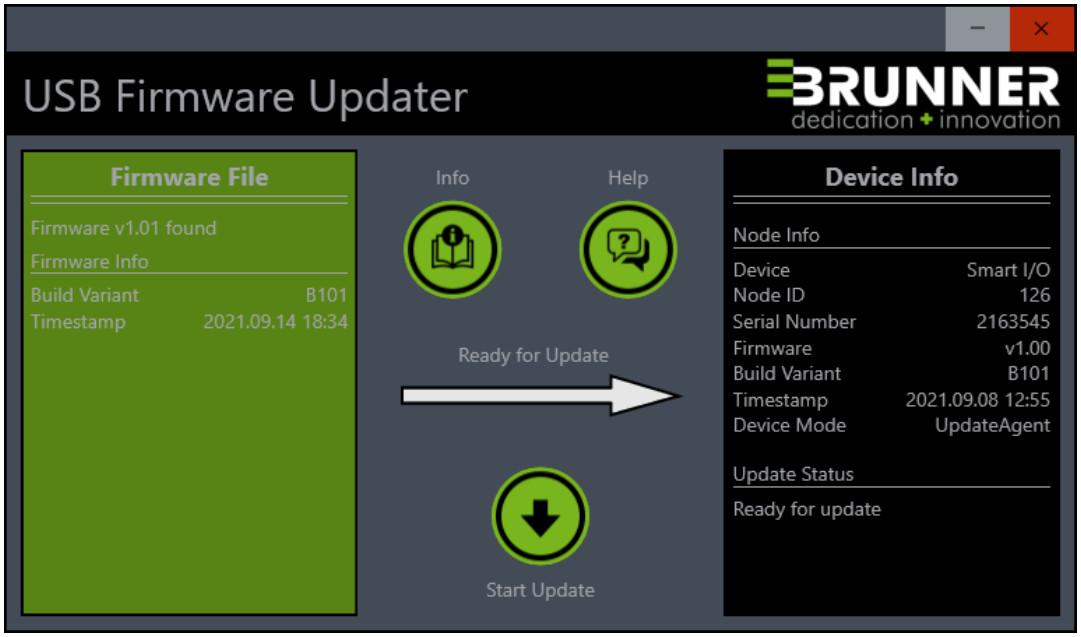
The Default Gateway must be a valid IP Address or 0.0.0.0.

9.12 I/O State Page

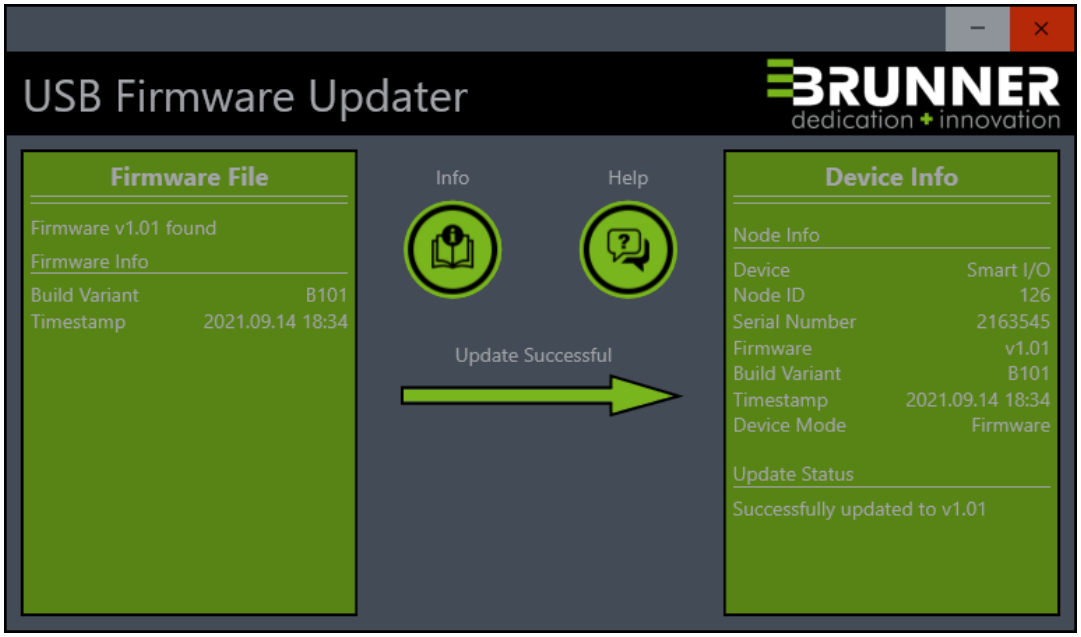
Please refer to the Object Dictionary, Object range 0x3800 ... 0x385F, 0x6000 ... 0x6FFF for further information.

10. Firmware Update

You'll need the *USB Firmware Updater* tool to update the firmware of your *Smart I/O* device.
Disconnect power and all other peripherals, then connect the *Smart I/O* device over USB.
Start up the *USB Firmware Updater* tool.



Click the "Start Update" button and wait for the update to complete.
Do not disrupt the update procedure!
After a successful update you'll see following screen.



You can close the *USB Firmware Updater*.

11. USB Driver Installation

If the *Smart I/O* is connected to a PC by USB, the operating system will recognize it as a USB HID device and load the standard HID driver without user interaction. No configuration steps are required.

12. Ordering Information

Order Number	**Name**	Specification
PRT.5245.020B.V001	Smart I/O	