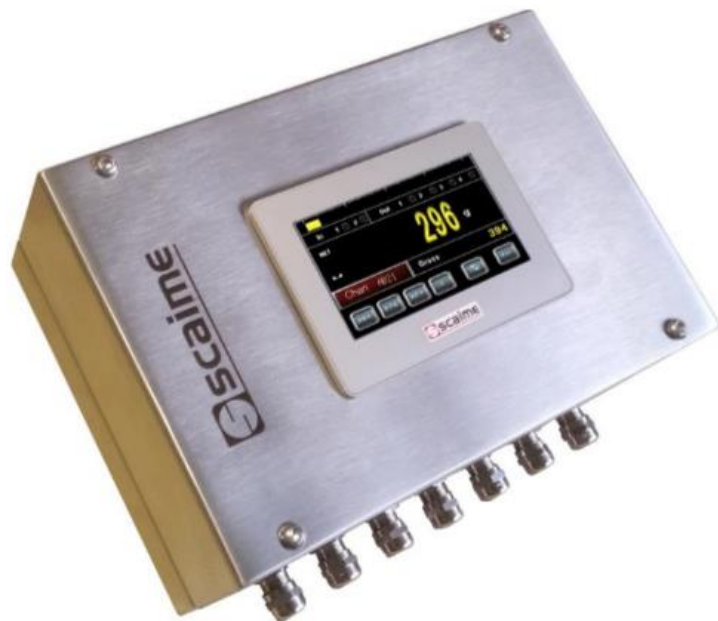




eNod4-T

Digital Process Transmitter



1 ENOD4 PRODUCT RANGE	7
1.1 General presentation.....	7
1.2 Versions	7
1.2.1 Communication protocol versions.....	7
1.2.2 IO+ version.....	7
1.3 eNodView Software	8
2 COMMUNICATION AND FUNCTIONING MODES	9
2.1 Communication protocols Modbus RTU and SCMBus	9
2.2 Functioning mode	9
2.3 HMI name	9
2.4 Simultaneous functioning of communications.....	10
2.4.1 Standard version	10
2.4.2 Profibus version	11
2.4.3 Ethernet versions	12
3 MODBUS RTU	13
3.1 Physical interfaces	13
3.2 Byte format.....	13
3.3 Modbus RTU supported functions	13
3.4 Frames structure	13
3.4.1 Function (03H/04H) – read N input registers (N = 30 max).....	13
3.4.2 Function (06H) – write single register	14
3.4.3 Function (10H) – preset multiple registers (N = 30 max)	14
3.4.4 Error frames.....	14
3.5 Address and Baud rate.....	14
3.6 Product identification.....	14
3.7 Measurement transmission	15
3.8 EEPROM error management.....	15
4 SCMBUS / FAST SCMBUS	16
4.1 Physical interfaces	16
4.2 SCMBus and fast SCMBus features	16
4.3 Byte format.....	16
4.4 Frames structure	17
4.4.1 Transmission organization	17
4.4.2 Reading request.....	17
4.4.3 Functional command request (tare, zero...)	17
4.4.4 Error frame	17
4.5 Address and Baud rate.....	18
4.6 Product identification.....	18
4.7 Measurement transmission	18
4.8 Continuous transmission.....	18
4.9 EEPROM error management.....	18
5 CANOPEN	19
5.1 Physical interface.....	19
5.2 LED CANopen	19
5.3 Frame format	19
5.4 Messages transfers hierarchy	20

5.5 eNod4 status remote management.....	21
5.5.1 NMT commands	22
5.5.2 Synchronization messages	23
5.5.3 Emergency messages	23
5.6 Error control services.....	24
5.6.1 Heartbeat and boot-up	24
5.6.2 Node guarding	25
5.7 Access to the object dictionary	26
5.7.1 SDO communication	26
5.7.2 PDO communications.....	28
5.8 CANopen command and response registers	30
5.9 Communication objects.....	30
5.9.1 0x1001 / 0x00 : error register.....	30
5.9.2 0x1003 : Pre-defined error field	30
5.9.3 0x1005 / 0x00 : synchronization messages COB-ID	31
5.9.4 0x100C / 0x00 : guard time.....	31
5.9.5 0x100D / 0x00 : life time factor	31
5.9.6 0x1010 : Store parameters	31
5.9.7 0x1014 / 0x00 : Emergency COB-ID.....	32
5.9.8 0x1016 : Heartbeat consumer time	32
5.9.9 0x1017 / 0x00 : Heartbeat producer time	32
5.9.10 0x4800 : Safety mode	32
5.9.11 Error behavior	33
5.10 PDO-related communication objects.....	33
5.10.1 RPDO default mapping.....	33
•0x1600 : RPDO1 mapping parameters	33
•0x1601 : RPDO2 mapping parameters	34
•0x1602 : RPDO3 mapping parameters	34
•0x1603 : RPDO4 mapping parameters	34
•0x1604 : RPDO5 mapping parameters	34
5.11 Product identification.....	35
•0x1009 : Manufacturer hardware version	35
•0x100A : Manufacturer software version	35
•0x1018 : Identity object	35
5.12 Measurement transmission	35
5.13 EEPROM error management.....	35
6 CANOPEN TPDO MAPPING	36
6.1 Default TPDOs Mapping.....	36
•0x1A00 : TPDO1 mapping.....	36
•0x1A01 : TPDO2 mapping.....	36
•0x1A02 : TPDO3 mapping.....	36
7 PROFIBUS DPV1	37
7.1 Physical interface.....	37
7.2 GSD file	37
7.3 Cyclic exchanges.....	37
7.3.1 Cyclic inputs modules	37
7.3.2 Cyclic inputs/outputs modules.....	37

7.4 Acyclic exchanges.....	38
7.5 eNod4 Profibus DP features	38
7.5.1 Sync.....	38
7.5.2 Freeze.....	38
7.5.3 Fail-safe.....	38
7.5.4 Profibus DP standard and extended diagnoses	39
7.6 Product identification.....	40
7.7 Measurement transmission	40
7.8 EEPROM error management.....	40
8 PROFIBUS MODULES LIST	41
9 MEASUREMENT AND STATUS	42
9.1 Measurement transmission	42
9.1.1 Gross measurement	42
9.1.2 Net measurement	42
9.1.3 Tare value	42
9.1.4 Factory calibrated points.....	42
9.1.5 Preset Tare value.....	42
9.1.6 Measurement status	43
9.2 Weighing diagnosis.....	44
9.2.1 Global weighing diagnosis	44
9.2.2 Sensor input control	45
10 PROCESSING FUNCTIONAL COMMANDS.....	46
10.1 Principles.....	46
10.2 Functional commands list	47
10.3 Functional commands description	48
10.3.1 Switch legal sealing	48
10.3.2 Clear DSD memory	48
10.3.3 Weighing result acquisition.....	48
10.3.4 DSD read	48
10.3.5 Backward DSD read.....	49
10.3.6 Reset.....	49
10.3.7 EEPROM storage	49
10.3.8 Restore default settings	49
10.3.9 Zero.....	49
10.3.10 Tare.....	49
10.3.11 Cancel tare	50
10.3.12 Cancel last command	50
10.3.13 Theoretical scaling.....	50
10.3.14 Zero adjustment	50
10.3.15 Start physical calibration	50
10.3.16 Calibration zero acquisition.....	50
10.3.17 Segment 1 acquisition	50
10.3.18 Segment 2/3 acquisition	50
10.3.19 Store calibration (end of physical calibration).....	50
10.3.20 Logical outputs 1-4 activation/deactivation	51
10.3.21 Zero offset	51
10.3.22 Preset tare	51

10.3.23	Sensor input reference	51
10.3.24	Sensor input control	51
11	CALIBRATION SETTINGS AND PROCEDURES	52
11.1	Principles	52
11.2	Calibration methods	52
11.3	Settings description	53
11.3.1	Maximum capacity	53
11.3.2	Number of calibration segments.....	53
11.3.3	Calibration loads 1/2/3	53
11.3.4	Sensor sensitivity	53
11.3.5	Scale interval.....	53
11.3.6	Zero calibration	54
11.3.7	Span coefficients 1/2/3	54
11.3.8	Span adjusting coefficient	54
11.3.9	Calibration place g value / place of use g value.....	54
11.3.10	Zero offset	54
12	FILTERS	55
12.1	Principles	55
12.2	Settings list.....	55
12.3	Settings description	55
12.3.1	A/D conversion rate	55
12.3.2	Filters activation & order.....	56
12.3.3	Low-pass filter cut-off frequency.....	56
12.3.4	Band-stop filter high cut-off frequency.....	57
12.3.5	Band-stop filter low cut-off frequency	57
12.4	Limitations	57
13	CONFIGURATION OF INPUT/OUTPUT	58
13.1	Principles	59
13.1.1	Logical inputs	59
13.1.2	Analog output (IO+ version)	59
13.1.3	Logical outputs	60
13.2	Settings description	61
13.2.1	Logical inputs assignment.....	61
13.2.2	Holding time (debounced time)	62
13.2.3	Analog output(s) assignment (IO+ version)	63
13.2.4	External value to control analog output (IO+ version).....	63
13.2.5	Logical outputs 1&2 assignment.....	64
13.2.6	Logical outputs 3&4 assignment.....	65
13.2.7	Set points functioning	65
13.2.8	Set points high and low values	66
13.3	Input/output level.....	67
14	LEGAL FOR TRADE OPTIONS	68
14.1	Principles	69
14.2	Settings description	69
14.2.1	Legal for trade switch	69
14.2.2	Legal for trade sealing.....	70
14.2.3	Legal for trade software version	71

- 14.2.4 Legal for trade counter.....71
- 14.2.5 Legal for trade checksum.....71
- 14.2.6 Alibi memory or DSD (Data Storage Device)71
- 14.2.7 Zero functions71
- 14.2.8 Stability criterion.....72
- 14.2.9 Decimal point position72
- 14.2.10 Unit.....73
- 14.2.11 Options.....73
- 15 MODBUS RTU REGISTERS TABLE74
- 16 CRC-16 CALCULATION ALGORITHM77

1 ENOD4 PRODUCT RANGE

1.1 General presentation

eNod4 is a high-speed digital process transmitter with programmable functions and powerful signal processing capabilities. **eNod4** offers operating modes for advanced process control both static and dynamic.

Quick and accurate:

- Analog to digital conversion rate up to 1920 meas/s with maximum scaled resolution of $\pm 500\,000$ points.
- Digital filtering and measurement scaling.
- Measurement transmission up to 1 000 meas/s.

Easy to integrate into automated system:

- **USB, RS485** and **CAN** communication interfaces supporting **ModBus RTU**, **CANopen®** and **PROFIBUS-DPV1** (depending on version) communication protocols.
- Digital Inputs/Outputs for process control.
- Setting of node number by rotary switches and communication baud rate by dip switches.
- Integrated selectable network termination resistors.
- Wiring by plug-in terminal blocs.

1.2 Versions

1.2.1 Communication protocol versions

- Strain gauges load-cell conditioner with **CANopen®** and **ModBus RTU** communication.
- Strain gauges load-cell conditioner with **Profibus DP-V1** and **ModBus RTU** communication.
- Strain gauges load-cell conditioner with **Modbus TCP** and **ModBus RTU** communication.
- Strain gauges load-cell conditioner with **EtherNet/IP** and **ModBus RTU** communication.
- Strain gauges load-cell conditioner with **Profinet IO** and **ModBus RTU** communication.
- Strain gauges load-cell conditioner with **EtherCAT** and **ModBus RTU** communication.

EDS, GSD, ESI and GSDML configuration files for above protocols can be downloaded from our web site: <http://www.scaime.com>

1.2.2 IO+ version

In conjunction with all communication protocol versions, **eNod4** can supports an opto-insulated board fitted with:

- 2 additional digital inputs and 1 speed sensor dedicated input.
- 0-5V or 0-10V analog output voltage.
- 4-20mA, 0-24mA, 0-20mA or 4-20mA with alarm at 3.6mA analog output current.

1.3 eNodView Software

So as to configure **eNod4**, SCAIME provides eNodView software tool. **eNodView** is the software dedicated to eNod devices and digital load cell configuration from a PC. This simple graphical interface allows accessing the whole functionalities of **eNod4** for a complete setting according to the application.

eNodView features and functions:

- eNod4 control from a PC
- Calibration system
- Modification/record of all parameters
- Measure acquisition with graphical display
- Numerical filters simulation
- Frequential analysis FFT
- Process control
- Network parameter

eNodView software is available in English and French version and can be downloaded from our web site: <http://www.scaime.com> or ordered to our sales department on a CD-ROM support.

2 COMMUNICATION AND FUNCTIONING MODES

Name	Modbus address	Acyclic DPV1 slot/index	Profibus cyclic IN/OUT	CANopen® index/sub-index	Type	Access
Functioning mode / Serial protocol	0x003E	/	/	0x2000 / 0x00	Uint	RW
HMI name	0x0034	0x0B / 0x00		0x3701 / 0x00	String	RW

2.1 Communication protocols Modbus RTU and SCMBus

Modbus RTU, SCMBus, and fast SCMBus communication protocols are accessible through AUX, USB. Modbus RTU or Profibus only depending on version on DB9 connection.

The protocol can be changed via the « Functioning mode/ serial protocol » register (see below).

bits b9b8	Protocol
00	SCMBus
01	Modbus RTU
11	Fast SCMBus

Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).

2.2 Functioning mode

The « Functioning mode/ serial protocol » register offers the possibility to change the eNod4 application according to the following list:

bits b1b0	Functioning mode				
	eNod4-T	eNod4-C	eNod4-D	eNod4-F	eNod4-B
00	Transmitter	Transmitter	Transmitter	Transmitter	Transmitter
01	/	/	Dosing by filling	Dosing	Belt scale
10	/	Checkweigher transmitter on request	Dosing by unfilling	/	Belt weigh feeder

Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).

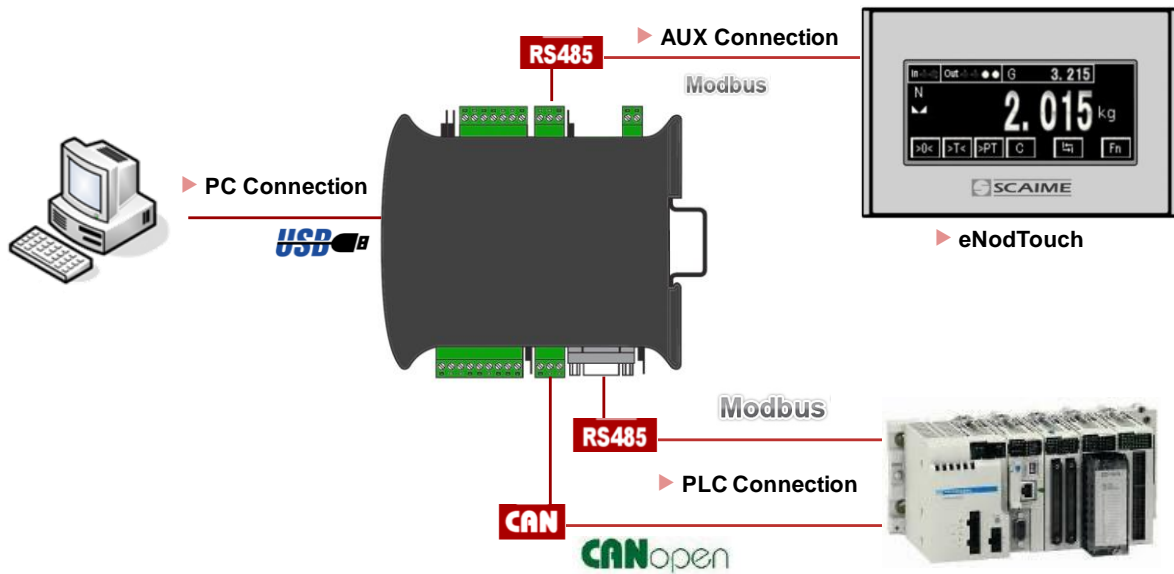
2.3 HMI name

The “HMI name” is a string of 4 characters freely usable to identify the node on any HMI connected to eNod.

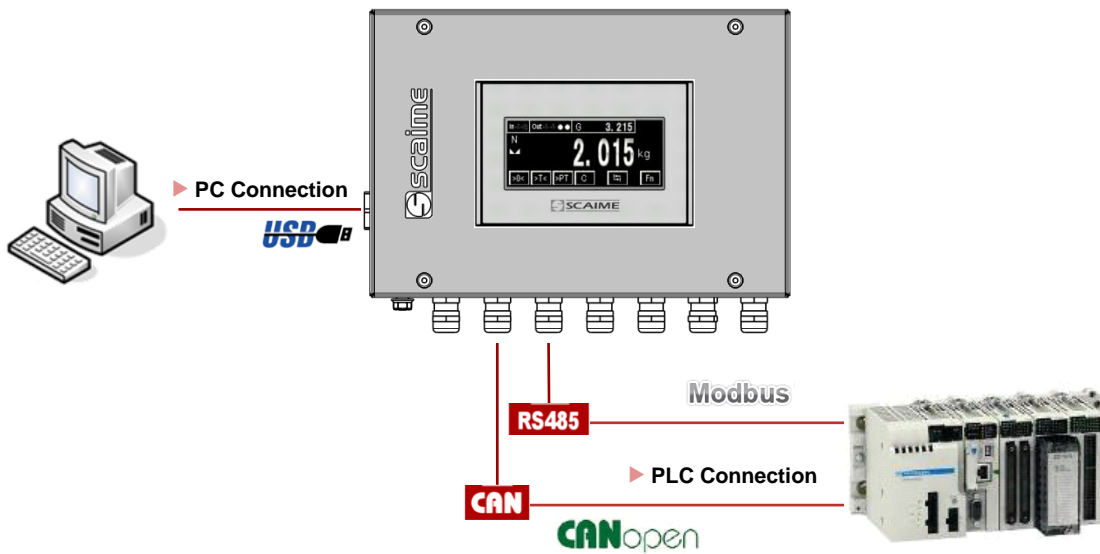
2.4 Simultaneous functioning of communications

2.4.1 Standard version

- DIN Version



- BOX Version

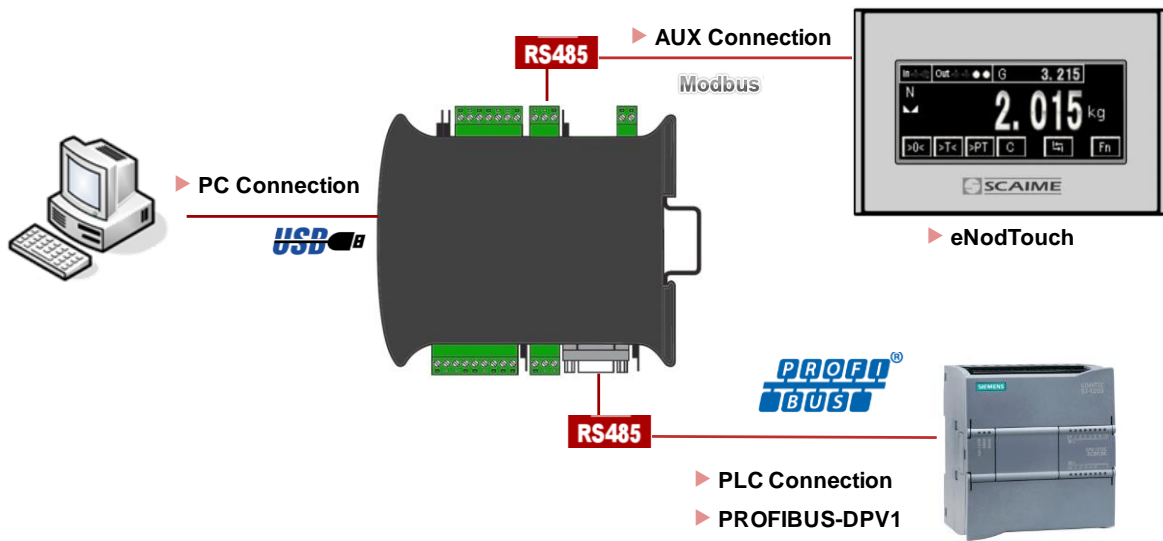


Simultaneous Communication	RS485 PLC	RS485 AUX	CAN
USB	Yes*	No	Yes*
RS485 PLC		Yes	No
RS485 AUX			Yes*

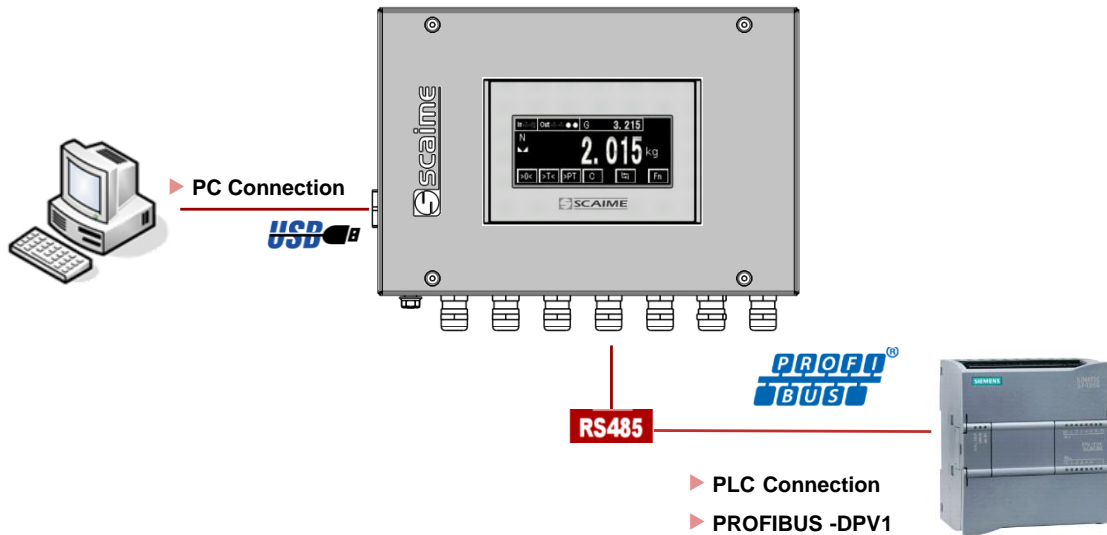
(*)Simultaneous use of CAN or RS485 PLC communication with USB or RS485 AUX can reduce performance of this interface.

2.4.2 Profibus version

- DIN Version



- BOX Version

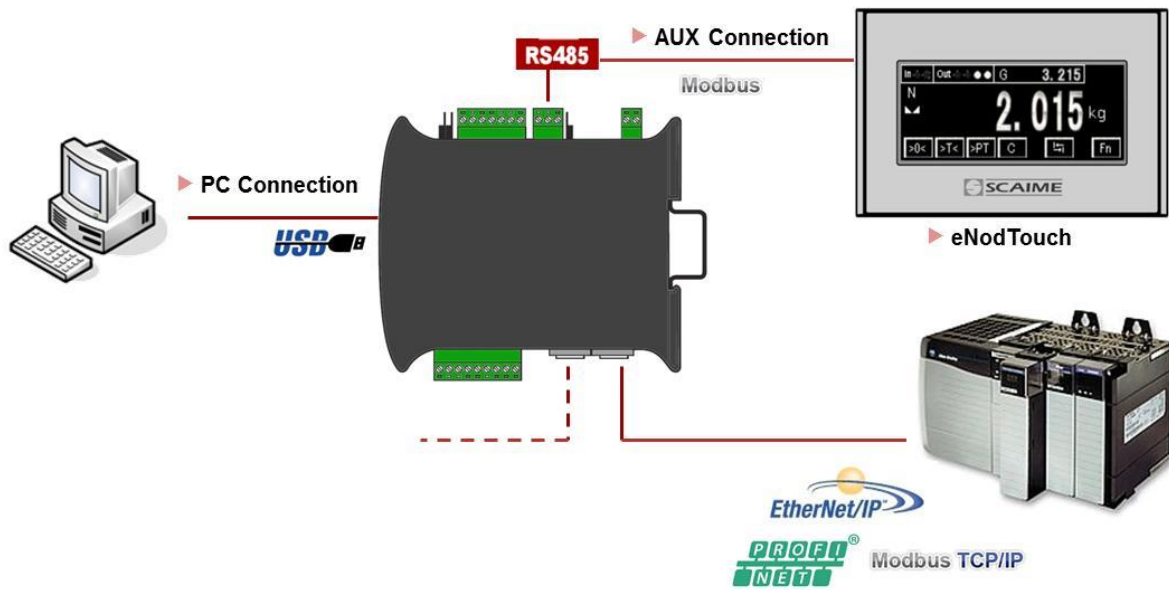


Simultaneous Communication	Profibus	RS485 AUX
USB	Yes*	No
Profibus		Yes*

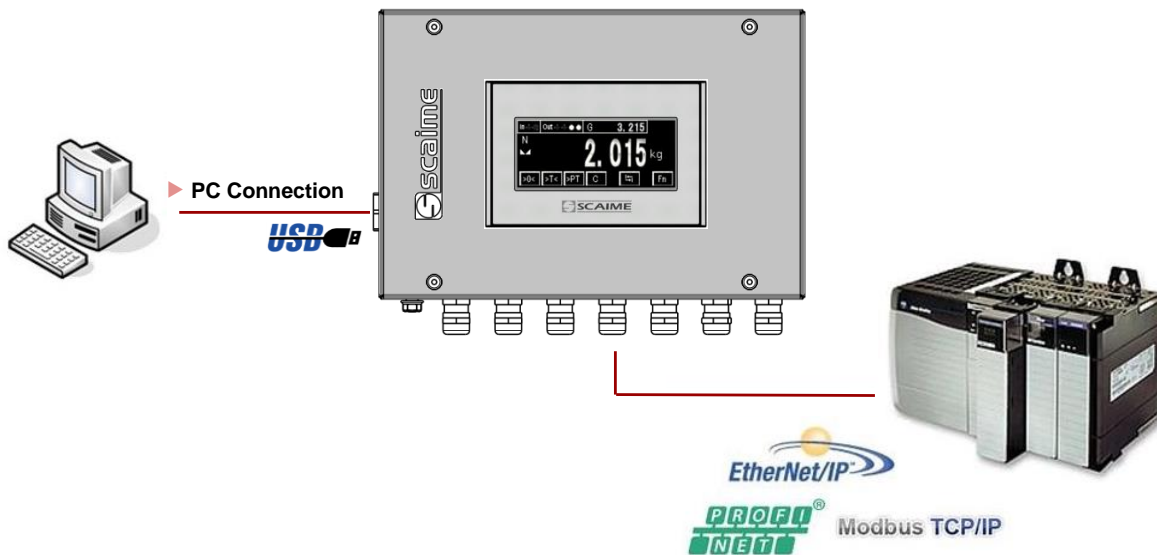
(*) Simultaneous use of Profibus with USB or RS485AUX can reduce performance of this interface.

2.4.3 Ethernet versions

- DIN Version



- BOX Version



Simultaneous Communication	Ethernet	RS485 AUX
USB	Yes*	No
Ethernet		Yes*

(*)Simultaneous use of Ethernet with USB or RS485 AUX can reduce performance of this interface.

3 MODBUS RTU

3.1 Physical interfaces

Modbus RTU communication protocol can be used either through **eNod4** USB port, AUX port. Modbus RTU or Profibus only depending on version on DB9 connection.

USB port behaves as a full duplex interface whereas the DB9 and AUX ports support half-duplex RS485 communication. Supported baud rates are 9600, 19200, 38400, 57600, and 115200.

For a complete description of the recommendations about **eNod4** RS485 connection, please refer to the user manual “characteristics and functioning” of the **eNod4**.

Note: using **eNod4** through USB requires installing first the necessary USB drivers available on the website <http://www.scaime.com>.

3.2 Byte format

Data transmitted to **eNod4** thanks to Modbus RTU communication protocol must respect following format:

- 1 start bit
- 8 data bits
- no parity
- 2 stop bits

Every Modbus RTU frame is ended by a CRC-16 2-bytes code whose polynomial generator is

$$G(x) = x^{16} + x^{15} + x^2 + 1$$

(cf. CRC-16 calculation algorithm).

3.3 Modbus RTU supported functions

As a Modbus RTU slave, **eNod4** supports following Modbus RTU functions:

Function	Code
<i>read N registers*</i>	03 _H / 04 _H
<i>write 1 register*</i>	06 _H
<i>write N registers*</i>	10 _H

* 1 register = 2 bytes, maximum admitted value for N is 30.

Note: Broadcast addressing is not allowed by **eNod4**.

3.4 Frames structure

During a read or write transaction, the two bytes of a register are transmitted MSB first then LSB.

If a data is coded on **4 bytes** (that means it requires two registers), **the two LSB are stored in the low address register and the two MSB are stored in the high address register.**

3.4.1 Function (03H/04H) – read N input registers (N = 30 max)

- request command sent to the slave :

slave address	03 _H or 04 _H	starting register offset	N registers	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

- slave response :

<i>slave address</i>	<i>03_H or 04_H</i>	<i>NB *</i>	<i>data 1</i>	<i>...</i>	<i>CRC16</i>
1 byte	1 byte	1 byte	2 bytes	2 bytes	2 bytes

* NB: number of read bytes (= N*2)

3.4.2 Function (06H) – write single register

- request command sent to the slave :

<i>slave address</i>	<i>06_H</i>	<i>register offset</i>	<i>data</i>	<i>CRC16</i>
1 byte	1 byte	2 bytes	2 bytes	2 bytes

- slave response :

<i>slave address</i>	<i>06_H</i>	<i>register offset</i>	<i>data</i>	<i>CRC16</i>
1 byte	1 byte	2 bytes	2 bytes	2 bytes

3.4.3 Function (10H) – preset multiple registers (N = 30 max)

- request command sent to the slave :

<i>slave address</i>	<i>10_H</i>	<i>starting register offset</i>	<i>N registers</i>	<i>NB</i>	<i>Data 1</i>	<i>...</i>	<i>CRC16</i>
1 byte	1 byte	2 bytes	2 bytes	1 byte	2 bytes	2 bytes	2 bytes

- slave response :

<i>slave address</i>	<i>10_H</i>	<i>starting register offset</i>	<i>N registers</i>	<i>CRC16</i>
1 byte	1 byte	2 bytes	2 bytes	2 bytes

3.4.4 Error frames

- frame format in case of a transaction error :

<i>slave address</i>	<i>Function code + 80_H</i>	<i>error code</i>	<i>CRC16</i>
1 byte	1 byte	1 byte	2 bytes

- Error codes meaning :

<i>Error code</i>	<i>Meaning</i>	<i>description</i>
01_H	<i>illegal function</i>	<i>Modbus-RTU function not supported by eNod4</i>
02_H	<i>illegal data address</i>	<i>register address requested out of eNod4 register table</i>
03_H	<i>illegal data value</i>	<i>forbidden data values for the requested register</i>
04_H	<i>eNod4 not ready</i>	<i>eNod4 is not ready to answer (for example measurement request during a taring operation)</i>

3.5 Address and Baud rate

<i>Address Modbus RTU</i>	<i>Meaning</i>	<i>Access</i>	<i>Type</i>
0x0001	<i>Address and Baud rate</i>	RO	Uint

Reads the address and baud rate selected on the front panel via the rotary switches and dipperswitches.

3.6 Product identification

Software and product versions of the **eNod4** are accessible via Modbus RTU.

<i>Address Modbus RTU</i>	<i>Meaning</i>	<i>Access</i>	<i>Type</i>
<i>0x0000</i>	<i>SW and product version</i>	<i>RO</i>	<i>Uint</i>

The 12 LSB bits define the software version (073_H = 115) and the 4 MSB bits define the product version (6_H for the **eNod4**).

3.7 Measurement transmission

As a master/slave protocol, measurement transmission in Modbus protocol is only done on master request.

3.8 EEPROM error management

Functioning and calibration parameters are stored in EEPROM. After every reset the entireness of parameters stored in EEPROM is checked. If a default appears, measurements are set to 0xFFFF and default is pointed out in measurement status.

4 SCMBUS / FAST SCMBUS

4.1 Physical interfaces

SCMBus and fast SCMBus communication protocols can be used either through **eNod4** USB port and AUX port.

USB port behaves as a full duplex interface whereas the DB9 and AUX ports support half-duplex RS485 communication. Supported baud rates are 9600, 19200, 38400, 57600, and 115200.

For a complete description of the recommendations about **eNod4** RS485 connexion, please refer to the user manual “characteristics and functioning” of the **eNod4**.

Note : using **eNod4** through USB requires installing first the necessary USB drivers available on the website <http://www.scaime.com>.

4.2 SCMBus and fast SCMBus features

SCMBus and its variant fast SCMBus can be imbricate into ModBus RTU protocol if the setting ‘communication protocol’ is set to SCMBus or fast SCMBus. That means that **eNod4** continues answering Modbus RTU frames but it also allows the device to send frames coded according to SCMBus/fast SCMBus format.

Each protocol has its advantages:

- in SCMBus measurements are transmitted as ASCII with the decimal point and the unit integrated to the frame
- fast SCMBus is dedicated to fast measurement transmission as the frames are the most compact as possible
- both protocols allow to communicate without any master request (continuous transmission or sampling triggered by a logical input)

4.3 Byte format

Data transmitted to **eNod4** thanks to SCMBus or fast SCMBus communication protocol must respect following format:

- 1 start bit
- 8 data bits
- no parity
- 2 stop bits

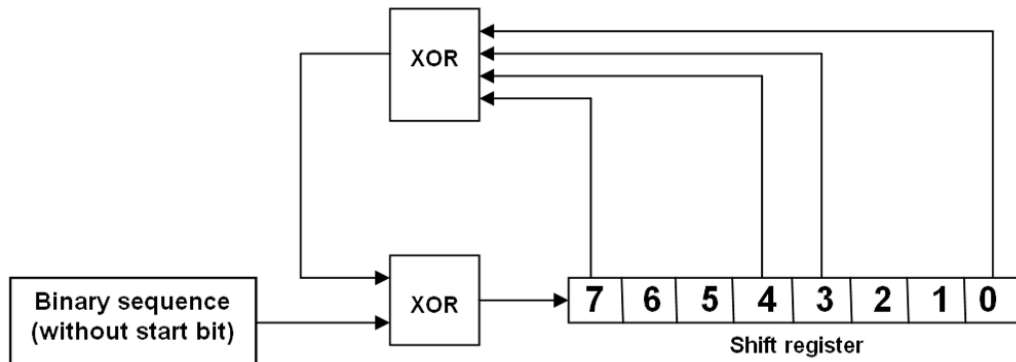
in SCMBus protocol, data is encoded as ASCII numeral characters (30_H 39_H) and ASCII hexadecimal characters (3A_H 3F_H).

in fast SCMBus protocol, data is encoded as signed hexadecimal (see frame structure paragraph) below.

SCMBus CRC-8 byte is generated by the following polynomial:

$$G(x) = x^8 + x^7 + x^4 + x^3 + 1$$

The CRC-8 polynomial result can be determined by programming the algorithm corresponding to the following diagram:



Note: The frame error detection can be ignored. Value **0xFF** of the CRC-8 always is admitted by **eNod4** and a received frame which is ended by such CRC-8 is considered as a frame without any error.

- Fast SCMBus checksum byte is obtained by summing all the frame previous bytes then setting b7 bit to 1.

4.4 Frames structure

4.4.1 Transmission organization

- frame : **eNod4** address first
- byte : lsb first
- multi-bytes data : MSB first

4.4.2 Reading request

- request

Address	Command	CR	CRC
1 Hex byte	1 Hex byte (command)	1 ASCII byte (0D _H)	1 Hex byte

- SCMBus response

Address	Status	Value	CR	CRC
1 Hex byte	2 Hex bytes	N ASCII Hex bytes	1 ASCII byte (0D _H)	1 Hex byte

If the 'decimal point position' and the 'unit' settings are assigned to a non-null value, the response frame when transmitting measurement contains the decimal point character (2E_H) and the unit that is separated from the measurement value by a space ASCII character (20_H).

- Fast SCMBus response

STX	Status word	Value	Cks	ETX
02 _H	2 Hex bytes	3 signed Hex bytes (2's complement)	Σ of previous bytes and b7 bit set to 1	03 _H

Note: Because values are encoded in signed hexadecimal bytes format (2's complement) some data bytes can be equal to **STX (02_H)** or **ETX (03_H)** or **DLE (10_H)** so before those specific bytes values a **DLE (10_H)** byte is inserted. The **eNod4** address is not transmitted in the frame.

4.4.3 Functional command request (tare, zero...)

- request :

Address	Command	CR	CRC
1 Hex byte	1 Hex byte (command)	1 ASCII byte (0D _H)	1 Hex byte

- response (SCMBus and fast SCMBus) :

Address	Command	CR	CRC
1 Hex byte	1 Hex byte (command)	1 ASCII byte (0D _H)	1 Hex byte

If the command execution is successful, **eNod4** sends back the request frame that has been received as an acknowledgement.

4.4.4 Error frame

In case of an error upon reception of a request, **eNod4** sends back an error frame that contains an error code:

- response (SCMBus and fast SCMBus) :

Address	Error code	CR	CRC
1 Hex byte	1 Hex byte (command)	1 ASCII byte (0D _H)	1 Hex byte

- The error codes are listed below:

<i>Error code</i>	<i>Meaning</i>	<i>Description</i>
<i>FE_H</i>	<i>unknown command</i>	<i>requested command is not supported by eNod4</i>
<i>FF_H</i>	<i>error during command execution</i>	<i>ex. : tare when gross meas.<0</i>

4.5 Address and Baud rate

Address and baud rate identical to Modbus RTU (See § Modbus RTU)

4.6 Product identification

Product identification identical to Modbus RTU (See § Modbus RTU)

4.7 Measurement transmission

Measurement transmission can be triggered by a master request but it might also be triggered and used through the following options:

- transmission triggered by a rising or falling edge on a logical input
- transmission at a configurable period (defined in ms) while a logical input is maintained at a given logical level
- continuous transmission at a configurable period (defined in ms) after a master request. The transmission is then stopped by another master instruction, be careful not to use this mode in half-duplex at a too high rate.

4.8 Continuous transmission

SCMbus and fast SCMbus communication protocols allow **eNod4** to transmit measurements at a user-defined rate without the need for successive master queries. To perform this measurement acquisition mode, it is necessary to set first the 'sampling period' (in ms):

<i>Address SCMbus</i>	<i>Description</i>	<i>Accès</i>	<i>Type</i>
<i>0x003F</i>	<i>SCMbus Measurement transmission period</i>	<i>RW</i>	<i>Uint</i>

A value of 0 implies that measurement transmission is synchronized on the A/N conversion rate. The continuous transmission is triggered and stopped by reception of the following commands:

<i>SCMbus/fast SCMbus functional command</i>	<i>Command code</i>
<i>start net measurement transmission</i>	<i>E0_H</i>
<i>start factory calibrated points transmission</i>	<i>E1_H</i>
<i>start brut measurement transmission</i>	<i>E2_H</i>
<i>stop continuous transmission</i>	<i>E3_H</i>

Note 1: the measurement transmission rate also depends on the baud rate. So, to achieve the fastest transmission, it is necessary to use the highest baud rate.

Note 2: as RS485 is a half-duplex communication medium, it can be a little hard to transmit the 'stop continuous transmission' query if the bandwidth is saturated. Therefore, prefer USB communication channel to reach the highest measurement transmission rate.

4.9 EEPROM error management

EEPROM management identical to Modbus RTU (See § Modbus RTU)

5 CANOPEN

5.1 Physical interface

eNod4 is equipped with a CAN 2.0A compatible interface supporting **CANopen® communication protocol**. The device can be connected to a CAN bus using **CANH** and **CANL** connections. A REF pin can also be connected. Supported baud rates are 50000, 125000, 250000, 500000 and 1000000.

For a complete description of the recommendations about **eNod4** CAN connexion, please refer to documentation “characteristics and functioning”.

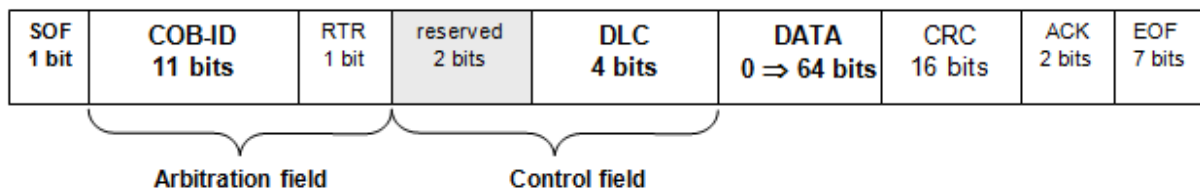
5.2 LED CANopen

The subsequent table describes the meaning of the LEDs for the CAN:

Color	State	Meaning
Red	Single Flash	At least one of the error counters of the CAN controller has reached or exceed the warning level (too many error frame)
	Double Flash	A guard event (NMT-slave or NMT-master) or a heartbeat event (heartbeat consumer) has occurred
	On	The CAN controller is bus off
	Flash	Self-test: while the device is performing its power up testing, the LED shall be flashing red
Green	Blinking	The device is in state PRE OPERATIONAL
	Single Flash	The device is in state STOPPED
	On	The device is in state OPERATIONAL

5.3 Frame format

Every data frame sent on the CAN bus has the following structure:



- Start of frame (SOF) : 1 bit

The beginning of a request or a data frame is indicated by the transmission of one dominant bit.

- **Arbitration field : 12 bits**

This field contains the message COB-ID on 11 bits and the RTR bit, dominant for data frames and recessive for remote frames.

- **Control field : 6 bits**

The first two bits are reserved and must be transmitted as dominant. The four remaining bits encode the size of the transmitted data in bytes. This is called «**Data length code**» (DLC) with $0 \leq \text{DLC} \leq 8$.

- **Data : de 8 à 64 bits**

For each byte, the most significant bit (MSB) is transmitted first.

- **Cyclic Redundancy Check (CRC) : 16 bits**

The result of the CRC calculation is made up of 15 bits that guarantee the integrity of the transmitted message. The last bit is used to delimit the field and always is transmitted as dominant.

- **Acknowledgement (ACK) : 2 bits**

During two bus clock periods, the bus is available for acknowledgement of the message. All the nodes that received the message without error generate a dominant bit. Else, an error frame is generated. The second bit is always recessive.

- **End of frame (EOF) : 7 bits**

The end of the frame is represented by a sequence of 7 consecutive recessive bits.

The CANopen® layer defines particularly the content of the arbitration and the control fields and the data field structure.

5.4 Messages transfers hierarchy

CANopen® is a communication protocol especially dedicated to industrial applications. It allows connecting up to 127 different devices on a same bus giving them the possibility to access the bus at any time. Simultaneous emissions are managed by an arbitration system that uses priority levels.

This control hierarchy of data transfers guarantees that there is no frame collision on the bus while ensuring a high level of reliability in communications. The low priority messages are cancelled and reissued after a delay.

The protocol defines several message types characterized by their COB-ID (Communication Object Identifier) that determines the message priority level. The COB-ID is composed of a function code and the node identifier (between 1 and 127).

The node identifier is the device's address on the network. The function code specifies the priority and the purpose of the message. **Assignment of a particular identifier to each device connected to the bus is mandatory.**

eNod4 supports 6 different message types :

- read/write requests : **SDO** (Service Data Object)
- real time transfers : **PDO** (Process Data Object)
- nodes state management : **NMT** (Network Management)
- warnings : **EMCY** (Emergency)
- synchronization events : **SYNC** (Synchronization)
- node status indications : **Boot-up/Heartbeat** and **Node guarding**

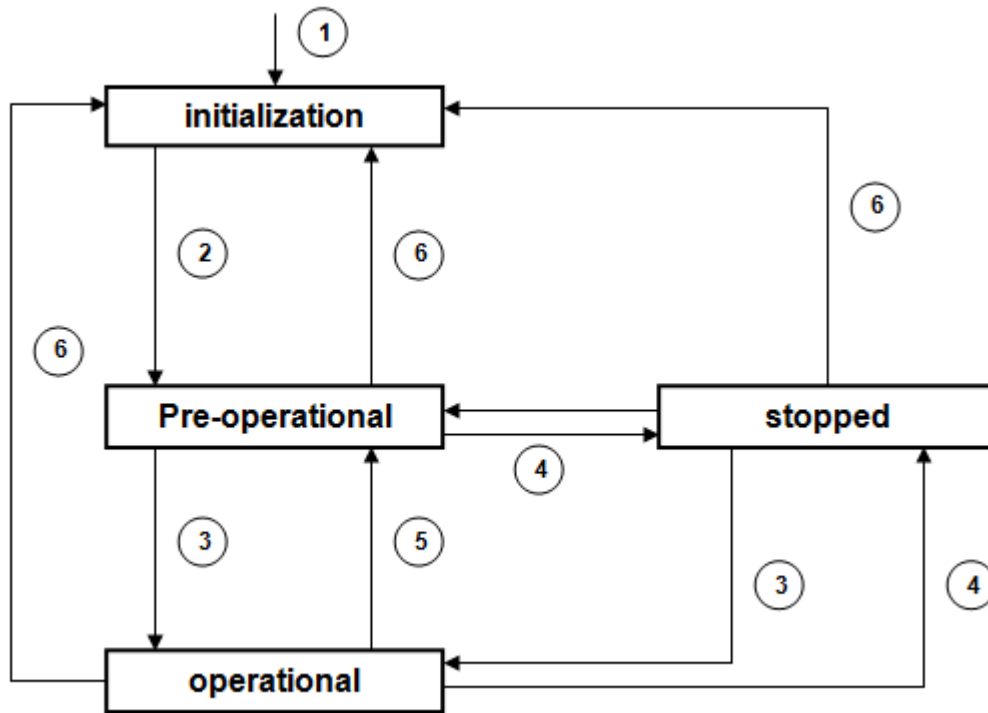
<i>CANopen® messages</i>	<i>COB-ID (hex)</i>
NMT	0
SYNC	80
EMCY	81-FF
TPDO1	181 – 1FF
RPDO1	201 – 280
RPDO2	301 – 380
RPDO3	401 – 480
RPDO4	501 – 580
RPDO5(IO+ version)	681 – 6FF
TPDO2	281 – 2FF
TPDO3	381 – 3FF
SDO (Tx)	581 – 5FF
SDO (Rx)	601 – 67F
Heartbeat/Boot-up	701 – 77F

[5.5 eNod4 status remote management](#)

For the CANopen® network, **eNod4** is considered as a **NMT slave**. It means that its state can be modified by a **NMT master** present on the bus.

As other CANopen® nodes, **eNod4** can be set into one of the four existing states, allowing or forbidding the reception/emission of CAN messages.

These four states constitute the following NMT state machine:



- ⇒ 1 : *eNod4* device power-up
- ⇒ 2 : automatic transition after the end of initialization
- ⇒ 3 : reception of a 'Start Node' indication
- ⇒ 4 : reception of a 'Stop Node' indication
- ⇒ 5 : reception of an 'Enter pre-operational mode' indication
- ⇒ 6 : reception of a 'Reset node' or a 'Reset communications' indication

eNod4 communication capacities for each state are given in the following table :

	<i>Initialization</i>	<i>Pre-operational</i>	<i>Operational</i>	<i>Stopped</i>
<i>SDO</i>		X	X	
<i>PDO</i>			X	
<i>SYNC</i>		X	X	
<i>Emergency</i>		X	X	
<i>NMT</i>		X	X	X
<i>Boot-up</i>	X			
<i>Heartbeat</i>		X	X	X

5.5.1 NMT commands

Except during the initialization phase, *eNod4* is able to handle any NMT master's requests for changing its current state. All these network management messages are constituted the same way: **a two-byte data frame with a COB-ID equal to zero**:



The 2nd byte of the data field contains the node identifier of the device concerned by the request. Its value must be between 0_H and 7F_H. The 0_H value means that the NMT command concern all the nodes of the network.

The 1st byte codes the command sent to the node. There are five existing commands supported:

- ⇒ « **Start node** »: 01_H. **eNod4** is set into **operational** state
- ⇒ « **Stop node** »: 02_H. **eNod4** is set into **stopped** state
- ⇒ « **Reset node** »: 81_H. Resets **eNod4** (with the same effects as a power-up), back into **initialization** state.
- ⇒ « **Reset communication** »: 82_H. Back into **initialization** state and communication parameters reset.
- ⇒ « **Enter pre-operational mode** »: 80_H. **eNod4** is set into **pre-operational** state

5.5.2 Synchronization messages

SYNC messages are emitted on the bus by a producer node (generally the NMT master). This service is unconfirmed so the consumer nodes do not have to respond to SYNC messages. A SYNC message does not carry any data (DLC = 0). **eNod4** is only seen as a SYNC messages consumer whose COB-ID is stored at index 1005_H, sub-index 00_H of the object dictionary.

5.5.3 Emergency messages

eNod4 internal errors are reported via emergency frames. Two types of errors can trigger the transmission of an emergency message:

- communication errors
- A/D converter input signal range exceeded

Every emergency frame is built as follows:

COB-ID	DLC	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
80 _H + ID eNod4	8	emergency code		error register content	additional information				

Emergency message is an unconfirmed service. A frame is emitted when a new error occurs and when it is acknowledged. The table below describes the emergency standard codes supported by **eNod4-T** and the translation of the additional information bytes (in ASCII):

Emergency codes (hex.)	Meaning
0	error acknowledged
3200	voltage error
8120	CAN bus communication error
8130	life guard error
Additional information's (hex.)	
4B4F	no error

474C	<i>life time has elapsed or Heart Beat not received</i>
564F	<i>sensor signal outside of the input signal range</i>
5054	<i>CAN transmitter in error passive state</i>
5052	<i>CAN receiver in error passive state</i>

The error register value is also part of the emergency telegram so as to indicate if other internal errors have been detected.

The number of reported errors is given by an error counter in the **pre-defined error field** located at index 1003_H, sub-index 00_h and the last reported error can be read from the same entry at sub-index 01_h.

5.6 Error control services

CANopen® uses smart mechanisms to control permanently the nodes state on the bus. **eNod4** supports **Boot-up** and **Heartbeat** messages and **Node guarding protocol**. Using both services is not allowed. If both are configured so as to be functional, only the Heartbeat mechanism is used.

5.6.1 Heartbeat and boot-up

eNod4 state control can be achieved through the use of Heartbeat and boot-up mechanisms :

- ⇒ **Boot-up**: this message sent by **eNod4** means that its initialization phase is complete and that the node has entered into **pre-operational** state. It consists in the following frame :

COB-ID	DLC	byte 1
700_H + ID eNod4	1	0

- ⇒ **Heartbeat** :

- **producer mode** : if a Heartbeat period (in ms) different from 0 is set in the entry '*producer heartbeat time*' of the object dictionary, **eNod4** generates at this period a frame containing its state coded on one byte. The corresponding frame is similar to the **Boot-up** mechanism frame:

COB-ID	DLC	byte 1
700_H + ID eNod4	1	eNod4 NMT state

eNod4 NMT state byte can take the different following values :

- ⇒ 04_H : the node is in the **«stopped»** state
- ⇒ 05_H : the node is in the **«operational»** state
- ⇒ 7F_H : the node is in the **«pre-operational»** state

Using Heartbeat protocol allows a NMT master to check that all nodes connected to the bus are working correctly.

- **Consumer mode**: **eNod4** also can be configured so as to monitor the NMT state of one particular node of the network (generally the NMT master). The node ID and a period are defined in the entry '*consumer heartbeat time*' of the object dictionary

If corresponding heartbeat is not received within this time, then **eNod4** sends an emergency telegram and switches to pre-operational state.

5.6.2 Node guarding

Node guarding protocol is another way to check the nodes state. But unlike Heartbeat protocol, it needs requests from a NMT master. In this case, the NMT master sends periodically a remote transmit request (remote frame) to the node with **COB-ID 700_H + ID eNod4**. **eNod4** has to respond by sending a single-byte data frame with its coded state.

This frame is similar to Heartbeat frame but there is an important difference. Most significant bit of the state byte is a toggle-bit. The value of this bit must alternate between two consecutive responses from the NMT slave. The value of the toggle-bit of the first response after the Guarding Protocol becomes active is 0.

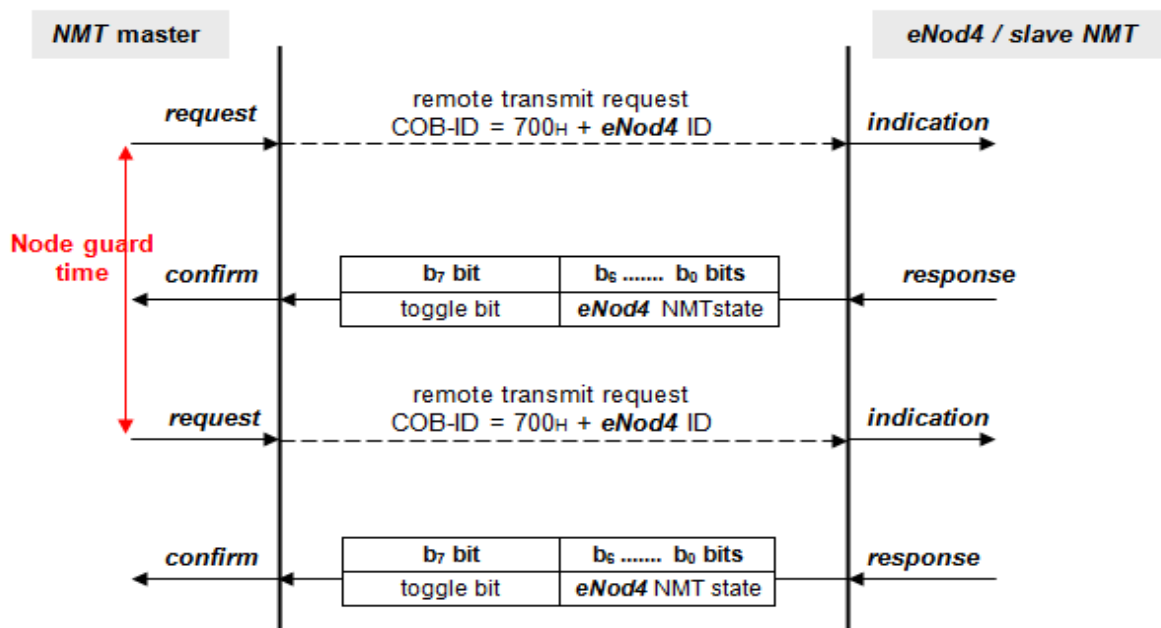
It is only reset to 0 when a 'reset communications' or a 'reset node' command is received. If two consecutive responses have the same value of the toggle-bit, then the new response should be handled as if it was not received by the NMT master.

Two parameters of the object dictionary are necessary to set and define node guarding protocol: the 'guard time' and the 'life time factor':

- ⇒ **Guard time**: this parameter expressed in milliseconds indicates the period with which the node is being polled by the NMT master. This value can be different from one node to another.
- ⇒ **Life time factor**: when node guarding protocol is active, **node life time** is given by multiplication of the guard time and the life time factor.

Node guarding activation is effective when guard time has been set (and if Heartbeat protocol is not used) and after reception of the first remote transmit request. If life time factor is also configured and if no remote transmit request is handled within the node life time, **eNod4 sends an emergency telegram then switches to pre-operational state**. The life guarding error is acknowledged when the state is changed by a NMT command and after reception of a new remote transmit request.

Switching to the stopped NMT state because of a node guarding error may cause **eNod4** to be set into a configurable safety mode where parts of its functioning are inhibited



5.7 Access to the object dictionary

The most important element of a CANopen® compatible device is its **object dictionary (OD)**. Each node object that can be accessed via the bus is part of a table called object dictionary. The dictionary entries can be addressed by a couple of an index (2 bytes) and a sub-index (1 byte) with the following organization:

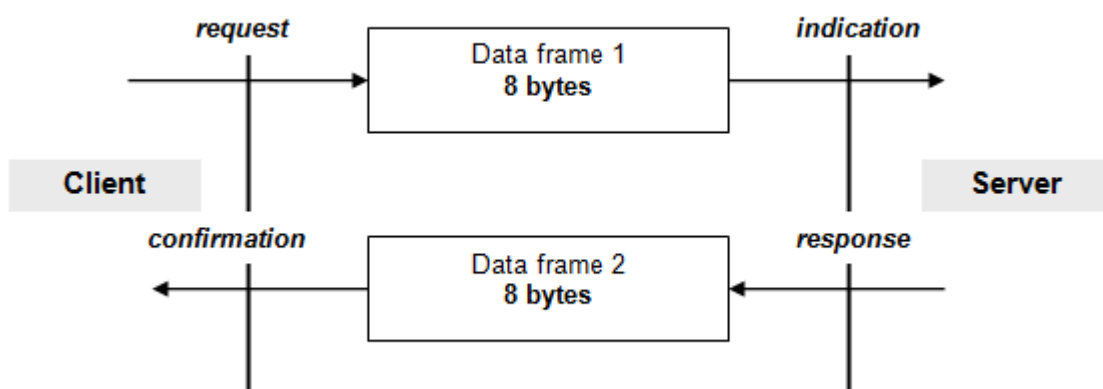
<i>Index (hex.)</i>	<i>Object type</i>
0000	<i>reserved</i>
0001 ⇒ 001F	<i>static data types</i>
0020 ⇒ 003F	<i>complex data types</i>
0040 ⇒ 005F	<i>manufacturer specific complex data bytes</i>
0060 ⇒ 007F	<i>device profile specific static data types</i>
0080 ⇒ 009F	<i>device profile specific complex data types</i>
00A0 ⇒ 0FFF	<i>reserved</i>
1000 ⇒ 1FFF	<i>communication profile area</i>
2000 ⇒ 5FFF	<i>manufacturer specific profile area</i>
5FFF ⇒ 9FFF	<i>standardized device profile area</i>
A000 ⇒ FFFF	<i>reserved</i>

Only the greyed elements of the table are accessible through **eNod4** OD.

The whole object dictionary is accessible and can be configured from usual CANopen® configuration tools. This can be done using **eNod4** available **EDS** file.

5.7.1 SDO communication

The model for SDO communication is a client/server model as described below:



The node that sends the request is the client application whereas **eNod4** only behaves as the server application. There are two types of requests, write and read requests. Both have the same structure:

COB-ID	DLC	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
11 bits	1 byte	Command byte	Index		sub-index	Data			
580 _H or 600 _H + ID eNod4	8	see table	LSB	MSB	/	LSB	-	-	MSB

The client request uses the SDO(Rx) COB-ID (600_H + ID **eNod4** and the server uses the SDO(Tx) COB-ID (580_H + ID **eNod4**).

The **command byte** depends on the requested data length:

Client request	Server response
read data ⇒ 40 _H	43_H ⇒ 4-bytes data
	4B_H ⇒ 2-bytes data
	4F_H ⇒ 1-byte data
write 4-bytes data ⇒ 23 _H	60_H
write 2-bytes data ⇒ 2B _H	
write 1-byte data ⇒ 2F _H	

For a read request, the value of the four last bytes of the frame (data) does not matter.

If an error occurs during a SDO communication **eNod4** responds with the command byte 80_H and the four data bytes contain one of the following SDO abort codes. The data transfer is aborted.

SDO abort codes (hex.)	Description
5040001	<i>SDO command specifier not supported</i>
6010001	<i>unsupported access to an object</i>
6010002	<i>attempt to write a read-only object</i>
6020000	<i>the object does not exist in the object dictionary</i>
6040042	<i>the number and length of the objects to be mapped would exceed PDO length</i>
6040047	<i>impossible operation (for example reading a net/gross value during a tare or a zero)</i>
6070012	<i>data type does not match, length of service parameter too high</i>
6070013	<i>data type does not match, length of service parameter too low</i>
6090011	<i>Sub index object does not exist.</i>
6090030	<i>value range of parameter exceeded</i>
6090031	<i>value of parameter written too high</i>
6090032	<i>value of parameter written too low</i>
8000020	<i>data cannot be stored to the application</i>

5.7.2 PDO communications

SDO protocol is not the only way to access the object dictionary. PDO allow to transfer data without including their index and sub-index in the frame. Both are stored in an OD specific field called PDO mapping.

The model used for PDO transmissions also is different. It is a Producer/Consumer model in which data are sent by a producer node (TPDO) to a consumer node (RPDO) without any confirmation.

Each PDO is described by a combination of two parameters of the OD: the **PDO communication parameters** and the **PDO mapping**. The PDO communication parameters describe the functioning of the PDO and the PDO mapping describes its content. *eNod4* uses **3 TPDO** (2 are programmable) and **5 RPDO**.

The PDO transmission mode can be set in the corresponding object with the following attributes:

- ⇒ **Synchronous:** PDO transmission/reception is triggered by the reception of one or more SYNC messages. Several options are available :
 - cyclic: PDO is sent/received after reception of n ($1 \leq n \leq 240$) SYNC messages.
 - acyclic: PDO is sent at reception of the first SYNC message following a specific device event (activation of a logical input assigned to 'send TPDO' or data variation superior to +/- delta)
 - on remote transmit request : PDO is sent after the first SYNC message following a remote transmit request frame with the PDO COB-ID.
- ⇒ **Asynchronous:** PDO transmission/reception does not depend on the SYNC messages on the CAN bus. Several options are available :
 - on remote transmit request : PDO is sent at reception of a remote transmit request frame with the PDO COB-ID.
 - activation of a logical input assigned to 'send TPDO' or data variation superior to +/- delta)
 - on a timer event : PDO is sent periodically (with an adjustable period).

The following table recaps the trigger modes that can be chosen by entering the hexadecimal code in the PDO communication parameter:

Code (hex)	cyclic	acyclic (event)	synchronous	asynchronous	remote transmit request	Effect
00		X	X		X	<p>PDO transmission/reception after a SYNC message following one of these events :</p> <ul style="list-style-type: none"> - activation of a logical input assigned to 'send TPDO' - mapped object variation superior to +/- delta - Receipt of remote transmit request.
01 – F0 (= n)	X		X		X	<p>PDO transmission after n SYNC messages</p> <ul style="list-style-type: none"> - Or after receipt of remote transmit request following at less one SYNC.
F1 - FB	<i>reserved</i>					
FC			X		X	data update at reception of a remote transmit request and PDO transmission after reception of a SYNC message
FD				X	X	data update and PDO transmission at reception of a remote transmit request
FE				X	X	<p>PDO transmission is triggered by one of these events :</p> <ul style="list-style-type: none"> - activation of a logical input assigned to 'send TPDO' - mapped object variation superior to +/- delta - receipt of remote transmit request <p>more for TPDO2 and 3, functioning is identical to code FF</p>
FF				X	X	<p>Periodic TPDO emission. Period can be configured (min = 1 ms).</p> <p>RPDO handled upon reception</p> <p>TPDO emission after receipt of remote transmit request</p>

- **Note 1:** for RPDO1-2-3-4-5 and TPDO1, only the transmission types FF_H and 00_H are supported. That means data are updated either immediately upon reception (FF_H) or after next **SYNC** following the RPDO reception (00_H). For TPDO1, the FE_H transmission type means that the TPDO1 is emitted by **eNod4** every time it's mapped value changes. The 00_H transmission type is similar but the emission is triggered by the reception of a **SYNC** object.

- ❑ **Note 2:** RPDO1-2-3-4-5 and TPDO1 mapping are configured with default values that cannot be modified.

5.8 CANopen command and response registers

In CANopen® communication protocol, the ‘*command register*’ is mapped into **RPDO1** and the ‘*response register*’ is mapped into **TPDO1**. When in operational NMT state, **eNod4** is able to handle the **functional commands** received through **RPDO1** and the ‘*response register*’ value changes are automatically transmitted through **TPDO1**.

5.9 Communication objects

Some settings are specific as defined by the CANopen® communication specification.

5.9.1 0x1001 / 0x00 : error register

- **access :** RO
- **data type :** unsigned8
- **default value :** 0
- **mappable ? :** N
- **admitted values :** see table below

Description: The device internal errors are indicated by flag bits of this byte. b_0 bit (generic error) is set to 1 if at least one error is detected.

Bit set to 1	Meaning
b_0	generic error detected
b_1	reserved (0)
b_2	A/D converter input voltage error
b_3	reserved (0)
b_4	CAN bus communication error
b_5	reserved (0)
b_6	reserved (0)
b_7	EEPROM error

5.9.2 0x1003 : Pre-defined error field

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	reported errors counter	R/W	0	N	unsigned8
0x01	last reported error	RO	0	N	unsigned32

Description: This entry of the object dictionary stores the errors that have been reported by emergency telegrams. The reported errors counter (sub-index 00_H) is accessible through write or read request but 0 is the only allowed value for writing transactions. By writing a zero to this sub-index, the error counter is reset and the last reported error (sub index 01_H) is erased. An attempt to write another value is ignored and **eNod4** answers the SDO abort code 0x06090030.

5.9.3 0x1005 / 0x00 : synchronization messages COB-ID

- access : R/W
- data type : unsigned32
- default value : 0x80
- mappable ? : N
- admitted values : 0x80 or from 0x7E0 up to 0x7E3

Description: This object contains the message COB-ID value supported by **eNod4** as synchronization messages (used for PDO activating).

5.9.4 0x100C / 0x00 : guard time

- access : R/W
- data type : unsigned16
- default value : 0
- mappable ? : N
- admitted values : from 0 up to 65535

Description: The 'life guard' is one of the two parameters used by the node guarding protocol (errors detection). When 'Heartbeat time' is inactive and 'life guard' is different from 0, **eNod4** responds to NMT master periodic (period equal to life guard) remote transmit requests.

5.9.5 0x100D / 0x00 : life time factor

- access : R/W
- data type : unsigned8
- default value : 0
- mappable ? : N
- admitted values : from 0 up to 255

Description: By multiplying the 'life guard' by the 'life time factor', the node life time (cf. §4) can be determined. When node guarding is active, if the node has not be polled within this duration (in ms), **eNod4** state is set to stopped. **eNod4** behavior while stopped can be configured via the object at index 0x4800.

5.9.6 0x1010 : Store parameters

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	largest sub-index	RO	0x01	N	unsigned8
0x01	save all parameters	R/W	0x01	N	unsigned32

In CANopen® communication protocol, storing all settings into **eNod4** EEPROM memory requires writing through SDO the ASCII string « save » (65766173_H) to sub-index 0x01 of entry 0x1010 of the object dictionary (called 'save all parameters' in the EDS file).

- 0x65 ⇒ e
- 0x76 ⇒ v
- 0x61 ⇒ a
- 0x73 ⇒ s

When accessing to sub-index 1 with a SDO read request, **eNod4** cell responds with a value of 1 that means that parameters are stored in non-volatile memory only on request.

Note: Restore to default settings functional command is not available in CANopen® communication protocol.

5.9.7 0x1014 / 0x00 : Emergency COB-ID

- access : RO
- data type : unsigned32
- default value : 0x81
- mappable ? : N
- admitted values : from 0x81 up to 0xFF

Description: The COB-ID of emergency messages transmitted by **eNod4** is stored at this index. Its value automatically is updated if the node identifier is modified.

5.9.8 0x1016 : Heartbeat consumer time

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	max. sub-index	RO	1	N	unsigned8
0x01	heartbeat consumer time	RW	0	N	unsigned32

Description: **eNod4** is able to monitor the Heartbeat generated by another node (see §4) of the network (in general the master). Two settings must be defined through the object at sub-index 0x01:

- ⇒ the heartbeat time period, coded on the 16 LSB bits
- ⇒ the node ID of the heartbeat producer to monitor, coded on the 16 MSB bits (from 0x01 up to 0x7F)

5.9.9 0x1017 / 0x00 : Heartbeat producer time

- access : R/W
- data type : unsigned16
- default value : 0
- mappable ? : N
- admitted values : from 0 up to 65535

Description: If a period different from 0 is written into this index, **eNod4** periodically generates a Heartbeat frame. It is expressed in ms and must be comprised between 1 and 65535.

5.9.10 0x4800 : Safety mode

- access : R/W
- data type : unsigned8
- default value : 0H
- admitted values : see table below

Description: This entry defines **eNod4** functioning when in stopped NMT state. The safety mode is used when the bit b0 of the byte is set to 1. The functioning mode is then inhibited and outputs logical level are given by b1, b2, b3 and b4 bits. The **eNod4** leaves the safety mode upon reception of a new NMT command.



Warning: In safety mode, when the **eNod4** is in stopped NMT state, functioning is also inhibited using Modbus and SCMBus communication. For example, it is not possible to tare or make a zero using RS 485 AUX port, or eNodView connected via USB.

<i>b0</i>	<i>Effect</i>	<i>Notes</i>
0	safety mode disabled	only valid in stopped state
1	safety mode enabled	No action possible also using Modbus or SCMBus communication
<i>b1</i>		
0	output 1 inhibited	depending on the chosen logic
1	output 1 set active	
<i>b2</i>		
0	output 2 inhibited	depending on the chosen logic
1	output 2 set active	
<i>b3</i>		
0	output 3 inhibited	depending on the chosen logic
1	output 3 set active	
<i>b2</i>		
0	output 4 inhibited	depending on the chosen logic
1	output 4 set active	

5.9.11 Error behavior

<i>Sub-index</i>	<i>Description</i>	<i>Access</i>	<i>Default value</i>	<i>Mappable (PDO) ?</i>	<i>Type</i>
0x00	Number of error classes	LS	1	N	unsigned 8
0x01	communication error	L/E	0	N	unsigned 8

Description: If a communication error occurs in operational mode, **eNod4** shall enter by default autonomously the pre-operational mode.

Alternatively, by error class coding in object 0x1029 **eNod4** can enter in the following states:

- **0: Pre-operational**
- **1: No state change**
- **2: Stopped. refer to 'Safety mode'**

5.10 PDO-related communication objects

5.10.1 RPDO default mapping

- [0x1600 : RPDO1 mapping parameters](#)

<i>Sub-index</i>	<i>Description</i>	<i>Access</i>	<i>Default value</i>	<i>Type</i>
0x00	number of supported objects	RO	1	unsigned8

0x01	1 st object mapping	RO	0x20030008 (command register)	unsigned32
------	--------------------------------	----	----------------------------------	------------

When in operational NMT state, **eNod4** is able to handle the **functional commands** received through **RPDO1** and the '**response register**' value changes are automatically transmitted through **TPDO1**.

- [0x1601 : RPDO2 mapping parameters](#)

Sub-index	Description	Access	Default value	Type
0x00	number of supported objects	RO	1	unsigned8
0x01	1 st object mapping	RO	0x30010120 (calibration load 1)	unsigned32

- [0x1602 : RPDO3 mapping parameters](#)

Sub-index	Description	Access	Default value	Type
0x00	number of supported objects	RO	2	unsigned8
0x01	1 st object mapping	RO	0x25000020 (zero offset)	unsigned32
0x02	2 nd object mapping	RO	T, C : 0x30050120 D, F, B : 0x30060120 (span adjusting coefficient)	unsigned32

- [0x1603 : RPDO4 mapping parameters](#)

Sub-index	Description	Access	Default value	Type
0x00	number of supported objects	RO	2	unsigned8
0x01	1 st object mapping	RO	0x30020020 (maximum capacity)	unsigned32
0x02	2 nd object mapping	RO	0x30040020 (sensitivity)	unsigned32

- [0x1604 : RPDO5 mapping parameters](#)

Sub-index	Description	Access	Default value	Type
0x00	number of supported objects	RO	1	unsigned8
0x01	1 st object mapping	RO	0x50500010 (External value to control analog output)	unsigned32

Note: RPDO5 is default deactivated. To use it you have first to activate it setting 0 to the 32nd bit of the RPDO5 COB-ID object (0x1404/0x01).

5.11 Product identification

The device identification settings are a part of the communication profile area of the object dictionary.

- [0x1009 : Manufacturer hardware version](#)

Sub-index	Description	Access	Default value	Type
0x00	manufacturer hardware version	RO	0x32302E31 (1.04)	visible string

- [0x100A : Manufacturer software version](#)

Sub-index	Description	Access	Default value	Type
0x00	manufacturer software version	RO	0x30302E31 (1.00)	visible string

- [0x1018 : Identity object](#)

Sub-index	Description	Access	Default value	Type
0x00	max. sub-index	RO	0x04	unsigned8
0x01	vendor ID	RO	0x00000142	unsigned32
0x02	product code	RO	0x000816E0	unsigned32
0x03	revision number	RO	0x00010000	unsigned32
0x04	serial number	RO	0x000186A1	unsigned32

5.12 Measurement transmission

CANopen® includes smart transmission mechanisms that are presented in § “PDO communication”. All the measurements can be exchanged either through SDO read requests or through TPDO.

TPDO trigger sources are described in a table of § “PDO communication” and all these communication modes are possible for measurement transmission. Thanks to the various TPDO transmission types, **eNod4** offers possibility to have a high measurement transmission rate (up to 1000 meas/s) or to limit the bus occupation by causing the exchange on an event.

5.13 EEPROM error management

Functioning and calibration parameters are stored in EEPROM NOV RAM. After every reset the entireness of parameters stored in EEPROM is checked. If a default appears, measurements are set to 0xFFFF and default is pointed out to object dictionary (see bit 6 of object 0x5003/0x00).

6 CANOPEN TPDO MAPPING

6.1 Default TPDOs Mapping

- 0x1A00 : TPDO1 mapping

Sub-index	Description	Access	Default value	Type
0x00	number of supported objects	RO	0x01	Unsigned8
0x01	1 st mapped object	RO	0x20040008 (response register)	Unsigned32

In NMT operational state, **eNod4** is able to execute any received command through RPDO1 and consequently changes of response register are sent through TPDO1

- 0x1A01 : TPDO2 mapping

Sub-index	Description	Access	Default value	Type
0x00	number of supported objects	R/W	0x02	Unsigned8
0x01	1 st mapped object	R/W	0x50010020 (Gross measurement)	Unsigned32
0x02	2 nd mapped object	R/W	0x50030010 (Measurement status)	Unsigned32
0x03	3 rd mapped object	R/W	0	Unsigned32

- 0x1A02 : TPDO3 mapping

Sub-index	Description	Access	Default value	Type
0x00	number of supported objects	R/W	0x03	Unsigned8
0x01	1 st mapped object	R/W	0x50000020 (Net measurement)	Unsigned32
0x02	2 nd mapped object	R/W	0x51000008 (logical inputs level)	Unsigned32
0x03	3 rd mapped object	R/W	0x52000008 (logical outputs level)	Unsigned32

Note : TPDO2 and TPDO3 mapping are programmable. TPDO1 mapping is not programmable.

To set a new mapping, the procedure is as following:

- Set **eNod4-F** in '**pre-operational mode**' (default state after a reset or a power on).
- Disable current TPDO mapping setting to zero **number of supported objects** (sub-index 0x00).
- Write new mapping.
- Write in sub-index 0x00 **number of supported objects** the exact number of objects to map.
- Save in EEPROM (SAVE command in object 0x1010 sub-index 0x00).

7 PROFIBUS DPV1

7.1 Physical interface

An **eNod4** device compatible version can be connected to a Profibus DPV1 network thanks to the SUBD 9-pin female connector. **eNod4** supports baud rates between 9600 kbps and 12 Mbps with automatic detection.

For a complete description of the recommendations about **eNod4** Profibus DPV1 connection, please refer to “*Characteristics and functioning*” documentation.

7.2 GSD file

eNod4 capabilities for Profibus communication are described in the corresponding GSD file (that can be easily read as an ASCII text file). Here are the main information’s contained in this file:

- *product and vendor identifications*
- *hardware and software versions*
- *supported baud rates*
- *description of the IN/OUT cyclic modules*
- *DPV0/DPV1 supported functions*
- *name of the associated bitmap*

7.3 Cyclic exchanges

7.3.1 Cyclic inputs modules

eNod4 GSD file defines several modules that can be integrated to the Profibus DP cyclic frames. All types of measurements processed by **eNod4** can be selected (net/gross/factory calibrated measurement). Moreover it is possible to add the data that contain information about the measurement or the device status.

7.3.2 Cyclic inputs/outputs modules

eNod4 cyclic outputs might be used in combination with cyclic inputs so as to request specific commands to **eNod4** (like tare, calibration...) or to modify the values of some settings (for example, set point values). A simple protocol described below allows writing or reading data by writing commands into **eNod4** cyclic outputs.

- The module “**Reg. Command/Response**” uses the eNod4 functional command mechanism defined in another chapter. The only difference concerns the “reset” and “restore default parameters” commands which cannot be sent via cyclic exchanges immediately after an eNod4 reboot. To use them with this cyclic exchange module, another command should be used before (cancel tare for example).

The ‘**command/response register**’ described by **eNod4 GSD** file is constituted by 2 IN/OUT bytes and is working almost the same way. Its functioning will be detailed later in this document.

- The ‘**read/write request register**’ described by **eNod4** GSD file is constituted by 6 IN/OUT bytes:

IN	OUT
Transaction status (2 bytes)	Transaction request (2 bytes)
Data read/written (4 bytes)	Data to be written (4 bytes)

- ⇒ the Profibus DP master can transmit a read or write request to **eNod4** by writing a specific code (see the codes listed in the appendix) into the transaction request register.
- ⇒ for a write request, the 4 following OUT bytes can be used so as to enter the new value of the accessed setting
- ⇒ **eNod4** IN are then updated:

- Transaction status is set to 0xFFFF in case of an error otherwise it takes the same value as the one entered in the transaction request word.
 - For a read transaction, the value of the requested setting is copied into the four IN following bytes.
 - For a write transaction the value of the data to be written is copied into the four IN following bytes.
 - For 16-bits size data, the data is read/written through the 16 last bits. Ignore the 16 upper bits.
- ❑ **Note:** For writing request, you must first send the data to be written and then the writing request.
- **Example**
 - **Reading Maximum capacity value:**
 1. Send 0x0420 to the *Transaction request register*,
 2. Get the *Transaction status* and the read data into the *Data read/written register*.
 - **Writing Maximum capacity to 500000:**
 1. Send 0x0007A120 in the *Data to be written register*,
 2. Then, send 0x0421 to the *Transaction request register*,
 3. Get the *Transaction status* and the written data into the *Data read/written register*.
- ❑ **Note:** the transaction request register must be set to 0x0000 before every new transaction.

7.4 Acyclic exchanges

For DPV1 class 1 and class 2 masters, another mechanism is available to write/read data into/from **eNod4** device. Acyclic exchanges are supported authorizing a class 1 master to send acyclic requests so as to read or write data into memory area defined by a couple of slot/index.

A class 2 master can also send the same requests after initiating a connexion (up to 2) with **eNod4**. The communication happens without disturbing the cyclic data exchange established between the device and the class 1 master.

- ❑ **Note:** be careful when using acyclic read/write requests. The 4-bytes size data are transmitted as 2x16 bits blocks that need to be inverted so as to keep their consistency. For example, 500 000_d (0007A120_H) is read/written through acyclic requests as A1200007_H.

7.5 eNod4 Profibus DP features

7.5.1 Sync

The *Sync* command can be transmitted from a master to one or more slaves. When receiving this command, **eNod4** device is set into *Synchro* mode. Thus, the cyclic outputs of all the addressed slaves are maintained in their current state. The cyclic outputs state is not updated until a new *Sync* command is received. The *Synchro mode* stops upon reception of a *Unsync* command.

7.5.2 Freeze

The reception of a *Freeze* command causes **eNod4** cyclic inputs to be frozen. Their state is not updated until a new *Freeze* command is received. The *Freeze mode* stops upon reception of an *Unfreeze* command.

7.5.3 Fail-safe

eNod4 supports the Profibus DP *Fail-safe* mode. When the Profibus DP master asks **eNod4** to switch to *Fail-Safe* mode, **eNod4** functioning changes according to following rules:

- ⇒ **eNod4** Profibus DP cyclic inputs goes on being refreshed
- ⇒ **eNod4** Profibus DP cyclic outputs are no more received

eNod4 remains as a “read-only device” until the master goes back to a normal working mode.

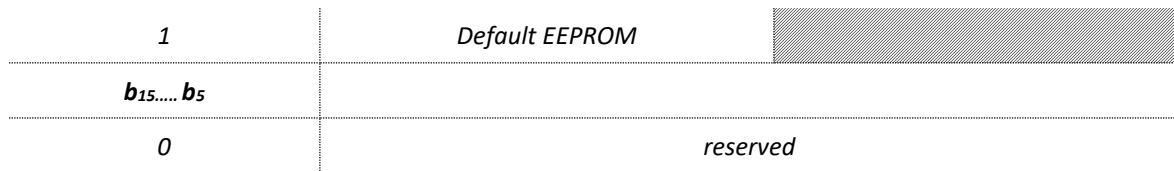
7.5.4 Profibus DP standard and extended diagnoses

eNod4 diagnosis frame is composed of a standard and a specific (called extended) parts that allow to inform the Profibus DP master about the device functioning on the communication bus and about device internal errors. The Profibus diagnosis frame has the following structure:

bytes 1-4			bytes 5-6			byte 7	bytes 8-9
standard diagnosis						length of extended diagnosis	extended diagnosis content
status 1	status 2	status 3	Address	Ident Hi	Ident Low	03	XX XX (see table below)

- ⇒ **status 1 & status 2 bytes** : both bytes describe **eNod4** current state from the Profibus point of view. Bit b_3 of status 1 byte is set to 1 if the extended diagnosis contains one or several errors.
- ⇒ **status 3 byte** : always 00_H
- ⇒ **Address** : Profibus address if the DP master that parameterized **eNod4** device
- ⇒ **Ident. High/Ident. Low bytes** : contain **eNod4** ident number (0D2D_H)
- ⇒ **extended diagnosis content** :

bits	meaning	note
b₀		
0	/	
1	input analog signal out of the A/D conversion range (negative quadrant)	
b₁		
0	/	
1	input analog signal out of the A/D conversion range (positive quadrant)	
b₂		
0	/	see §8 for the ‘maximum capacity’ setting description
1	gross meas.< (- max capacity)	
b₃		
0	/	see §8 for the ‘maximum capacity’ setting description
1	gross meas. > (max capacity)	
b₄		
0	EEPROM OK	



7.6 Product identification

As a DPV1 compatible device, **eNod4** supports record 0 of Identification and Maintenance (I&M0). A Profibus master can access its content through standard DPV1 request.

7.7 Measurement transmission

In Profibus DPV1 communication protocol, measurements (except for tare value) can be included in the cyclic input frame. The modular slave structure allows the user to select the variables according to his needs. These data cannot be accessed through acyclic requests.

7.8 EEPROM error management

Functioning and calibration parameters are stored in EEPROM NOV RAM. After every reset the entireness of parameters stored in EEPROM is checked. If a default appears, measurements are set to 0xFFFF and default is pointed out into module 1 and bit b₄ of extended diagnoses register.

8 PROFIBUS MODULES LIST

Name	Provided Size in octet	Provided Data	Consumed Size in octet	Consumed Data	Module number
Status + Gross Meas.(6 bytes IN)	6	Measurement status (2 bytes)	0	NA	1
		Gross measurement (4 bytes)			
Net Measurement (4 bytes IN)	4	Net Measurement	0	NA	2
Factory calib. Meas.(4 bytes IN)	4	Factory calibrated Meas.	0	NA	3
Logical I/O level (2 bytes IN)	2	Logical I/O level	0	NA	4
Measurement status (2 bytes IN)	2	Measurement Status	0	NA	5
Command/Resp Reg(2 bytes IN/OUT)	2	response register	2	command register (see Functional Commands)	6
R/W request Reg.(6 bytes IN/OUT)	6	Transaction status (2 bytes)	6	Transaction request (2 bytes)	7
		Data read/written (4 bytes)		Data to be written (4 bytes)	
Ana. output (2 b IN/OUT)	2	External value to control analog output (2 bytes)	2	External value to control analog output (2 bytes)	8

The module “**Command/Resp Reg**” uses the mechanism of **eNod4** functional commands defined in another chapter. The only difference is for “reset” and “Restore default settings” commands which cannot be sent via cyclic exchanges immediately after a restart of **eNod4**. To be able to use these commands, it must first be processed another command (“cancel Tare” for example).

Note: The “Command register” data **must be** set to 0x0000 before each new command.

The “**Ana. Output(2 b IN/OUT)**” module allows writing directly the analog output value. This is only possible when the analog output function assignment is set to « Level on request ».

9 MEASUREMENT AND STATUS

Name	Modbus address	Acyclic DPV1 slot/index	Profibus cyclic IN/OUT	CANopen® index/sub-index	Type	Access
Measurement status	0x007D	/	See modules list	0x5003 / 0x00 (M)	Uint	RO
Gross measurement	0x007E	/	See modules list	0x5001 / 0x00 (M)	Long	RO
Tare value	0x0080	/	R:0x0470 W:/	0x5004 / 0x01 (M)	Long	RO
Net measurement	0x0082	/	See modules list	0x5000 / 0x00 (M)	Long	RO
Factory calibrated points	0x0084	/	See modules list	0x5002 / 0x00 (M)	Long	RO
Preset Tare	0x0095	0x03 / 0x08	R:0x0496 W:0x0497	0x5004 / 0x02 (M)	Ulong	RW
Defective measurement debounced time	0x0A48	0x06 / 0x0D	R:0x0206 W:0x0207	0x4509/0x06	Uint	RW
Defective measurement alarm activation time	0x0A49	0x06 / 0x0E	R:0x0208 W:0x0209	0x4509/0x07	Uint	RW
Sensor input control reference	0x0A44	0x0A / 0x00	R: 0x044C W: 0x044D	0x5004 / 0x03 (M)	long	RW
Sensor input control result	0x0A46	0x0A / 0x01	R: 0x024E W: /	0x5004 / 0x04 (M)	Int	RO
Sensor input control result max. tolerance	0x0A47	0x0A / 0x02	R: 0x020A W: 0x020B	0x5004 / 0x05	Uint	RW

Note: eNod4 Ethernet, see Ethernet I/O scanning chapter.

9.1 Measurement transmission

The **eNod4** transmits measurement after signal and data processing through different protocols available. The accessible variables are:

9.1.1 Gross measurement

The ‘*gross measurement*’ stands for the digital value after measurement scaling. It is affected by all the ‘zero’ functions (power-up zero, zero tracking and zero requests).

9.1.2 Net measurement

The ‘*net measurement*’ stands for the digital value after measurement scaling and tare subtraction.

9.1.3 Tare value

The ‘*tare value*’ stores the calibrated value that is subtracted from the ‘*gross measurement*’ so as to give the ‘*net measurement*’.

9.1.4 Factory calibrated points

The ‘*factory calibrated points*’ contains the measurement value without the user calibration layer. It is directly linked to the analog input voltage.

9.1.5 Preset Tare value

A previous calculated tare can be restored using this variable.

9.1.6 Measurement status

The measurement status contains information on eNod4 measurement parameters.

The 'measurement status' bytes contain information about every measurement processed by **eNod4**. See the flags meaning in the table below:

<i>bits</i>	<i>Meaning</i>	<i>Note</i>
<i>b₁ b₀</i>		
00	<i>gross measurement</i>	
01	<i>net measurement</i>	<i>only in SCMBus/fast communication protocols</i>
10	<i>factory calibrated measurement</i>	<i>not significant otherwise (00)</i>
11	<i>tare value</i>	
<i>b₃ b₂</i>		
00	<i>measurement OK</i>	
01	<i>Defect: sensor input control result out of tolerances OR Sensor input control command in progress OR failed (timeout) OR Sensor input reference command in progress</i>	<i>causes a logical output assigned to the 'defective measurement' function to be set active. Causes the analog output assigned to a weight or flow rate image to be set in error mode.</i>
10	<i>gross meas. < (- max capacity) OR gross meas. > (max capacity)</i>	
11	<i>analog signal out of the A/D converter input range</i>	
<i>b₄</i>		
0	<i>motion</i>	<i>causes an output assigned to the 'motion' function to be set active</i>
1	<i>no motion</i>	
<i>b₅</i>		
0	<i>measurement out of the ¼ of division</i>	
1	<i>zero in the ¼ of division</i>	
<i>b₆</i>		
0	<i>EEPROM OK</i>	<i>See Note 1</i>
1	<i>EEPROM failure</i>	
<i>b₇</i>		
0	<i>reserved</i>	<i>1 in SCMBus and fast SCMBus, 0 otherwise</i>
1		
<i>b₈</i>		
0	<i>IN1 Initial level</i>	

<i>bits</i>	<i>Meaning</i>	<i>Note</i>
1		
b₉		
0	<i>IN2 logical level</i>	
1		
b₁₀		
0	<i>OUT1 logical level</i>	
1		
b₁₁		
0	<i>OUT2 logical level</i>	
1		
b₁₂		
0	<i>OUT3 logical level</i>	
1		
b₁₃		
0	<i>OUT4 logical level</i>	
1		
b₁₄		
0	<i>no tare</i>	
1	<i>at least a tare has been processed</i>	
b₁₅		
0		
1	<i>reserved</i>	<i>1 in SCMBus and fast SCMBus, 0 otherwise</i>

Note 1: Functioning and calibration parameters are stored in EEPROM. After every reset the entireness of parameters stored in EEPROM is checked. If a defect appears, measurements are set to 0xFFFF and defect is pointed out in measurement status. Causes a logical output assigned to the 'defective measurement' function to be set active. Causes the analog output assigned to a weight or flow rate image to be set in error mode.

9.2 Weighing diagnosis

9.2.1 Global weighing diagnosis

An internal alarm flag reflects the integrity of the whole measurement chain. It's used to set logical output active or optional analog output in an error mode in order to warn about any defection on the measurement chain (defective measurement).

This variable is set active when at least one of the followings conditions occurs:

- all that set bit2 or bit3 of **Measurement status**:
 - sensor input control result out of tolerances
 - sensor input control command in progress
 - sensor input control command failed (timeout)
 - sensor input reference command in progress
 - gross meas. < (- max capacity)
 - gross meas. > (max capacity)
 - analog signal out of the A/D converter input range
- the one that set bit6 of **Measurement status**: EEPROM failure

This internal alarm flag is featured with adjustable specific de-bounced time and minimal activation time:

9.2.1.1 Defective measurement debounced time

The internal alarm flag is set active only after error conditions have always been true during this de-bounced time. It's expressed in ms.

9.2.1.2 Defective measurement alarm activation time

The internal alarm flag remains active for this minimal "*defective measurement alarm activation time*" when it come to be active and whatever the error conditions are during activation. It is expressed in ms.

9.2.2 Sensor input control

eNod4 features a weighing diagnosis system allowing to check the integrity of analog sensor input by electrically simulating a load, resulting to a simulated weight value. This diagnostic system can be used together with the others defects detection systems in order to achieve overall integrity check of the measurement chain. This system involves two phases initiated by the user:

- The first, just after user calibration, allows taking a simulated reference weight value when the measuring chain integrity is OK.
- The second, when the user wants to check the integrity of the system, allows to make the difference between a new simulated weight value and the reference. Then this difference can be compared with a dedicated maximum tolerance value.

9.2.2.1 Sensor input control reference

Reference value expressed in factory calibrated points for the sensor(s) input control test. The value is automatically determined and stored after executing the **sensor input reference** command. When the **sensor input reference** command is in progress the bits b3b2 in the **Measurement status** are set to 0b01. Its default value is zero.

9.2.2.2 Sensor input control result

Result of sensor(s) input control test expressed in 1/10 of user weight unit. Its value is automatically determined and stored after executing the **sensor input control** command. This test result represents the weight difference between the reference value and the current test value. It is set to -1 when the **sensor input control** command is in progress or the command failed, these conditions cause the bits b3b2 in the **Measurement status** to be set to 0b01. Its default value is zero.

9.2.2.3 Sensor input control result max. tolerance

The **Sensor input control result** variable is compared with the **Sensor input control result max. tolerance** parameter which is expressed in 1/10 of user weight unit and has a default value of 30. If the **sensor input control result** value is greater than or equal to **Sensor input control result max. tolerance** then the bits b3b2 in the **Measurement status** are set to 0b01.

10 PROCESSING FUNCTIONAL COMMANDS

Name	Modbus address	Acyclic DPV1 slot/index	Profibus cyclic IN/OUT	CANopen® index/sub-index	Type	Access
Command register	0x0090	/	See modules list	0x2003 / 0x00 (M)	Uint	RW
Response register	0x0091	/	See modules list	0x2004 / 0x00 (M)	Uint	RO

10.1 Principles

The **eNod4** is able to handle several functional commands thanks to a couple of registers (except in SCMBus protocols):

the command register : dedicated to accept the functional commands

the response register : gives the state of the command currently being processed by **eNod4** (no command, in progress, finished, failed)

- **00_H** ⇒ free to accept a new command
- **01_H** ⇒ command execution in progress
- **02_H** ⇒ command execution complete
- **03_H** ⇒ error during command execution

Note 1: IMPORTANT except in SCMBus/fast SCMBus protocols, to accept a new command, the command register must be set to **00_H** first. This causes the response register to be set back to **00_H**.

10.2 Functional commands list

Functional command	Command code	Note
Set to idle (00_H) response register	00 _H	See § above
Switch legal sealing	CB _H	Putting in service the instrument in legal for trade use
Clear DSD memory	CC _H	Cannot recover
Weighing result acquisition	CD _H	
DSD read	CE _H	
Backward DSD read	CF _H	
reset*	D0 _H	
EEPROM Back up	D1 _H	
restore default settings	D2 _H	
zero*	D3 _H	
tare*	D4 _H	
cancel tare*	D5 _H	
cancel last command	D6 _H	
theoretical scaling	D7 _H	
zero adjustment	D8 _H	
start physical calibration	D9 _H	physical calibration procedure
calibration zero acquisition	DA _H	
segment 1 acquisition	DB _H	
segment 2 acquisition	DC _H	
segment 3 acquisition	DD _H	
Store calibration	DE _H	end of calibration (physical/theoretical) procedure
OUT1 activation/deactivation*	E6 _H	only possible if the outputs are assigned to the associated function
OUT2 activation/deactivation*	E7 _H	
OUT3 activation/deactivation*	E8 _H	
OUT4 activation/deactivation*	E9 _H	
zero offset	F0 _H	
Preset tare*	F2 _H	
Sensor input reference	EF _H	
Sensor input control	FD _H	

Note: Only the commands with a * can be handled by **eNod4** in SCMBus and fast SCMBus protocols.

10.3 Functional commands description

10.3.1 Switch legal sealing

When the legal for trade switch of the instrument is activated (see § *legal for trade switch*), this command allows a software sealing of the eNod4 module when putting in service the instrument. When activated the read only bit b1 of MSB '*Legal for trade switch and version*' variable is modified. Access to some specific functions / parameters becomes forbidden (functioning mode, calibration, filtering, stability...). The software sealing is constituted of both *legal for trade counter* and *legal for trade checksum* values, their values have to be indelibly labelled on the instrument in order to verify that their values are kept unchanged when official verification occurs.

10.3.2 Clear DSD memory

There is no reason to use this command when the instrument is in service. This command will fail anyway when the legal for trade sealing is activated.

This function allows to reinitiate the DSD memory when reconditioning the instrument for a new and different use with new parameters. This command's effect cannot recover.

10.3.3 Weighing result acquisition

When a weighing result acquisition command is carried out, eNod4 first verifies that all the acceptance conditions of the weighing result are met.

If the acceptance conditions of the weighing result are fulfilled, then an automatic registration process is performed in the eNod4 built-in alibi memory (DSD). The data is immediately saved in a FIFO buffer in FRAM type memory (fast access) and then stored permanently in Flash memory in a second time. Otherwise, for example if the stability conditions are not reached in the allowed period of 5 seconds, the registration process fails, a command failure message is returned. There is no DSD storage.

A maximum of 130816 records can be stored permanently. Each record is identified by a unique number of 32-bit size. This identifier is incremented each time a weighing result is acquired and recorded or transmitted.

The minimum time between two DSD recording operations is 50ms.

Otherwise, for example if the stability conditions are not reached in the allowed period of 5 seconds, the registration process fails, a command failure message is returned. There is no DSD storage.

10.3.4 DSD read



it is impossible to read a DSD record simultaneously through the specific menu of eNodTouch and one of the communication interfaces of eNod4 (priority HMI).

This command reads a DSD record. You must first enter the ID number of the corresponding weighing record in the *DSD current record ID to read* variable.

From firmware version **V1.24 (Standard)** and **V2.12 (Ethernet)**, when a Weighing result acquisition command is successful, the following variables take immediately the values of the last acquisition done:

- Read record ID
- Net weight in read record
- Tare value in read record
- Weighing status in read record
- Record checksum of read record

10.3.5 Backward DSD read



it is impossible to read a DSD record simultaneously through the specific menu of eNodTouch and one of the communication interfaces of eNod4 (priority HMI).

This command decrements the ID number of the weighing record present in the *DSD current record ID to read* variable and then reads this DSD record.

10.3.6 Reset

The 'reset' functional command execution is similar to the device power-up. This reboot phase is necessary if the address or/and the baud rate are modified and some settings changes are only taken into account after an EEPROM storage (see § *EEPROM Storage*) followed by a reset (hardware or software).

10.3.7 EEPROM storage

eNod4 configuration and calibration are stored in a non-volatile memory (EEPROM). If changes are made in the device configuration, sending to **eNod4** the 'EEPROM storage' functional command will allow **eNod4** to keep these modifications after a power shutdown or the reception a 'reset' functional command.

Moreover the settings listed below need to be stored and will only be taken into account at the next device reboot:

- span adjusting coefficient
- calibration place **g** value
- place of use **g** value
- stability criterion
- legal for trade activation switch
- power-up zero
- A/D conversion rate
- communication protocol
- Zero (if option is activated / set b_0 equal to 1 in « Options » object)
- Tare (if option is activated / set b_1 equal to 1 in « Options » object)

10.3.8 Restore default settings

The 'restore default settings' command causes **eNod4** to be set back to its default configuration. The default configuration corresponds to the one on delivery that means with factory settings. Be careful when using this command, all the default settings are recovered including the stored calibration and the legal for trade indicators.

Note: For **eNod4** Ethernet models, the network configuration parameters are also reinitiated. This functional command is not available in CANopen® communication protocol.

10.3.9 Zero

When receiving a 'zero' functional command, **eNod4** acquires a volatile zero (gross measurement is set to 0) value if the following conditions are respected:

- measurement is stable
- Current gross measurement is within a $\pm 10\%$ ($\pm 2\%$ if the legal for trade option is enabled) range of the 'maximum capacity'.

Otherwise, after five seconds the command is cancelled and an execution error is reported.

10.3.10 Tare

When receiving a 'tare' functional command, **eNod4** acquires a volatile tare (net measurement is set to 0) value if the measurement is stable otherwise, after five seconds the command is cancelled and an execution error is reported. If the tare acquisition is successful b_{14} bit of the 'measurement status' is set to 1.

10.3.11 Cancel tare

This command erases the current tare value if at least one tare has been previously processed. It also causes b_{14} bit of the *'measurement status'* to be set back to 0.

10.3.12 Cancel last command

This command sets the response register to **00H** and allows **eNod4** to ignore the functional command previously received (for example to exit a sequential procedure like a physical calibration).

10.3.13 Theoretical scaling

The *'theoretical scaling'* functional command involves the *'maximum capacity'* and the *'sensor sensitivity'* (or more specifically of the weighing platform) settings. When used, this command realizes an automatic scaling to migrate from the factory calibration to the user calibration. This calibration must then be saved by sending to **eNod4** the *'store calibration'* functional command. Using the *'zero adjustment'* functional command is also recommended so as to completely adapt **eNod4** to the application.

10.3.14 Zero adjustment

The *'zero adjustment'* functional command allows the user to set his calibration zero value by asking **eNod4** to acquire the current factory calibrated measurement. This acquisition duration depends on the measurement stability; if stability is not reach after 5 seconds, *'zero adjustment'* command is cancelled and an execution error is reported. If it is correctly achieved, this calibration zero modification must then be saved by sending to **eNod4** the *'store calibration'* functional command. This functional command can be used any time and has no effect on the user-span that can have been previously configured through a physical or a theoretical calibration procedure.

10.3.15 Start physical calibration

In order to handle a physical calibration with 1 up to 3 know references, **eNod4** first must be told to enter the physical calibration mode. It is the first step of a sequential procedure.

10.3.16 Calibration zero acquisition

The *'calibration zero acquisition'* is the second step of the physical calibration procedure. It can only be used if the *'start physical calibration'* functional command has been previously received. This acquisition duration depends on the measurement stability; if stability is not reach after 5 seconds, *'calibration zero acquisition'* command is cancelled and an execution error is reported.

Note: In specific cases (silo for example), this step is not mandatory because it is possible to command a "zero adjustment" when the silo is empty.

10.3.17 Segment 1 acquisition

It consists in applying a known reference on the sensor then sending the *'segment 1 acquisition'* functional command. This acquisition duration depends on the measurement stability; if stability is not reach after 10 seconds, *'actual segment acquisition'* command is cancelled and an execution error is reported.

10.3.18 Segment 2/3 acquisition

Only if the *'calibration zero acquisition'* and *"Segment 1 acquisition"* are successful, next step consists in applying a known reference on the sensor then sending the *'segment X acquisition'* functional command where X depends on the value stored in the *'number of calibration segments'* register (see §8). This acquisition duration depends on the measurement stability; if stability is not reach after 10 seconds, *'actual segment acquisition'* command is cancelled and an execution error is reported.

10.3.19 Store calibration (end of physical calibration)

Only if the *'segment 1/2/3 acquisition'* is successful, next step consists in validating the new calibration by storing the zero and the span that have been determined in EEPROM.

Note: This functional command has to be transmitted at the end of a physical calibration, after a *'zero adjustment'*, a *'theoretical scaling'* or a *'zero offset'*.

10.3.20 Logical outputs 1-4 activation/deactivation

If the corresponding logical outputs are assigned to the *'level on request'* function, they can be enabled/disabled by transmitting one of these functional commands. Upon first reception, the corresponding output is enabled and on next reception it will be disabled. If the requesting logical output is assigned to the wrong function, **eNod4** reports an error.

10.3.21 Zero offset

It is also possible to adjust the calibration zero value without acquiring a new one. By entering a positive or negative value into the *'delta zero'* register, the user can quantify the offset (in factory calibrated points) that has to be added or subtracted from the actual calibration zero. This calibration zero modification must then be saved by sending to **eNod4** the *'store calibration'* functional command.

10.3.22 Preset tare

With this command it is possible to retrieve a tare value defined previously.

Important: Preset tare value must be stored in corresponding parameter before to send this command.

10.3.23 Sensor input reference

Sensor input reference command will cause **eNod4** to handle special sequence to acquire **sensor input control reference** value of the load cell sensor input. This command must not be realized when any process cycle that use weight is in progress (because weight variables do not reflect the real weight whilst command is in progress). This command can fail (error in response register) in case of stability timeout on sensor input. The execution time of this command depends on the weight filtering settings. For any further information about this functionality and result variables see § *"Weighing diagnosis"* in the § *"Measurement and status"*.

10.3.24 Sensor input control

Sensor input control command will cause **eNod4** to handle special test on sensor input and to deliver a test result. This command must not be realized when any process cycle that use weight is in progress (because weight variables do not reflect the real weight whilst command is in progress). This command can fail (error in response register) in case of stability timeout on sensor input. The execution time of this command depends on the weight filtering settings. For any further information about this functionality and result variables see *"Weighing diagnosis"* § in the *Measurement and status* §.

11 CALIBRATION SETTINGS AND PROCEDURES

Name	Modbus address	Acyclic DPV1 slot/index	Profibus cyclic IN/OUT	CANopen® index/sub-index	Type	Access
Maximum capacity	0x000C	0x02 / 0x00	R:0x0420 W:0x0421	0x3002 / 0x00 (M)	Ulong	RW
Number of calibration segments	0x000E	0x02 / 0x01	R:0x0222 W:0x0223	0x3000 / 0x00	Uint	RW
Calibration load 1	0x000F	0x02 / 0x02	R:0x0424 W:0x0425	0x3001 / 0x01 (M)	Ulong	RW
Calibration load 2	0x0011	0x02 / 0x03	R:0x0426 W:0x0427	0x3001 / 0x02	Ulong	RW
Calibration load 3	0x0013	0x02 / 0x04	R:0x0428 W:0x0429	0x3001 / 0x03	Ulong	RW
Sensor sensitivity	0x0015	0x02 / 0x05	R:0x042A W:0x042B	0x3004 / 0x00 (M)	Ulong	RW
Scale interval	0x0017	0x02 / 0x06	R:0x022C W:0x022D	0x3003 / 0x00	Uint	RW
Zero calibration	0x0018	0x03 / 0x04	R:0x0434 W:0x0435	0x3006 / 0x00	Long	RW
Span coefficient 1	0x001A	0x03 / 0x05	R:0x0436 W:0x0437	0x3005 / 0x04	Float	RW
Span coefficient 2	0x001C	0x03 / 0x06	R:0x0438 W:0x0439	0x3005 / 0x05	Float	RW
Span coefficient 3	0x001E	0x03 / 0x07	R:0x043A W:0x043B	0x3005 / 0x06	Float	RW
Span adjusting coefficient	0x0020	0x03 / 0x00	R:0x042E W:0x042F	0x3005 / 0x01 (M)	Ulong	RW
Calibration place g value	0x0022	0x03 / 0x01	R:0x0430 W:0x0431	0x3005 / 0x02	Ulong	RW
Place of use g value	0x0024	0x03 / 0x02	R:0x0432 W:0x0433	0x3005 / 0x03	Ulong	RW
Zero offset	0x0092	0x03 / 0x03	R:0x0472 W:0x0473	0x2500 / 0x00 (M)	Long	RW

11.1 Principles

eNod4 analog channel is factory configured to deliver :

- **500 000 pts for 2 mV/V** on the Wheatstone bridge input

The measurement scaling in **eNod4** can be adapted to his application by the user. Some settings and the 2 calibration methods allow the user to define his specific span according to his sensors characteristics.



When using eNod4 for legal for trade purpose, it is imperatively required to activate the legal for trade switch BEFORE any calibration procedure (cf § legal for trade switch).

11.2 Calibration methods

Measurement scaling can be defined using one of the two following methods:

- **Theoretical calibration** involving the sensitivity of the sensor and a user-defined corresponding capacity

- **Physical calibration** involving 1, 2 or 3 know loads (for a load cell) or 1,2 or 3 measured voltages (for the 0-10 V analog channel)

Both can be achieved thanks to the functional commands.

11.3 Settings description

11.3.1 Maximum capacity

The '*maximum capacity*' stands for the maximum sensor/load cell signal range. When the absolute value of the gross measurement exceeds its value plus 9 divisions, the b_3 bit (positive overloading) or the b_2 bit (negative overloading) of the measurement status is set to 1 (it can activate a logical output if it is assigned to the '*defective measurement*' function).

The zero acquisition (on request or at power-up) is done only if the gross measurement value is contained between a $\pm 10\%$ range of the '*maximum capacity*' ($\pm 2\%$ if the *legal for trade* option is active).

The '*maximum capacity*' setting also allows calibrating **eNod4** in case of a theoretical calibration in association with the sensor sensitivity. Measurement scaling will be automatically adapted so as to deliver a gross measurement value equivalent to the '*maximum capacity*' for an analog signal corresponding to the sensor sensitivity.

After a theoretical calibration, the maximum capacity can be changed to fit to the application.

Admitted values : from 1 up to 10000000.

11.3.2 Number of calibration segments

The '*number of calibration segments*' defines how many calibration segments are used during the physical calibration procedure. Linear installations only need one segment.

Admitted values : from 1 up to 3.

11.3.3 Calibration loads 1/2/3

Before starting a physical calibration procedure, each calibration segment must be given a corresponding user value (for example, 1000 points for a 1 kg load).

Admitted values : from 1 up to 10000000.

11.3.4 Sensor sensitivity

The '*sensor sensitivity*' setting is used to achieve a theoretical calibration. The stored value for this parameter can be:

- the **load cell sensitivity in mV/V** for the low-level analog channel
- an **input signal voltage in V** for the analog 0-10V analog channel

The user can adapt the value delivered by **eNod4** for the associated signal using the '*maximum capacity*' and the '*sensor sensitivity*'.

This setting is expressed with a 10^{-5} factor (197500 is equivalent to a 1.975 mV/V load cell sensitivity or a 1.975 V input voltage).

Admitted values : from 1 up to 1000000.

11.3.5 Scale interval

The '*scale interval*' is the minimal difference between two consecutive indicated values (either gross or net).

Admitted values : 1/2/5/10/20/50/100

11.3.6 Zero calibration

Zero calibration value corresponds to the A/D converter points measured during the 'zero acquisition' step of a physical calibration.

For a theoretical calibration this value must be set. It can be set automatically with the 'zero adjustment' command.

Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).

Admitted values : from 0 up to +-10000000

11.3.7 Span coefficients 1/2/3

These coefficients are computed and written during calibration process. Writing these coefficients could be done if you want to restore a previous calibration.

Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).

Admitted values : different from 0.

11.3.8 Span adjusting coefficient

The 'span adjusting coefficient' allows adjusting initial calibration. Adjustment applies linearly on the whole calibration curve. This coefficient has a 10^{-6} factor (1000000 is equivalent to a span adjusting coefficient that is equal to 1).

Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).

Admitted values : from 900000 up to 1100000.

11.3.9 Calibration place g value / place of use g value

When the calibration place and the place of use of a measuring chain are different, a deviation can appear due to the difference of g (gravity) between the 2 places.

The eNod4 calculates a ratio applied to the measure which compensates the difference of gravity between the 2 places.

The g value are expressed in 10^{-6} m.s^{-2} (9805470 is equivalent to $g = 9.805470 \text{ m.s}^{-2}$).

The **eNodView** software can help to determine the g value of a place.

Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).

Admitted values : different from 0.

11.3.10 Zero offset

The 'Zero offset' value contains the offset in factory calibrated points that can be added/subtracted (if its value is positive or negative) to the zero calibration value when using the 'zero offset' functional command. Once the command has been successfully achieved, this register is set to 0.

Note: The 'Zero offset' value is not stored into EEPROM memory and is always equal to 0 after a device power-up or a software reset

Admitted values : different from 0.

12 FILTERS

Name	Modbus address	Acyclic DPV1 slot/index	Profibus cyclic IN/OUT	CANopen® index/sub-index	Type	Access
A/D conversion rate	0x0036	0x04 / 0x00	R:0x0240 W:0x0241	0x4000 / 0x00	Uint	RW
filters activation	0x0037 LSB	0x04 / 0x01 LSB	R:0x0242 LSB W:0x0243 LSB	0x4001 / 0x01 (byte)	Byte	RW
Low-pass order	0x0037 MSB	0x04 / 0x01 MSB	R:0x0242 MSB W:0x0243 MSB	0x4001 / 0x02 (byte)	Byte	RW
Low-pass cut-off frequency	0x0038	0x04 / 0x02	R:0x0244 W:0x0245	0x4001 / 0x03	Uint	RW
Band-stop high cut-off frequency	0x0039	0x04 / 0x03	R:0x0246 W:0x0247	0x4001 / 0x04	Uint	RW
Band-stop low cut-off frequency	0x003A	0x04 / 0x04	R:0x0248 W:0x0249	0x4001 / 0x05	Uint	RW

12.1 Principles

eNod4 contains 4 filtering layers that are user-configurable :

- filtering related to the A/D conversion rate (with rejection of the mains frequency)
- a low-pass Bessel-type filter
- a band-stop filter
- a self-adaptive filter

Except for the A/D conversion rate that is always enabled, none of these filters is mandatory. However, to perform accurate measurements we recommend setting a combination of filters. **eNodView** software may be helpful in designing the best filter configuration for the application.

12.2 Settings list

Here is the list of the settings that have an impact on the filters configuration:

12.3 Settings description

12.3.1 A/D conversion rate

It contains a code which represents the A/D conversion rate and the rejection. See table below:

b_4	Rejection	
0	60 Hz	
1	50 Hz	

$b_3 b_2 b_1 b_0$	A/D conversion rate (measures/s)	
	50-Hz rejection	60-Hz rejection
0000	100	120
0001	50	60
0010	25	30
0011	12.5	15
0100	6.25	7.5
1001	1600	1920
1010	800	960
1011	400	480
1100	200	240

Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).

12.3.2 Filters activation & order

This setting allows to define what filters are enabled in **eNod4** signal processing chain.

Note : the filters activation & order setting can be accessed through a 16-bits register except in CANopen® communication protocol where this word is divided into 2 8-bits registers :

b_0	Meaning
0	band-stop filter disabled
1	band-stop filter enabled

b_1	Meaning
0	self-adaptive filter disabled
1	self-adaptive filter enabled

$b_{10} b_9 b_8$	Meaning
000	low-pass filter disabled
010	2 nd order low-pass filter
011	3 rd order low-pass filter
100	4 th order low-pass filter

Note: In CANopen® communication protocol (according to version), this word is divided into 2 bytes of 8-bits registers. Bits b8 to b15 are therefore equivalent to bits b0 to b7 of the corresponding address (see CANopen® Register table).

12.3.3 Low-pass filter cut-off frequency

This register contains the low-pass filter cut-off frequency expressed in Hz and multiplied by 100. That means that 690 is equivalent to 6.90 Hz. The value must be compliant with the table shown in §12.4.

Admitted values : from 10 up to 20000.

12.3.4 Band-stop filter high cut-off frequency

This register contains the band-stop filter high cut-off frequency expressed in Hz and multiplied by 100. That means that 690 is equivalent to 6.90 Hz. **The value must be higher than the band-stop filter low cut-off frequency.**

Admitted values : from 10 up to 20000.

12.3.5 Band-stop filter low cut-off frequency

This register contains the band-stop filter low cut-off frequency expressed in Hz and multiplied by 100. That means that 690 is equivalent to 6.90 Hz. **The value must be lower than the band-stop filter high cut-off frequency.**

Admitted values : from 10 up to 20000.

12.4 Limitations

Recursive filters like **eNod4** low-pass filters are computed according to the filter order, the desired cut-off frequency and the sampling rate. There are some limitations to respect in order to ensure a safe functioning of the signal processing. They are listed in the table below :

A/D conversion rate (meas/s)	min low-pass cut-off frequency (Hz)			A/D conversion rate (meas/s)	min low-pass cut-off frequency (Hz)		
	50 Hz rejection				60 Hz rejection		
	2nd order	3rd order	4th order		2nd order	3rd order	4th order
6.25	0.10	0.10	0.10	7.5	0.10	0.10	0.15
12.5	0.10	0.10	0.15	15	0.10	0.15	0.20
25	0.10	0.15	0.25	30	0.15	0.20	0.30
50	0.15	0.25	0.50	60	0.20	0.30	0.60
100	0.25	0.50	1.00	120	0.30	0.60	1.20
200	0.50	1.00	2.00	240	0.60	1.20	2.40
400	1.00	2.00	4.00	480	1.20	2.40	4.80
800	2.00	4.00	8.00	960	2.40	4.80	9.60
1600	4.00	8.00	16.00	1920	4.80	9.60	19.20

13 CONFIGURATION OF INPUT/OUTPUT

Name	Modbus address	Acyclic DPV1 slot/index	Profibus cyclic IN/OUT	CANopen® index/sub-index	Type	Access
Logical input 1 functioning	0x0042 LSB	0x05 / 0x00 LSB	R:0x0250 LSB W:0x0251 LSB	0x4501 / 0x02	Byte	RW
Logical input 2 functioning	0x0042 MSB	0x05 / 0x00 MSB	R:0x0250 MSB W:0x0251 MSB	0x4501 / 0x03	Byte	RW
Logical input 3 functioning (IO+ version)	0x0041 LSB	0x05 / 0x02 LSB	R:0x026A LSB W: 0x026B LSB	0x4501 / 0x04	Byte	RW
Logical input 4 functioning (IO+ version)	0x0041 MSB	0x05 / 0x02 MSB	R:0x026A MSB W: 0x026B MSB	0x4501 / 0x05	Byte	RW
holding time	0x0043	0x05 / 0x01	R:0x0252 W:0x0253	0x4501 / 0x01	Uint	RW
Analog output functioning (IO+ version)	0x0040	0x06 / 0x0B	R:0x026C W:0x026D	0x4509 / 0x05	Uint	RW
External value to control analog output (IO+ version)	0x0032	0x06 / 0x0C	R:0x023C W:0x023D + See modules list	0x5050 / 0x00 (M)	Uint	RW
Output 1 functioning	0x0044 LSB	0x06 / 0x00 LSB	R:0x0254 LSB W:0x0255 LSB	0x4509 / 0x01	Byte	RW
Output 2 functioning	0x0044 MSB	0x06 / 0x00 MSB	R:0x0254 MSB W:0x0255 MSB	0x4509 / 0x02	Byte	RW
Output 3 functioning	0x0045 LSB	0x06 / 0x01 LSB	R:0x0256 LSB W:0x0257 LSB	0x4509 / 0x03	Byte	RW
Output 4 functioning	0x0045 MSB	0x06 / 0x01 MSB	R:0x0256 MSB W:0x0257 MSB	0x4509 / 0x04	Byte	RW
Set point 1 high value	0x0046	0x06 / 0x02	R:0x045A W:0x045B	0x4601 / 0x02	Long	RW
Set point 1 low value	0x0048	0x06 / 0x03	R:0x045C W:0x045D	0x4601 / 0x03	Long	RW
Set point 2 high value	0x004A	0x06 / 0x04	R:0x045E W:0x045F	0x4601 / 0x04	Long	RW
Set point 2 low value	0x004C	0x06 / 0x05	R:0x0460 W:0x0461	0x4601 / 0x05	Long	RW
Set point 3 high value	0x004E	0x06 / 0x06	R:0x0462 W:0x0463	0x4609 / 0x02	Long	RW
Set point 3 low value	0x0050	0x06 / 0x07	R:0x0464 W:0x0465	0x4609 / 0x03	Long	RW
Set point 4 high value	0x0052	0x06 / 0x08	R:0x0466 W:0x0467	0x4609 / 0x04	Long	RW
Set point 4 low value	0x0054	0x06 / 0x09	R:0x0468 W:0x0469	0x4609 / 0x05	Long	RW
1&2 Set points functioning	0x0056 LSB	0x06 / 0x0A LSB	R:0x0258 LSB W:0x0259 LSB	0x4601 / 0x01	Byte	RW
3&4 Set points functioning	0x0056 MSB	0x06 / 0x0A MSB	R:0x0258 MSB W:0x0259 MSB	0x4609 / 0x01	Byte	RW
Input level	0x0094 LSB	/	See modules list	0x5100 / 0x00 (M)	Byte	RO
Output level	0x0094 MSB	/	See modules list	0x5200 / 0x00 (M)	Byte	RO

13.1 Principles

eNod4 device is fitted with 2 logical inputs (4 logical inputs for IO+ version) and 4 logical outputs that are fully configurable.

13.1.1 Logical inputs

Each input can work individually in either positive or negative logic. A holding time (de-bounced time) attached to all inputs can be configured. The available functions are:

- **None:** the input has no function
- **Tare:** a rising (positive logic) or a falling edge (negative logic) causes a tare function to be triggered.
- **Zero:** a rising (positive logic) or a falling edge (negative logic) causes a zero function to be triggered.
- **Cancel tare:** a rising (positive logic) or a falling edge (negative logic) causes the current stored tare to be erased.
- **Transmit measurement:** (depend on communication protocol). A rising (positive logic) or a falling edge (negative logic) triggers a measurement transmission.
- **Measurement window:** only available in SCMBus/fast SCMBus protocols. Measurements are continuously transmitted at a rate defined by the '*sampling period*' while the input is maintained at the defined level.
- **Sensor input control:** a rising (positive logic) or a falling edge (negative logic) triggers a test routine of the sensor input and produces a test result.

13.1.2 Analog output (IO+ version)

An optional analog board in *current* and *voltage* might be used with **eNod4** to provide IO+ version. This must be asked when ordering **eNod4** product.

Voltage output might be set either 0-5V or 0-10V, and the current output to 4-20mA, 0-24mA, 0-20mA or 4-20mA with alarm at 3.6mA. Both output (current and voltage) might separately be enable. Settings are effective after **eNod4** reset.

Analog output affectation function is common to both *current* and *voltage* output and might be assigned to followings:

- **None :** analog outputs have no function.
- **Gross measurement :** analog outputs can be assigned to gross measurement copy. Maximal level value is related to **Maximum Capacity** parameter and works in mono-quadrant functioning. Bi-quadrant option can only be applied to gross measurement copy. When this option is activated, the lowest value of current and voltage levels corresponds to **-MC** and the highest value to **+MC**.
- **Net measurement :** analog outputs can be assigned to net measurement copy. Maximal value is related to **Maximum Capacity** parameter and works in mono-quadrant functioning only. The highest value of current and voltage levels corresponds to **+MC** in only one quadrant.
- **Level on request :** analog outputs are driven by master requests through the **external value to control analog output** variable (in 0.01% of the full scale of current or voltage analog outputs).

When analog output is assigned to "Gross measurement" or "Net measurement" its value jumps to a special error value when the internal alarm flag described in "Weighing diagnosis" § in the *Measurement and status* § is activated. This allows to warn about defection of the measurement chain.

The error value on analog output is defined depending on voltage or current settings as described in following table:

<i>Setting</i>	<i>Analog output error mode value</i>
0 - 5V	5.5 V
0 - 10V	11 V
4 - 20mA	no output current
0 - 20mA*	no output current
0 - 24mA*	no output current
4 mA - 20 mA with alarm at 3.6 mA	3.6 mA, voltage output is deactivated (High-Z state)

* no error detection possible in this setting

13.1.3 Logical outputs

The available functions are:

- **None:** the output has no function assigned.
- **Motion:** the output is assigned to copying the stability flag level.
- **Defective measurement:** the output level is set when the internal alarm flag described in “Weighing diagnosis” § in the *Measurement and status* § is activated. This allows to warn about defection of the measurement chain.
- **Set point :** each output can be assigned to a configurable set point (set point 1 corresponds to output 1, set point 2 to output 2, set point 3 to output 3 and set point 4 to output 4) .
- **Input image:** the output is at the same level as the logical input level (outputs 1 and 3 correspond to input 1, outputs 2 and 4 correspond to input 2).
- **Level on request:** the input level is set on master requests.

13.2 Settings description

13.2.1 Logical inputs assignment

The following tables describe the possible assignments.

Bits	Meaning	Note
$b_3 b_2 b_1 b_0$	input 1 assignment (and input 3 if IO+ version) assignment	
0000	none	the input has no function
0001	tare	
0010	zero	
0011	cancel tare	
0100	transmit measurement (note 1)	data is transmitted on the bus at every rising or falling edge (depending on the chosen logical)
0101	measurement window (note 1)	data is transmitted on the bus while the input is maintained at the right level (depending on the chosen logical). Transmission rate is fixed by the 'sampling rate' setting
0110	Sensor input control	Same like equivalent functional command describe in § functional commands
b_4	input 1&3 logical	
0	negative logic	defines the edge (or level) that triggers input 1 function
1	positive logic	
$b_6 b_5$	measurement to be transmitted (note 1)	
00	gross	
01	net	
10	factory calibrated points	

Bits	Meaning	Note
$b_{11} b_{10} b_9 b_8$ (note 2)	input 2 assignment (and input 4 if IO+ version) assignment	
0000	none	the input has no function
0001	tare	
0010	zero	
0011	cancel tare	
0100	transmit measurement (note 1)	data is transmitted on the bus at every rising or falling edge (depending on the chosen logical)

<i>Bits</i>	<i>Meaning</i>	<i>Note</i>
0101	<i>measurement window (note 1)</i>	<i>data is transmitted on the bus while the input is maintained at the right level (depending on the chosen logical). Transmission rate is fixed by the 'sampling rate' setting</i>
0110	<i>Sensor input control</i>	<i>Same like equivalent functional command describe in § functional commands</i>
<i>b₁₂</i> (note 2)	<i>input 2&4 logical</i>	
0	<i>negative logic</i>	<i>defines the edge (or level) that triggers input 1 function</i>
1	<i>positive logic</i>	
<i>b₁₄ b₁₃</i> (note 2)	<i>measurement to be transmitted (note 1)</i>	
00	<i>gross</i>	
01	<i>net</i>	
10	<i>factory calibrated points</i>	

Note 1: Only for SCMBus and fast SCMBus protocols communication, no effect otherwise.

Note 2: In CANopen® communication protocol (according to version), this word is divided into 2 bytes of 8-bits registers. Bits b8 to b15 are therefore equivalent to bits b0 to b7 of the corresponding address (see CANopen® Register table).

13.2.2 Holding time (debounced time)

The holding time (de-bounced time) corresponds to the minimum required stabilization time of the logical inputs before their activation. If the input level varies within this interval, the assigned command is ignored.

13.2.3 Analog output(s) assignment (IO+ version)

The following tables describe the possible assignments.

<i>bits</i>	<i>meaning</i>	<i>note</i>
<i>b3 b2 b1 b0</i>	<i>analog output(s) assignment</i>	
<i>0000</i>	<i>none</i>	<i>the output level does not vary</i>
<i>0001</i>	<i>copy gross weight</i>	<i>Adjustable polarity</i>
<i>0010</i>	<i>copy net weight</i>	
<i>0011</i>	<i>level on request</i>	<i>parameter External value to control analog output will drive analog output</i>
<i>b4</i>	<i>polarity</i>	
<i>0</i>	<i>unipolar</i>	<i>could be set only with gross measurement</i>
<i>1</i>	<i>bipolar</i>	
<i>b7 b6 b5</i>	<i>output voltage settings</i>	
<i>000</i>	<i>disable</i>	
<i>001</i>	<i>0 V - 5 V</i>	
<i>010</i>	<i>0 V - 10 V</i>	
<i>b10 b9 b8</i>	<i>output current settings</i>	
<i>000</i>	<i>disable</i>	
<i>001</i>	<i>4 mA - 20 mA</i>	
<i>010</i>	<i>0 mA - 20 mA</i>	
<i>011</i>	<i>0 mA - 24 mA</i>	
<i>100</i>	<i>4 mA - 20 mA with alarm at 3.6 mA</i>	<i>voltage output is inactive (High-Z state)</i>

13.2.4 External value to control analog output (IO+ version)

For use with *level on request* function. In this configuration *eNod4* will copy *external value to control analog output* parameter on analog output in current and voltage.

The external value parameter is expressed in 0.01% of full scale of analog output current or voltage.

13.2.5 Logical outputs 1&2 assignment

The following table describes the possible assignments.

<i>Bits</i>	<i>Meaning</i>	<i>Note</i>
<i>b3 b2 b1 b0</i>	<i>output 1 assignment</i>	
<i>0000</i>	<i>none</i>	<i>the output level does not vary</i>
<i>0001</i>	<i>set point 1</i>	<i>functioning described by the 'set point functioning' setting and by the 'set point 1 high and low values'</i>
<i>0010</i>	<i>motion</i>	<i>copies the motion flag of the status bytes</i>
<i>0011</i>	<i>defective measurement</i>	<i>reflect the internal alarm flag described in "Weighing diagnosis" § in the Measurement and status §</i>
<i>0100</i>	<i>input 1 image</i>	<i>copies input 1 level</i>
<i>0101</i>	<i>level on request</i>	<i>output 1 level is driven by the 'OUT1 activation/deactivation' functional command</i>
<i>b4</i>	<i>output 1 logical</i>	
<i>0</i>	<i>negative logic</i>	<i>defines the output level when enabled</i>
<i>1</i>	<i>positive logic</i>	
<i>b11 b10 b9 b8(note 1)</i>	<i>output 2 assignment</i>	
<i>0000</i>	<i>none</i>	<i>the output level does not vary</i>
<i>0001</i>	<i>set point 2</i>	<i>functioning described by the 'set point functioning' setting and by the 'set point 2 high and low values'</i>
<i>0010</i>	<i>motion</i>	<i>copies the motion flag of the status bytes</i>
<i>0011</i>	<i>defective measurement</i>	<i>reflect the internal alarm flag described in "Weighing diagnosis" § in the Measurement and status §</i>
<i>0100</i>	<i>input 2 image</i>	<i>copies input 2 level</i>
<i>0101</i>	<i>level on request</i>	<i>output 2 level is driven by the 'OUT2 activation/deactivation' functional command</i>
<i>b12 (note 1)</i>	<i>output 2 logical</i>	
<i>0</i>	<i>negative logic</i>	<i>defines the output level when enabled</i>
<i>1</i>	<i>positive logic</i>	<i>defines the output level when enabled</i>

Note 1: In CANopen® communication protocol (according to version), this word is divided into 2 bytes of 8-bits registers. Bits b8 to b15 are therefore equivalent to bits b0 to b7 of the corresponding address (see CANopen® Register table).

13.2.6 Logical outputs 3&4 assignment

The assignment is similar to the outputs 1&2 configuration parameter, see previous paragraph (replacing all references to output 1 with output 3 and all references to output 2 with output 4).

13.2.7 Set points functioning

The following table describes the possible assignments.

<i>Bits</i>	<i>Meaning</i>	<i>Note</i>
b0	set point 1 commutation mode	
0	<i>window</i>	<i>only if output 1 assigned to the 'set point' function</i>
1	<i>hysteresis</i>	
b2 b1	set point 1 comparison measurement	
00	<i>gross</i>	
01	<i>net</i>	
10	<i>Sensor input control result</i>	
b3	reserved (0)	
b4	set point 2 commutation mode	
0	<i>window</i>	<i>only if output 2 assigned to the 'set point' function</i>
1	<i>hysteresis</i>	
b6 b5	set point 2 comparison measurement	
00	<i>gross</i>	
01	<i>net</i>	
10	<i>Sensor input control result</i>	
b7	reserved (0)	
b8 (note 1)	set point 3 commutation mode	
0	<i>window</i>	<i>only if output 3 assigned to the 'set point' function</i>
1	<i>hysteresis</i>	
b10 b9 (note 1)	set point 3 comparison measurement	
00	<i>gross</i>	
01	<i>net</i>	
10	<i>Sensor input control result</i>	
b11 (note 1)	reserved (0)	
b12 (note 1)	set point 4 commutation mode	

<i>Bits</i>	<i>Meaning</i>	<i>Note</i>
0	<i>window</i>	<i>only if output 4 assigned to the 'set point' function</i>
1	<i>hysteresis</i>	
b14 b13 (note 1)	<i>set point 4 comparison measurement</i>	
00	<i>gross</i>	
01	<i>net</i>	
10	<i>Sensor input control result</i>	
b15 (note 1)	<i>reserved (0)</i>	

Note 1: In CANopen® communication protocol (according to version), this word is divided into 2 bytes of 8-bits registers. Bits b8 to b15 are therefore equivalent to bits b0 to b7 of the corresponding address (see CANopen® Register table).

13.2.8 Set points high and low values

Each set point can be configured by its commutation mode (hysteresis/window) and by a couple of values that are continuously compared to the gross or net measurement (depending on the configuration of the set point) in order to define the corresponding output logical level.

For more details about the set points functioning, see the *user manual, characteristics and functioning*.

Admitted values: from -1000000 to 1000000.

13.3 Input/output level

The level of the eNod4 Input/output can be read according to the following table:

<i>Bits</i>	<i>Meaning</i>	<i>Note</i>
<i>b0</i>		
<i>0</i>	<i>low</i>	<i>IN1 level</i>
<i>1</i>	<i>high</i>	
<i>b1</i>		
<i>0</i>	<i>low</i>	<i>IN2 level</i>
<i>1</i>	<i>high</i>	
<i>b2</i>	<i>With IO+ version only, else 0</i>	
<i>0</i>	<i>low</i>	<i>IN3 level</i>
<i>1</i>	<i>high</i>	
<i>b3</i>	<i>With IO+ version only, else 0</i>	
<i>0</i>	<i>low</i>	<i>IN4 level</i>
<i>1</i>	<i>high</i>	
<i>b7 ... b4</i>		
<i>0</i>	<i>reserved (0)</i>	
<i>b8 (note 1)</i>		
<i>0</i>	<i>low</i>	<i>OUT1 level</i>
<i>1</i>	<i>high</i>	
<i>b9 (note 1)</i>		
<i>0</i>	<i>low</i>	<i>OUT2 level</i>
<i>1</i>	<i>high</i>	
<i>b10 (note 1)</i>		
<i>0</i>	<i>low</i>	<i>OUT3 level</i>
<i>1</i>	<i>high</i>	
<i>b11 (note 1)</i>		
<i>0</i>	<i>low</i>	<i>OUT4 level</i>
<i>1</i>	<i>high</i>	
<i>b15 ... b12 (note 1)</i>		
<i>0</i>	<i>reserved (0)</i>	

Note 1: In CANopen® communication protocol (according to version), this word is divided into 2 bytes of 8-bits registers. Bits b8 to b15 are therefore equivalent to bits b0 to b7 of the corresponding address (see CANopen® Register table).

14 LEGAL FOR TRADE OPTIONS

<i>Name</i>	<i>Modbus address</i>	<i>Acyclic DPV1 slot/index</i>	<i>Profibus cyclic IN/OUT</i>	<i>CANopen® index/sub-index</i>	<i>Type</i>	<i>Access</i>
Legal for trade version	0x0004 LSB	0x01 / 0x00 LSB	R: 0x0210 LSB W: /	0x3600 / 0x02	Byte	RO
Legal for trade switch	0x0004 MSB	0x01 / 0x00 MSB	R: 0x0210 MSB W: 0x0211MSB	0x3600 / 0x01	Byte	RW
Legal for trade counter	0x0005	0x01 / 0x01	R: 0x0212 W: /	0x3600 / 0x03	Byte	RO
Legal for trade checksum	0x0006	0x01 / 0x01	R: 0x0214 W: /	0x3600 / 0x04	Uint	RO
Options	0x0A50	0x0B / 0x01	R: / W: /	0x3600 / 0x05	Uint	RW
Zero functions	0x0007	0x01 / 0x02	R:0x0216 W:0x0217	0x3500 / 0x00	Uint	RW
Stability criterion	0x0008 LSB	0x01 / 0x03 LSB	R:0x0218 LSB W:0x0219 LSB	0x3605 / 0x00	Byte	RW
decimal point position	0x0008 MSB	0x01 / 0x03 MSB	R:0x0218 MSB W:0x0219 MSB	0x3700 / 0x02	Byte	RW
Unit	0x0009	0x01 / 0x04	R:0x041A W:0x041B	0x3700 / 0x01	String	RW
Oldest DSD record ID	0x0028	0x09 / 0x00	R: / W: /	0x3800 / 0x01	Ulong	RO
DSD current record ID to read	0x0A60	0x09 / 0x02	R: / W: /	0x3800 / 0x03	Ulong	RW
DSD latest record ID	0x0A8E	0x09 / 0x01	R: / W: /	0x3800 / 0x02	Ulong	RO
Read record ID	0x0A90	0x09 / 0x03	R: / W: /	0x3800 / 0x04	Ulong	RO
Net weight in read record	0x0A92	0x09 / 0x04	R: / W: /	0x3800 / 0x05	Ulong	RO
Tare value in read record	0x0A94	0x09 / 0x05	R: / W: /	0x3800 / 0x06	Ulong	RO
Weighing status in read record	0x0A96	0x09 / 0x06	R: / W: /	0x3800 / 0x07	Uint	RO
Record checksum of read record	0x0A97	0x09 / 0x07	R: / W: /	0x3800 / 0x08	Uint	RO

14.1 Principles

eNod4 is an analog data processing unit evaluated as a part of a non-automatic weighing instrument (NAWI) or an automatic weighing instrument (AWI) like an automatic gravimetric filling instrument or a catchweigher. This instrument is not intended for direct sales to the public. It is suitable for conditioning OIML R60 certified strain gauges load cell(s) with analog output.

For those weighing applications in a regulated metrological field, some specific parameters are required and also some functions are provided with a restricted action compared to their functioning in non-regulated use.



When using eNod4 for legal for trade purpose, it is imperatively required to activate the legal for trade switch BEFORE any calibration procedure (cf § legal for trade switch).

Legal for trade mode has to be activated internally in order to respect metrological requirements. All the functionalities will be conformed to the essential requirements for certified weighing instruments.

Important remark

Switch in legal for trade mode is not accepted if the following conditions are not fulfilled:

- The weight unit has to be one of the following values → mg, g, kg, t, ct, µg, ozt,
- The stability criterion has to be 0.25d,
- The *Measuring range / Division* ratio has to be higher or equal to 100 and lower or equal to 6000,
- The filters set-up time has to be lower than 1 second in transmitter mode,
- *Division* value must be lower than 100. If higher or equal to 10 and lower or equal to 50, the decimal point value can only be 0 or 3.

14.2 Settings description

14.2.1 Legal for trade switch

This setting activates (**b₀ bit set to 1**) or deactivates (**b₀ bit set to 0**) criteria and parameters related to the use of **eNod4** in OIML compliance. This setting has to be activated imperatively previously any other (especially the calibration step of the instrument).

The '*legal for trade*' option activation leads to the following changes:

- Change in the precision and internal computation of the weight result (that requires to re-calibrate the instrument after activation/ deactivation of '*legal for trade*' option),
- the '*legal for trade counter*' is incremented every time a storage into nonvolatile memory is requested if one or several metrological settings have been modified.
- a new '*legal for trade checksum*' value is calculated every time a storage into nonvolatile memory is requested if one or several metrological settings have been modified (parameters list see § below),
- taring is now impossible if gross measurement is negative and higher than the measuring range,
- no more weight display if a weight result exceeds the measuring range + 9 divisions,
- the measurement weight values cannot be read during the 2 seconds that follow the device reset (error frame in Modbus RTU, value set to -1 in CANopen® and in Profibus DP) and during zero and tare acquisitions. A flag is set in the measurement status in order to indicate to an associated HMI that the delivered weight value can be displayed or not.
- Some limitations apply to the filter response time.

14.2.2 Legal for trade sealing

eNod4 is fitted with a software sealing for metrological parameters and once activated, is composed of an event counter and a CRC value of specific and adjustment parameters.

Legal for trade sealing is a read only variable of the **eNod4 (b₁ bit set to 1 of the MSB part of Legal for trade switch and version** variable). Activation / deactivation (commutation) of the legal for trade sealing has to be done through a specific command (*Switch legal sealing*). Legal for trade sealing is generally performed when putting in service the instrument.

When sending this command, the legal for trade counter is incremented, a new determination of the legal for trade checksum is done and their new values stored in the non-volatile memory with also the whole configuration of eNod4 (like a command *EEPROM storage*).

These values shall be marked on the terminal device connected after the official verification when putting in service the instrument. Any mismatch between the displayed values on the terminal and those marked will signify a broken sealing.

Note: sealing will be material on the eNod4 housing and load cell connections. If a junction box is used it will have to be sealed too.

When the legal for trade sealing is activated, the following parameters cannot be written or modified any more:

- Serial number
- Legal for trade mode activating (and legal for trade version)
- Weight unit
- Initial zero setting device activation / deactivation
- Zero setting device activation / deactivation
- Decimal point position
- Stability criterion
- Measuring range
- Number of calibration segments
- Division
- Calibration zero
- Span coefficient 1
- Span coefficient 2
- Span coefficient 3
- Span adjusting coefficient
- Calibration place g value
- Place of use g value

And also to the following parameters that are also subject to some limitations:

- A/D conversion rate
- Mains frequency rejection
- Low pass order activation / deactivation
- Low pass cut-off frequency
- Band-stop activation / deactivation
- High cut-off frequency band-stop filter
- Low cut-off frequency band-stop filter
- Self-adaptive filter activation / deactivation

It will be no more possible to execute the following commands (setting in error of the response register):

- Zero adjustment
- Theoretical scaling
- Start physical calibration
- Calibration segment 1 acquisition
- Zero offset correction

14.2.3 Legal for trade software version

This RO value identifies the version of the legally-relevant software part that is dedicated to the metrology and the measurement exploitation.

14.2.4 Legal for trade counter

If the 'legal for trade switch' is enabled, the legal for trade counter is incremented every time a backup into EEPROM is requested if at least one (or several) of the settings described above at § *legal for trade sealing* has been modified.

14.2.5 Legal for trade checksum

If the 'legal for trade switch' is enabled, a new legal for trade checksum is calculated every time a backup into EEPROM is requested if at least one (or several) of the settings listed above at § *legal for trade sealing* has been modified.

14.2.6 Alibi memory or DSD (Data Storage Device)

eNod4 is fitted with a data storage device (DSD) so called alibi memory. Any weighing result is stored internally and can be recalled on demand. 130816 records can be stored permanently at maximum. Any record is identified by a unique 32-bit long number. This identifier is incremented each time a weighing result is stored and transmitted. The minimum time between two DSD recording operations is 50ms.

You will find in the variable *DSD latest record ID*, the ID of the oldest record memorized.

DSD record structure:

Size (bytes)	Type	eNod4-T DSD record
4	Ulong	Read record ID
4	long	Net weight in read record
4	long	Tare value of read record
2	Uint	Weighing status in read record
2	Uint	Record checksum of read record

Weighing status in read record:

bit	Weighing status in read record
0	gross meas.< (- max capacity) OR gross meas. > (max capacity)
1	no motion
2	zero in the ¼ of division
3, 4, 5	Tare status (000: no tare, 001: semiautomatic tare, 010: preset tare, 011-111: reserved)
6, 7, 8, 9	Decimal point position (from 0 up to 7)
10, 11, 12	Unit (000: µg, 001: mg, 010:g, 011:kg, 100:t, 101:ct, 110: ozt, 111: other unit out of legal for trade)
13, 14, 15	Reserved

14.2.7 Zero functions

The zero tracking and the initial zero setting can be respectively enabled by setting b₀ bit or b₁ bit to 1. When activated, both options are effective on a ±10% range of the 'maximum capacity' (±2% if the 'legal for trade' option is enabled).



When the initial zero is used, you must use a stability criterion other than 0 to be not affected by transient effects at power-up.

14.2.8 Stability criterion

The stability criterion defines the interval on which measurements are considered as stable. Motion is indicated by b_4 bit of the measurement status register. A measurement is stable if X consecutive measurements following the reference measurement are included in the stability interval (see following table) else the current measurement becomes the new reference measurement. X depends on the A/D conversion rate.

Bits $b_2 b_1 b_0$	Stability criterion	Note
000	no motion detection (always stable)	
001	0,25d	1d = 1 scale interval
010	0,5d	
011	1d	
100	2d	
101	3d	
110	5d	
111	10d	

A/D conversion rate (meas/s)		X
50-Hz rejection	60-Hz rejection	
6,25	7,5	1
12,5	15	2
25	30	3
50	60	5
100	120	9
200	240	17
400	480	33
800	960	65
1600	1920	129

14.2.9 Decimal point position

Although **eNod4** measurements are integer values it is however possible to store a 'decimal point position' so as to design a display related to the application. Its value represents the number of decimal digits. If the variable is set to Zero, it means that decimal point is not used.

When using eNod4 for legal for trade purpose (legal for trade switch activated) some limitations apply on the decimal point value (see above).

Note: the decimal point is directly integrated to SCMBus protocol frames (see § SCMBus).

Admitted values: from 0 up to 7.

14.2.10 Unit

It is possible to store the display unit into the **eNod4**.

When using eNod4 for legal for trade purpose (legal for trade switch activated) some limitations apply on the unit value (see above).

Note: the unit is directly integrated to SCMBus protocol frames (see § SCMBus).

14.2.11 Options

<i>bit</i>	<i>Weighing status in read record</i>
<i>0</i>	<i>Saved Zero in non-volatile memory</i>
<i>1</i>	<i>Saved Tare in non-volatile memory</i>
<i>2 - 15</i>	<i>Reserved</i>

15 MODBUS RTU REGISTERS TABLE

<i>Chapter</i>	<i>Name</i>	<i>Modbus Address</i>	<i>Type</i>	<i>Access</i>
Modbus	Firmware revision	0x0000	Uint	RO
Modbus	Node number / baud rate	0x0001	Uint	RO
Legal for trade	Legal for trade version	0x0004 LSB	Byte	RO
Legal for trade	Legal for trade switch	0x0004 MSB	Byte	RW
Legal for trade	Legal for trade counter	0x0005	Byte	RO
Legal for trade	Legal for trade checksum	0x0006	Uint	RO
Legal for trade	Zero functions	0x0007	Uint	RW
Legal for trade	Stability criterion	0x0008 LSB	Byte	RW
Legal for trade	decimal point position	0x0008 MSB	Byte	RW
Legal for trade	Unit	0x0009	String	RW
Calibration	Maximum capacity	0x000C	Ulong	RW
Calibration	Number of calibration segments	0x000E	Uint	RW
Calibration	Calibration load 1	0x000F	Ulong	RW
Calibration	Calibration load 2	0x0011	Ulong	RW
Calibration	Calibration load 3	0x0013	Ulong	RW
Calibration	Sensor sensitivity	0x0015	Ulong	RW
Calibration	Scale interval	0x0017	Uint	RW
Calibration	Zero calibration	0x0018	Long	RW
Calibration	Span coefficient 1	0x001A	Float	RW
Calibration	Span coefficient 2	0x001C	Float	RW
Calibration	Span coefficient 3	0x001E	Float	RW
Calibration	Span adjusting coefficient	0x0020	Ulong	RW
Calibration	Calibration place g value	0x0022	Ulong	RW
Calibration	Place of use g value	0x0024	Ulong	RW
Legal for trade	DSD latest record ID	0x0028	Ulong	RO
I/O	External value to control analog output	0x0032	Uint	RW
HMI	HMI name	0x0034	String	RW
Filter	A/D conversion rate	0x0036	Uint	RW
Filter	filters activation	0x0037 LSB	Byte	RW
Filter	Low-pass order	0x0037 MSB	Byte	RW
Filter	Low-pass cut-off frequency	0x0038	Uint	RW
Filter	Band-stop high cut-off frequency	0x0039	Uint	RW
Filter	Band-stop low cut-off frequency	0x003A	Uint	RW
Protocol	Functioning mode / Serial protocol	0x003E	Uint	RW
SCMbus	SCMbus transmission period	0x003F	Uint	RW

<i>Chapter</i>	<i>Name</i>	<i>Modbus Address</i>	<i>Type</i>	<i>Access</i>
I/O	Analog output functioning (optional)	0x0040	Uint	RW
I/O	Logical inputs 3 functioning (optional)	0x0041 LSB	Byte	RW
I/O	Logical inputs 4 functioning (optional)	0x0041 MSB	Byte	RW
I/O	Logical input 1 functioning	0x0042 LSB	Byte	RW
I/O	Logical input 2 functioning	0x0042 MSB	Byte	RW
I/O	holding time	0x0043	Uint	RW
I/O	Output 1 functioning	0x0044 LSB	Byte	RW
I/O	Output 2 functioning	0x0044 MSB	Byte	RW
I/O	Output 3 functioning	0x0045 LSB	Byte	RW
I/O	Output 4 functioning	0x0045 MSB	Byte	RW
I/O	Set point 1 high value	0x0046	Long	RW
I/O	Set point 1 low value	0x0048	Long	RW
I/O	Set point 2 high value	0x004A	Long	RW
I/O	Set point 2 low value	0x004C	Long	RW
I/O	Set point 3 high value	0x004E	Long	RW
I/O	Set point 3 low value	0x0050	Long	RW
I/O	Set point 4 high value	0x0052	Long	RW
I/O	Set point 4 low value	0x0054	Long	RW
I/O	1&2 Set points functioning	0x0056 LSB	Byte	RW
I/O	3&4 Set points functioning	0x0056 MSB	Byte	RW
State Register	Measurement status	0x007D	Uint	RO
State Register	Gross measurement	0x007E	Long	RO
State Register	Tare value	0x0080	Long	RO
State Register	Net measurement	0x0082	Long	RO
State Register	Factory calibrated points	0x0084	Long	RO
Command	command register	0x0090	Uint	RW
Command	response register	0x0091	Uint	RO
Calibration	Zero offset	0x0092	Long	RW
State Register	Input levels	0x0094 LSB	Byte	RO
I/O	Input levels	0x0094 LSB	Byte	RO
State Register	output levels	0x0094 MSB	Byte	RO
I/O	output levels	0x0094 MSB	Byte	RO
State Register	Preset tare value	0x0095	Ulong	RW

***** JUMP TO EXTENSION MODBUS REGISTERS *****

<i>Chapter</i>	<i>Name</i>	<i>Modbus Address</i>	<i>Type</i>	<i>Access</i>
State Register	Sensor input control reference	0x0A44	long	RW
State Register	Sensor input control result	0x0A46	Int	RO
State Register	Sensor input control result max. tolerance	0x0A47	UInt	RW
State Register & I/O	Defective measurement debounced time	0x0A48	UInt	RW
State Register & I/O	Defective measurement alarm activation time	0x0A49	UInt	RW
Legal for trade	Options	0x0A50	UInt	RW
Legal for trade	DSD current record ID to read	0x0A60	Ulong	RW
Legal for trade	Oldest DSD record ID	0x0A8E	Ulong	RO
Legal for trade	Read record ID	0x0A90	Ulong	RO
Legal for trade	Net weight in read record	0x0A92	Ulong	RO
Legal for trade	Tare value in read record	0x0A94	Ulong	RO
Legal for trade	Weighing status in read record	0x0A96	UInt	RO
Legal for trade	Record checksum of read record	0x0A97	UInt	RO

16 CRC-16 CALCULATION ALGORITHM

