Software user manual

- annotations meaning :

- *MSB : most significant byte*
- *LSB : less significant byte*
- *RO : read only*
- *WO : write only*
- *RW : read / write access*
- *(M) : data can be mapped into a PDO (only for CANopen®)*

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 1600 meas/s with maximum scaled resolution of ±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 eNodView software tool

So as to configure **eNod4**, SCAIME provides **eNodView** software tool. **eNodView** is the software dedicated to **eNod** devices and ***AXD*** digital load cell configuration from a PC. Its 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 interface*
- *Numerical filters simulation*
- *Frequential analysis FFT*
- *Process control*

The **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.

1.3 Versions and options:

1.3.1 Versions:

- **eNod4 DIN**: Strain gauges load-cell conditioner with ***CANopen®*** et ***ModBus RTU*** communication. 22.5mm packaging width
  EDS configuration file for ***CANopen®*** can be downloaded from our web site:
  http://www.scaime.com
- **eNod4 PRO DIN** : Strain gauges load-cell conditioner with ***Profibus DP-V1*** et ***ModBus RTU*** communication. 35mm packaging width.
  GSD configuration file for ***Profibus DP-V1*** can be downloaded from our web site:
  http://www.scaime.com

1.3.2 Options :

With appropriate option the strain gauges load-cell can be exchanged with:

- 4/20mA analog signal.
- 0/10V analog signal.

2 ModBus RTU

2.1 Physical interfaces

ModBus RTU communication protocol can be used either through ***eNod4-T*** USB port or through the DB9 port (except for Profibus DPV1 compatible version).
USB port behaves as a RS232 interface whereas the DB9 port supports 2-wires RS485 (half-duplex) communication. Supported baud rates are 9600, 19200, 38400, 57600, and 115200.
For a complete description of the recommendations about ***eNod4-T*** RS485 connexion, please refer to documentation *ref. 196702*.

- **Note :** using ***eNod4-T*** through USB requires installing first the necessary USB drivers

2.2 Byte format

Data transmitted to ***eNod4-T*** 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. Appendix A : CRC-16 calculation algorithm).

2.3 ModBus RTU supported functions

As a ModBus RTU slave, ***eNod4-T*** supports following ModBus RTU functions :

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

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

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

2.4 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.**

2.4.1 Function (03_H/04_H) – 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 :

slave address	03_H or 04_H	NB *	data 1	…	CRC16
1 byte	1 byte	1 byte	2 bytes	2 bytes	2 bytes

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

2.4.2 Function (06_H) – write single register

⇨ request command sent to the slave :

slave address	06_H	register offset	data	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

⇨ slave response :

slave address	06_H	register offset	data	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

2.4.3 Function (10_H) – preset multiple registers (N = 30 max)

⇨ request command sent to the slave :

slave address	10_H	starting register offset	N registers	NB	Data 1	…	CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	2 bytes	2 bytes	2 bytes

⇨ slave response :

slave address	10_H	starting register offset	N registers	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

2.4.4 Error frames

⇨ frame format in case of a transaction error :

slave address	function code + 80_H	error code	CRC16
1 byte	1 byte	1 byte	2 bytes

⇨ Error codes meaning :

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

3 SCMBUS/FAST SCMBUS

3.1 Physical interfaces

SCMBus and fast SCMBus communication protocols can be used either through ***eNod4-T*** USB port or through the DB9 port (except for Profibus DPV1 compatible version).
USB port behaves as a RS232 interface whereas the DB9 port supports 2-wires RS485 (half-duplex) communication. Supported baud rates are 9600, 19200, 38400, 57600, and 115200. Some SCMBus/fast SCMBus functionalities are more adapted to full-duplex than to half-duplex communication. Please read carefully their description in the chapters below.
For a complete description of the recommendations about ***eNod4*** RS485 connexion, please refer to documentation *ref. 196702*.

- **Note :** using ***eNod4-T*** through USB requires installing first the necessary USB drivers

3.2 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-T*** 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)*

3.3 Byte format

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

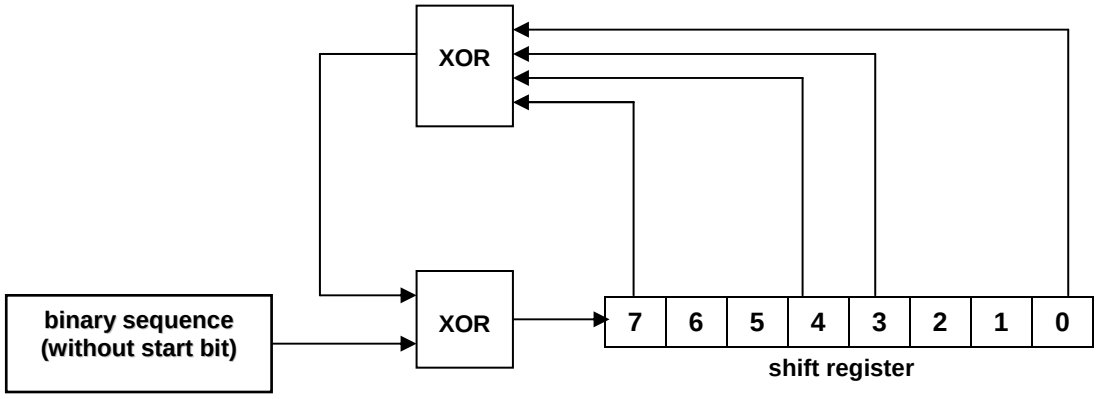
- 1 start bit
- 8 data bits
- no parity
- 2 stop bits

- 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-T*** 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.

3.4 Frames structure

3.4.1 transmission organization :

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

3.4.2 Reading request

⇨ request :

Address	Command	CR	CRC
1 **Hex** byte	1 **Hex** byte (command)	1 **ASCII** byte ($0D_H$)	1 **Hex** byte

⇨ response :

3.4.2.1 SCMBus:

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).

3.4.2.2 Fast SCMBus :

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. ***eNod4-T*** address is not transmitted in the frame.

3.4.3 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.

3.4.4 Error frame

In case of an error upon reception of a request, ***eNod4-T*** 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:

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

4 CANOPEN®

4.1 Physical interface

eNod4-T 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-T*** CAN connexion, please refer to documentation *ref. 196702*.

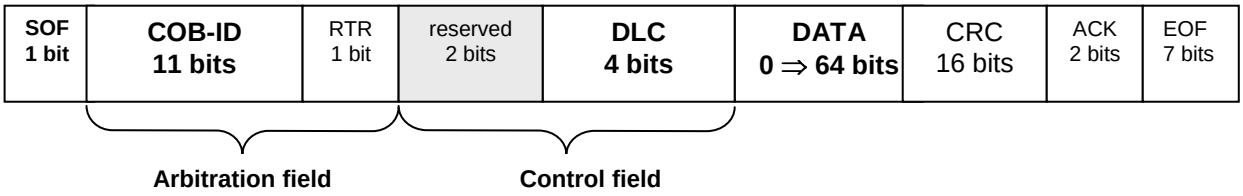
4.2 LED CANopen

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

4.3 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 ≤ DLC ≤ 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.

4.4 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.**

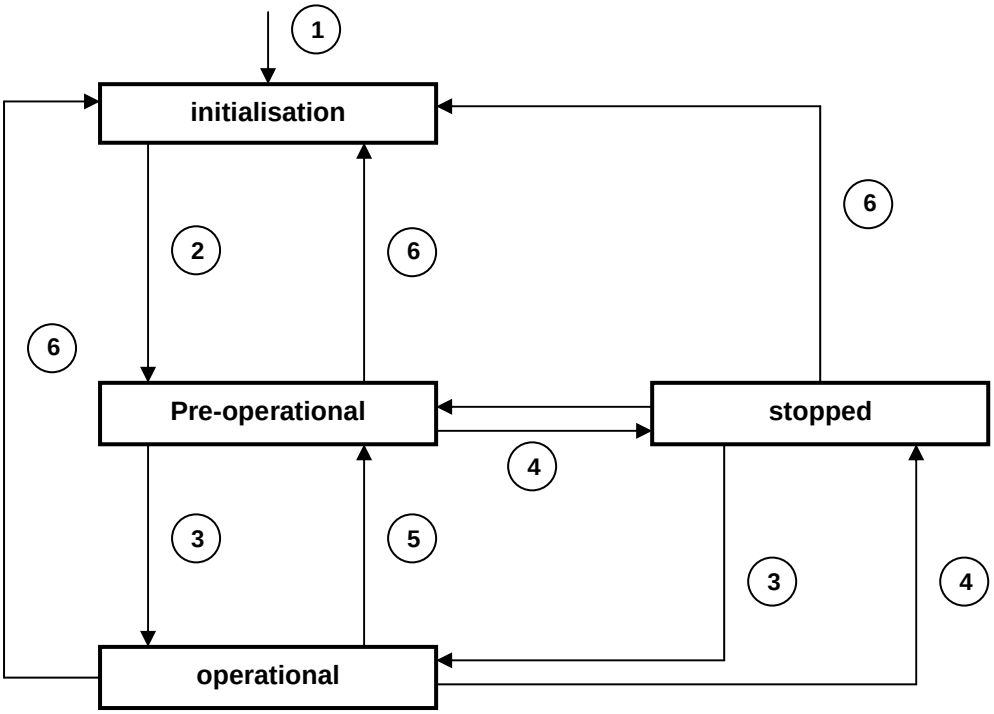
eNod4 supports 6 different message types :

- 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**

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

4.5 *eNod4* status remote management

For the CANopen® network, ***eNod4-T*** 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-T*** 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-T*** device power-up
- ⇨ 2 : automatic transition after the end of initialisation
- ⇨ 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 :

	Initialisation	Pre-operational	Operational	Stopped
SDO		X	X	
PDO			X	
SYNC		X	X	
Emergency		X	X	
NMT		X	X	X
Boot-up	X			
Heartbeat		X	X	X

4.5.1 NMT commands

Except during the initialisation phase, ***eNod4-T*** 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-bytes data frame with a COB-ID equal to zero:**

COB-ID	DLC	byte 1	byte 2
0	2	NMT code	Node identifier

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

4.5.2 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-T*** 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.

4.5.3 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 informations (hex.)	
4B4F	no error
474C	life time has elapsed or Heart Beat not received
564F	sensor signal outside of the input signal range
5054	CAN transmitter in error passive state
5052	CAN receiver in error passive state

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.

4.6 Error control services

CANopen® uses smart mechanisms to control permanently the nodes state on the bus. ***eNod4-T*** 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.

4.6.1 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-T*** means that its initialisation 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-T*** 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-T*** sends an emergency telegram and switches to pre-operational state.

4.6.2 Node guarding

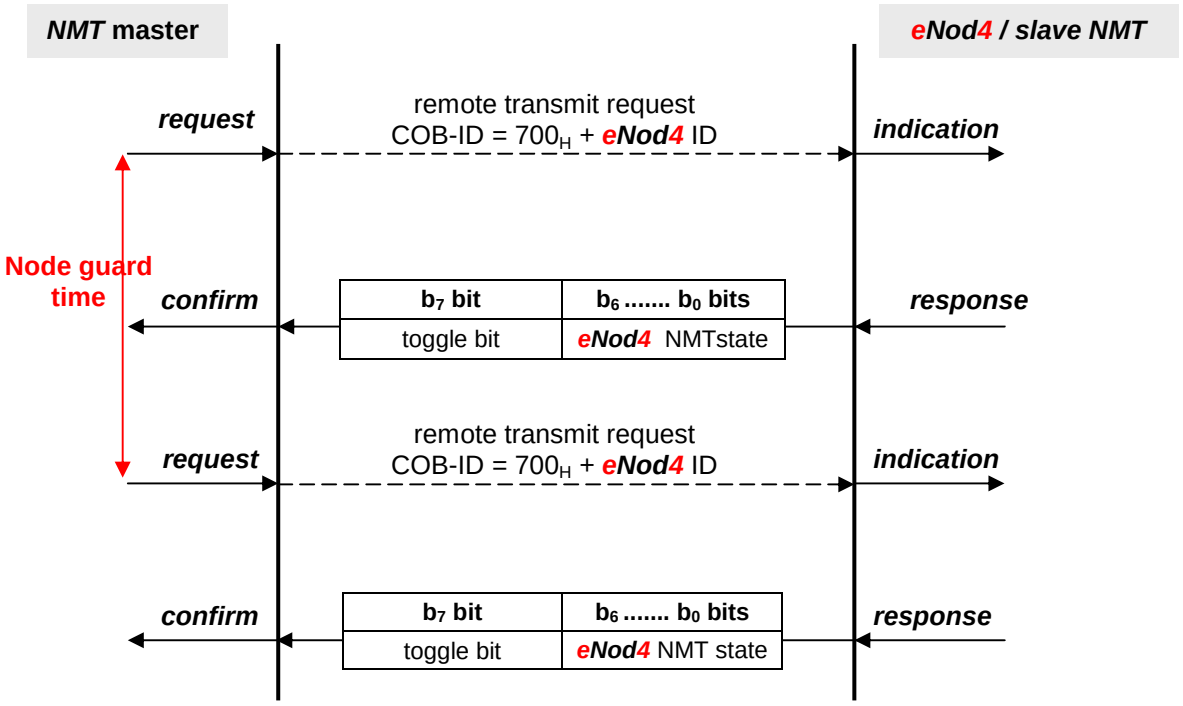
Node guarding protocol is another way to check the nodes state. But unlike Heartbeat protocol, it needs requests from a NMT master. In this case, the NMT master sends periodically a remote transmit request (remote frame) to the node with **COB-ID 700_H +** ID ***eNod4-T***. ***eNod4-T*** has to respond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



4.7 Access to the object dictionary

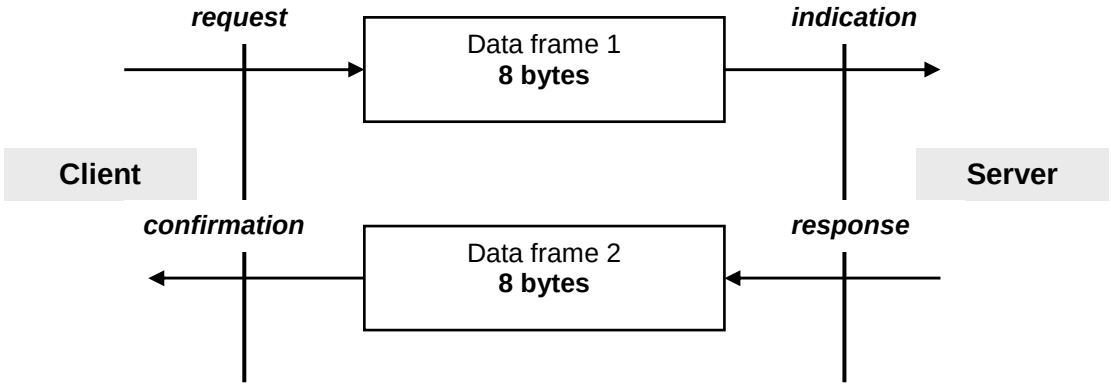
The most important element of a CANopen® compatible device is its **object dictionary (OD)**. Each node object that can be accessed via the bus is part of a table called object dictionary. The dictionary entries can be addressed by a couple of an index (2 bytes) and a sub-index (1 byte) with the following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

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

Only the greyed elements of the table are accessible through ***eNod4-T*** OD.
The whole object dictionary is accessible and can be configured from usual CANopen® configuration tools. This can be done using ***eNod4-T*** **available EDS file (named 196730_eNod4_T-X.eds)**.

4.7.1 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-T*** 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-T*** and the server uses the SDO(Tx) COB-ID (580_H + ID ***eNod4-T***).

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-T*** 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	SDO command specifier not supported
6010001	unsupported access to an object
6010002	attempt to write a read-only object
6020000	the object does not exist in the object dictionary
6040042	the number and length of the objects to be mapped would exceed PDO length
6040047	impossible operation (for example reading a net/gross value during a tare or a zero)
6070012	data type does not match, length of service parameter too high
6070013	data type does not match, length of service parameter too low
6090011	Sub index object does not exist.
6090030	value range of parameter exceeded
6090031	value of parameter written too high
6090032	value of parameter written too low
8000020	data can not be stored to the application
8000022	data can not be transferred or store to the application because of the present device state

4.7.2 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 **4 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	**PDO transmission/reception after a SYNC message following one of these events :** - 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	**PDO transmission after n SYNC messages** - Or after receipt of remote transmit request following at less one SYNC.
F1 - FB	reserved					
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	**PDO transmission is triggered by one of these events :** - activation of a logical input assigned to *'send TPDO'* - mapped object variation superior to +/- delta - receipt of remote transmit request more for TPDO2 and 3, functioning is identical to code FF
FF				X	X	**Periodic TPDO emission**. Period can be configured (min = 1 ms). RPDO handled upon reception TPDO emission after receipt of remote transmit request

- **Note 1:** for RPDO1-2-3-4 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 its 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 and TPDO1 mapping are configured with default values that can not be modified.

5 PROFIBUS DPV1

5.1 Physical interface

An ***eNod4-T*** 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-T*** Profibus DPV1 connection, please refer to documentation *ref. 196702*.

5.2 GSD file

eNod4-T 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 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*

5.3 Cyclic exchanges

5.3.1 Cyclic inputs modules

eNod4-T GSD file defines several modules that can be integrated to the Profibus DP cyclic frames. All types of measurements processed by ***eNod4-T*** 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.

5.3.2 Cyclic inputs/outputs modules

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

- **The *'read/write request register'*** described by ***eNod4-T*** 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-T*** 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-T*** 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**: the transaction request register **must** be set to 0x0000 before every new transaction.

- **The *'command/response register'*** described by ***eNod4-T*** 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.

5.4 Acyclic exchanges

For DPV1 class 1 and class 2 masters, another mechanism is available to write/read data into/from ***eNod4-T*** 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-T***. 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 $A1200007_H$.

5.5 eNod4 Profibus DP features

5.5.1 Sync

The *Sync* command can be transmitted from a master to one or more slaves. When receiving this command, ***eNod4-T*** 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.

5.5.2 Freeze

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

5.5.3 Fail-safe

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

⇨ ***eNod4-T*** Profibus DP cyclic inputs goes on being refreshed
⇨ ***eNod4-T*** 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.

5.5.4 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-T*** 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-T*** device

⇨ **Ident. High/Ident. Low bytes** : contain ***eNod4*** ident number ($0D2D_H$)

⇨ **extended diagnosis content** :

bits	meaning	note
b_0		
0	/	
1	input analog signal out of the A/D conversion range (negative quadrant)	
b_1		
0	/	
1	input analog signal out of the A/D conversion range (positive quadrant)	
b_2		
0	/	see §8 for the *'maximum capacity'* setting description
1	gross meas.< (- max capacity)	
b_3		
0		see §8 for the *'maximum capacity'* setting description
1	gross meas. > (max capacity)	
b_4		
0	EEPROM OK	
1	Default EEPROM	
b_{15} ..... b_5		
0	reserved	

6 MEASUREMENT TRANSMISSION

6.1 Principles

As a digital process transmitter, ***eNod4-T*** is able to transmit measurements after signal processing depending on the chosen communication protocol. Here are the accessible variables:

- *factory calibrated measurement*
- *gross measurement*
- *net measurement*
- *tare value*
- *measurement status*
- *logical I/O level*

The available communication protocols allow to transmit these measurements on the bus using several different mechanisms described below.

6.2 Transmission mechanisms

6.2.1 ModBus RTU

As a Master-Slave communication protocol, ModBus RTU only supports measurement transmission following a master request. But the frame size can be sufficient so as to contain all the measurement variables into only one slave answer.

6.2.2 SCMBus and fast SCMBus

SCMBus and fast SCMBus allow more flexible exchanges than ModBus RTU communication protocol. In fact, 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.

6.2.3 CANopen®

CANopen® includes smart transmission mechanisms that are presented in §4. All the measurements can be exchanged either through SDO read requests or through TPDO.
TPDO trigger sources are described in a table of §4 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.

6.2.4 Profibus DPV1

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 can not be accessed through acyclic requests.

6.3 Measurement access list & description

Variable	Data type	Access	SCMBus / fast SCMBus read command	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT
measurement status	unsigned16	RO		0x007D	0x5003 / 0x00 (M)	module 1
gross measurement	integer32	RO	0xA0	0x007E	0x5001 / 0x00 (M)	module 1
net measurement	integer32	RO	0xA2	0x0080	0x5000 / 0x00 (M)	module 2
tare value	integer32	RO	0xA1	0x0082	0x5004 / 0x01 (M)	**R : 0x0470 W : 0x0471**
factory calibrated measurement	integer32	RO	0xA3	0x0084	0x5002 / 0x00 (M)	module 3
logical I/O level	unsigned16 unsigned8 (CANopen®)	RO		0x0094	IN : 0x5100 / 0x00 (M) OUT : 0x5200 / 0x00 (M)	module 4
Preset tare	integer32	RW		0x0095	0x5004 / 0x02 (M)	**R : 0x0496 W : 0x0497**

6.3.1 Measurement status

- access : RO
- data type : unsigned16
- default value : /
- admitted values : /

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x007D	**0x5003 / 0x00 (M)**	**module 1 (2 first bytes)**	**/**

❑ **Note:** In SCMBus and fast SCMBus communication protocols the status bytes are included into every transmitted frame containing measurement data (cf. § 3).

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

bits	meaning	note
b_1 b_0		
00	gross measurement	only in SCMBus/fast communication protocols not significant otherwise (00)
01	net measurement	
10	factory calibrated measurement	
11	tare value	
b_3 b_2		
00	measurement OK	causes an output assigned to the *'defective measurement'* function to be set active
10	gross meas.< (- max capacity)	
10	gross meas. > (max capacity	

11	analog signal out of the A/D converter input range	
b_4		
0	motion	causes an output assigned to the *'motion'* function to be set active
1	no motion	
b_5		
0	measurement out of the ¼ of division	
1	zero in the ¼ of division	
b_6		
0	EEPROM OK	See hereunder **Note about EEPROM error management**
1	EEPROM failure	
b_7		
0	reserved	1 in SCMBus and fast SCMBus, 0 otherwise
1		
b_8		
0	IN1 logical level	
1		
b_9		
0	IN2 logical level	
1		
b_{10}		
0	OUT1 logical level	
1		
b_{11}		
0	OUT2 logical level	
1		
b_{12}		
0	OUT3 logical level	
1		
b_{13}		
0	OUT4 logical level	
1		
b_{14}		
0	no tare	
1	at least a tare has been processed	
b_{15}		
0	reserved	1 in SCMBus and fast SCMBus, 0 otherwise
1		

- **Note about 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 as following :

SCMBus / fast SCMBus	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
See b6 bit in status word attached to measurements	0x007D (Status word) see b6 bit	0x5003 / 0x00 (M) see b6 bit See §13.1.1 error register	module 1 (2 first bytes) See §5.5.4 b4 bit of extended diagnosis register	/

6.3.2 Gross measurement

- access : RO
- data type : integer32
- default value : /
- admitted values : /

SCMBus / fast SCMBus read command	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0xA0	0x007E	0x5001 / 0x00 (M)	module 1 (4 last bytes)	/

Description: 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).

6.3.3 Net measurement

- access : RO
- data type : integer32
- default value : /
- admitted values : /

SCMBus / fast SCMBus read command	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0xA1	0x0082	0x5000 / 0x00 (M)	module 2 (4 bytes)	/

Description: the *'net measurement'* stands for the digital value after measurement scaling and tare subtraction.

6.3.4 Tare value

- access : RO
- data type : integer32
- default value : 0
- admitted values : /

SCMBus / fast SCMBus read command	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0xA2	0x0080	0x5004 / 0x01 (M)	- read command : 0x0470 - write command : 0x0471	/

Description: the *'tare value'* stores the calibrated value that is subtracted from the *'gross measurement'* so as to give the *'net measurement'*.

6.3.5 Factory calibrated measurement

- access : RO
- data type: integer32
- default value : /
- admitted values : /

SCMBus / fast SCMBus read command	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0xA3	0x0084	0x5003 / 0x01 (M)	module 3	/

Description: the *'factory calibrated measurement'* contains the measurement value without the user calibration layer. That means it is directly linked to the analog input voltage (see §8).

6.3.6 Logical IN/OUT level

- access : RO
- data type : unsigned16 / unsigned8 (CANopen®)
- default value : 0
- admitted values : /

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0094	IN : 0x5100 / 0x00 (M) OUT : 0x5200 / 0x00 (M)	module 4	/

Description: the *'logical IN/OUT'* register allows reading any time ***eNod4-T*** logical inputs and outputs level.

- **Note** : the IN/OUT level flags can be accessed through a 16-bits register except in CANopen® communication protocol where this word is divided into 2 8-bits registers (1 dedicated to the IN and the other one dedicated to the OUT)

bits	meaning	note
b_0		
0	low	IIN1 level
1	high	
b_1		
0	low	IN2 level
1	high	
b_7 b_2		
0	reserved (0)	
b_8 (or b_0 in CANopen®)		
0	low	OUT1 level
1	high	
b_9 (or b_1 in CANopen®)		
0	low	OUT2 level
1	high	

b_{10} (or b_2 in CANopen®)		
0	low	OUT3 level
1	high	
b_{11} (or b_3 in CANopen®)		
0	low	OUT4 level
1	high	
b_{15} … b_{12} (or b_7 … b_4 in CANopen®)		
0	reserved (0)	

6.3.7 Preset tare:

- access : RW
- data type: integer32
- default value : /
- admitted values : /

SCMBus / fast SCMBus read command	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
	0x0095	**0x5004 / 0x02 (M)**	**R 0x0496 W 0x0497**	**/**

Description: Preset tare let to renew a tare value previously defined.

6.4 SCMBus/fast SCMBus specific transmission modes

SCMBus and fast SCMBus communication protocols allow ***eNod4-T*** 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):

- access : R/W
- data type : unsigned16
- default value : 100
- admitted values : from 0 up to 65535
- ModBus RTU register offset : 0x003F

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:

SCMBus/fast SCMBus functional command	Command code
start net measurement transmission	$E0_H$
start factory calibrated points measurement transmission	$E1_H$
stop continuous transmission	$E2_H$

- **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.

7 PROCESSING FUNCTIONAL COMMANDS

7.1 Principles

eNod4-T is able to handle several functional commands thanks to a couple of registers (except in SCMBus/fast SCMBus protocols, see §3) :

- ⇨ **the command register** : dedicated to accept the functional commands
- ⇨ **the response register :** gives the state of the command currently being processed by *eNod4-T*
  (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

This principle is used in CANopen®, ModBus RTU and Profibus DPV1.

- ❑ **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.**

7.1.1 Command register

- access : R/W
- data type : unsigned16 / unsigned8 (CANopen®)
- default value : 0
- admitted values : listed below

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0090	0x2003 / 0x00 (M)	module 6	/

7.1.2 Response register

- access : RO
- data type: unsigned16 / unsigned8 (CANopen®)
- default value : 0
- admitted values : **00_H, 01_H, 02_H, 03_H**

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0091	0x2004/ 0x00 (M)	module 6	/

- ❑ **Note:** 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-T* is able to handle the functional commands received through RPDO1 and the *'response register'* value changes are automatically transmitted through TPDO1.

7.2 Functional commands list

Functional command	Command code	Note
reset*	$D0_H$	
EEPROM storage	$D1_H$	see §7.3.1
restore default settings	$D2_H$	CANopen® see §7.3.3
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*	$F2_H$	

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

7.3 Functional commands description

7.3.1 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 §7.3.2) followed by a reset.

7.3.2 EEPROM storage

eNod4-T configuration and calibration are stored in a non-volatile memory (EEPROM). If changes are made in the device configuration, sending to ***eNod4-T*** the *'EEPROM storage'* functional command will allow ***eNod4-T*** 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*

❑ **Note** : In CANopen® communication protocol, another procedure can be applied to achieve an EEPROM storage :

In CANoppen® communication protocol, storing all settings into ***eNod4-T*** 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-T*** cell responds with a value of 1 that means that parameters are stored in non-volatile memory only on request.

⇨ **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

7.3.3 Restore default settings

The *'restore default settings'* command causes ***eNod4-T*** 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:** this functional command is not available in CANopen® communication protocol.

7.3.4 Zero

When receiving a *'zero'* functional command**,** ***eNod4-T*** 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 ±10% (±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.

7.3.5 Tare

When receiving a *'tare'* functional command**,** ***eNod4-T*** 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'* (see §6) is set to 1.

7.3.6 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'* (see §6) to be set back to 0.

7.3.7 Cancel last command

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

7.3.8 Theoretical scaling

The *'theoretical scaling'* functional command involves the *'maximum capacity'* and *the 'sensor sensitivity'* settings. When used, this command realizes an automatic scaling to migrate from the factory calibration to the user calibration (see §8). 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-T*** to the application.

7.3.9 Zero adjustment

The *'zero adjustment'* functional command allows the user to set his calibration zero value by asking ***eNod4-T*** 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-T*** 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.

7.3.10 Start physical calibration

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

7.3.11 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 (cf 7.3.9).

7.3.12 Segment 1/2/3 acquisition

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 5 seconds, ' *actual segment acquisition*' command is cancelled and an execution error is reported.

7.3.13 Store 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'*.

7.3.14 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.

7.3.15 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.

7.3.16 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 tis command.

8 CALIBRATION SETTINGS & PROCEDURES

8.1 Principles

Both ***eNod4-T*** analog channels are configured so as to deliver following values depending on the analog signal range:

- ⇨ **500 000 pts for 2 mV/V** on the low-level analog channel
- ⇨ **100 000 pts for 10 V_{DC}** on the 0-10V analog channel

The measurement scaling in ***eNod4-T*** 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.

8.2 Calibration methods

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

- ⇨ **a theoretical calibration** involving the sensitivity of the sensor and a user-defined corresponding capacity
- ⇨ **a physical calibration** involving 1, 2 or 3 know loads (for a load cell) or 1,2 or 3 measured voltages (for the 0-10V analog channel)

Both can be achieved thanks to the functional commands described in §7.

8.3 Settings list

Here is the list of settings that have an impact on the measurement scaling and on the calibration.

Setting	Data type	Access	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
maximum capacity	unsigned32	R/W	0x000C	0x3002 / 0x00	R : 0x0420 W : 0x0421	0x02 / 0x00
number of calibration segments	unsigned16	R/W	0x000E	0x3000 / 0x00	R : 0x0222 W : 0x0223	0x02 / 0x01
calibration load 1	unsigned32	R/W	0x000F	0x3001 / 0x01	R : 0x0424 W : 0x0425	0x02 / 0x02
calibration load 2	unsigned32	R/W	0x0011	0x3001 / 0x02	R : 0x0426 W : 0x0427	0x02 / 0x03
calibration load 3	unsigned32	R/W	0x0013	0x3001 / 0x03	R : 0x0428 W : 0x0429	0x02 / 0x04
sensor sensitivity	unsigned32	R/W	0x0015	0x3004 / 0x00	R : 0x042A W : 0x042B	0x02 / 0x05
scale interval	unsigned16	R/W	0x0017	0x3003 / 0x00	R : 0x022C W : 0x022D	0x02 / 0x06
Zero calibration	integer 32	R/W	0x0018	0x3006 / 0x00	R : 0x0434 W : 0x0435	0x03 / 0x04
span adjusting coefficient	float	R/W	0x0020	0x3005 / 0x01	R : 0x042E W : 0x042F	0x03 / 0x00
calibration place g value	unsigned32	R/W	0x0022	0x3005 / 0x02	R : 0x0430 W : 0x0431	0x03 / 0x01
place of use g value	unsigned32	R/W	0x0024	0x3005 / 0x03	R : 0x0432 W : 0x0433	0x03 / 0x02
Span coefficient 1	float	R/W	0x001A	0x3005 / 0x04	R : 0x0436 W : 0x0437	0x03 / 0x05
Span coefficient 2	float	R/W	0x001C	0x3005 / 0x05	R : 0x0438 W : 0x0439	0x03 / 0x06
Span coefficient 3	float	R/W	0x001E	0x3005 / 0x06	R : 0x043A W : 0x043B	0x03 / 0x07
zero offset	integer32	R/W	0x0092	0x2500 / 0x00	R : 0x0472 W : 0x0473	0x03 / 0x03

8.4 Settings description

8.4.1 Maximum capacity

- access : R/W
- data type : unsigned32
- default value : 500000
- admitted values : from 1 up to 1000000

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x000C	0x3002 / 0x00	- read command : 0x0420 - write command : 0x0421	0x02 / 0x00

***Description*:** 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 status bytes associated to the measurement 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) only is handled if the gross measurement value is comprised within a ±10% range of the *'maximum capacity'* (±2% if the legal for trade option is active).
The *'maximum capacity'* setting also allows calibrating ***eNod4-T*** 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.

8.4.2 Number of calibration segments

- access : R/W
- data type : unsigned16
- default value : 1
- admitted values : from 1 up to 3

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x000E	0x3000 / 0x00	- read command : 0x0222 - write command : 0x0223	0x02 / 0x01

***Description*:** the *'number of calibration segments'* defines how many calibration segments are used during the physical calibration procedure. Usually for linear installations, one segment is sufficient.

8.4.3 Calibration loads 1/2/3

- access : R/W
- data type : unsigned32
- default values : 10000 ; 20000 ; 30000
- admitted values : from 1 up to 1000000

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x000F 0x0011 0x0013	0x3001 / 0x01 0x3001 / 0x02 0x3001 / 0x03	- read commands : 0x0424 / 0x0426 / 0x0428 - write commands : 0x0425 / 0x0427 / 0x0429	0x02 / 0x02 0x02 / 0x03 0x02 / 0x04

***Description*:** before launching a physical calibration procedure, each calibration segment must be given a corresponding user value (for example, 1000 points for a 1 kg load).

8.4.4 Sensor sensitivity

- access : R/W
- data type : unsigned32
- default values : 200000

- admitted values : from 1 up to 1000000

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0015	0x3004 / 0x00	- read command : 0x042A - write command : 0x042B	0x02 / 0x05

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

⇨ 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 define the desired value delivered by ***eNod4-T*** for the associated signal using the *'maximum capacity'* and the *'sensor sensitivity'*.
This setting is expressed as a 10^{-5} value that means 197500 is equivalent to a 1.975 mV/V load cell sensitivity or a 1,975 V input voltage.

8.4.5 Scale interval

- access : R/W
- data type : unsigned16
- default value : 1
- admitted values : 1/2/5/10/20/50/100

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0017	0x3003 / 0x00	- read command : 0x022C - write command : 0x022D	0x02 / 0x06

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

8.4.6 Zero calibration

- access : R/W
- data type : integer32
- admitted values : from 0 up to ±1000000

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0018	0x3006 / 0x00	R: 0x0434 W: 0x0435	0x03 / 0x04

***Description*:** Zero reference in A/D converter points.
Zero calibration value corresponds to the A/D converter points measured during the '*zero acquisition*' step of a physical calibration. For a theoretical calibration it is necessary to set this value. It can be set automatically with the '*zero adjustment*' command.

❑ **Note:** A modification of this setting only is taken into account after EEPROM storage and device reboot (hardware or software).

8.4.7 Span adjusting coefficient

- access : R/W
- data type : unsigned32
- default value : 1000000
- admitted values : from 900000 up to 1100000

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0020	0x3005 / 0x01	- read command : 0x042E - write command : 0x042F	0x03 / 0x00

***Description*:** the *'span adjusting coefficient'* allows adjusting initial calibration. Adjustment applies linearly on the whole calibration curve. The unit for this coefficient is10^{-6} that means a value of 1000000 is equivalent to a span adjusting coefficient that is equal to 1.

- ❑ **Note:** A modification of this setting only is taken into account after EEPROM storage and device reboot (hardware or software).

8.4.8 Calibration place g value / place of use g value

- access : R/W
- data type :unsigned32
- default value : 9805470 (for both settings)
- admitted values : different from 0

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0022 0x0024	0x3005 / 0x02 0x3005 / 0x03	- read command : 0x0430/0432 - write command : 0x0431/0433	0x03 / 0x01 0x03 / 0x02

Description*:** if ***eNod4-T and the associated measuring chain have been calibrated somewhere different from the place where they are used, the gravity difference can affect the measure aptness. In order to compensate this error, it is possible to enter both gravity coefficients into ***eNod4***. The device automatically calculates the ratio and adapts its measurement scaling. The unit for this coefficient is 10^{-6} that means a value of 9805470 is equivalent to a g = 9.805470 $m.s^{-2}$.

Among ***eNodView*** software features, there is a tool that may help the user to determine the g value.

- ❑ **Note:** A modification of these settings only is taken into account after EEPROM storage and device reboot (hardware or software).

8.4.9 Span coefficients 1/2/3

- access : R/W
- data type : float
- admitted values : different from 0

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
Span coef. 1 0x001A	0x3005 / 0x04	R: 0x0436 W: 0x0437	0x03 / 0x05
Span coef. 2 0x001C	0x3005 / 0x05	R: 0x0438 W: 0x0439	0x03 / 0x06
Span coef. 3 0x001E	0x3005 / 0x06	R: 0x043A W: 0x043B	0x03 / 0x07

***Description*:** coefficients are automatically produced during calibration process. Writing by you these coefficients is only valid for a copy of a previous calibration.

- **Note**: A modification of these settings only is taken into account after EEPROM storage and device reboot (hardware or software).

8.4.10 Zero offset

- access : R/W
- data type : integer32
- default value : 0
- admitted values : different from 0

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0092	0x2500 / 0x00	- read command : 0x0472 - write command : 0x0473	0x03 / 0x03

***Description*:** 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 (see §7). Once the command has been successfully achieved, this register is reset to 0.

❑ **Note**: contrary to the other calibration settings, 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.

9 FILTERING OPTIONS

9.1 Principles

eNod4-T contains several 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** (with configurable order)
⇨ **a band-stop 2nd order Bessel-type filter** (with configurable bandwidth)
⇨ **a self-adaptive filter** (rather dedicated to static applications)

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

9.2 Limitations

Recursive filters like ***eNod4-T*** low-pass are generated by calculations that imply the filter order, the desired cut-off frequency and the sampling rate. So, there are some limitations it is necessary to respect in order to ensure a safe functioning of the signal processing. They are listed in the table below :

⇨ **50-Hz rejection :**

2nd order Bessel filter									
A/D conversion rate (meas/s)	6.25	12.5	25	50	100	200	400	800	1600
min low-pass cut-off frequency (Hz)	0.10	0.10	0.10	0.15	0.25	0.50	1.00	2.00	4.00

3rd order Bessel filter									
A/D conversion rate (meas/s)	6.25	12.5	25	50	100	200	400	800	1600
min low-pass cut-off frequency (Hz)	0.10	0.10	0.15	0.25	0.50	1.00	2.00	4.00	8.00

4th order Bessel filter

A/D conversion rate (meas/s)	6.25	12.5	25	50	100	200	400	800	1600
min low-pass cut-off frequency (Hz)	0.10	0.15	0.25	0.50	1.00	2.00	4.00	8.00	16.00

⇨ **60-Hz rejection :**

2nd order Bessel filter									
A/D conversion rate (meas/s)	7.5	15	30	60	120	240	480	960	1920
min low-pass cut-off frequency (Hz)	0.10	0.10	0.15	0.20	0.30	0.60	1.20	2.40	4.80

3nd order Bessel filter									
A/D conversion rate (meas/s)	7.5	15	30	60	120	240	480	960	1920
min low-pass cut-off frequency (Hz)	0.10	0.15	0.20	0.30	0.60	1.20	2.40	4.80	9.60

4th order Bessel filter									
A/D conversion rate (meas/s)	7.5	15	30	60	120	240	480	960	1920
min low-pass cut-off frequency (Hz)	0.15	0.20	0.30	0.60	1.20	2.40	4.80	9.60	19.20

9.3 Settings list

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

Setting	Data type	Access	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
A/D conversion rate	unsigned16	R/W	0x0036	0x4000 / 0x00	module 7	0x04 / 0x00
filters activation and order	unsigned16 unsigned8 (CANopen®)	R/W	0x0037	0x4001 / 0x01	module 7	0x04 / 0x01
				0x4001 / 0x02		
low-pass cut-off frequency	unsigned16	R/W	0x0038	0x4001 / 0x03	module 7	0x04 / 0x02
band-stop high cut-off frequency	unsigned16	R/W	0x0039	0x4001 / 0x04	module 7	0x04 / 0x03
band-stop high cut-off frequency	unsigned16	R/W	0x003A	0x4001 / 0x05	module 7	0x04 / 0x04

9.4 Settings description

9.4.1 A/D conversion rate

- access : R/W
- data type : unsigned16
- default value : 0x0010
- admitted values : see table below

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0036	**0x4000/ 0x00**	**- read command : 0x0240 - write command : 0x0241**	**0x04 / 0x00**

Description: contains a code that 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 (meas/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:** A modification of these settings only is taken into account after EEPROM storage and device reboot (hardware or software).

9.4.2 Filters activation & order

- access : R/W
- data type : unsigned16 / unsigned8
- default value : 0x0400
- admitted values : 0/2/3/4 (MSB) ; from 0 up to 3 (LSB)

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0037	**0x4001/ 0x01** **0x4001 / 0x02**	**- read command : 0x0242** **- write command : 0x0243**	**0x04 / 0x01**

Description: 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 :
  - 0x4001 / 0x01 : filters activation
  - 0x4001 / 0x02 : low-pass filter order

b_0	Meaning
0	band-stop filter disabled
1	band-stop filter enabled
b_1	
0	self-adaptive filter disabled
1	self-adaptive filter enabled
$b_{10}\ b_9\ b_8$ (or $b_2\ b_1\ b_0$ in CANopen®)	
000	low-pass filter disabled
010	2^{nd} order low-pass filter

011	3rd order low-pass filter
100	4th order low-pass filter

9.4.3 Low-pass filter cut-off frequency

- access : R/W
- data type : unsigned16
- default value : 500
- admitted values : from 10 up to 20000

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0038	**0x4001/ 0x03**	**- read command : 0x0244 - write command : 0x0245**	**0x04 / 0x02**

Description: 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. Be careful to enter a value in compliance with the table shown in §9.2.

9.4.4 Band-stop filter high cut-off frequency

- access : R/W
- data type : unsigned16
- default value : 1000
- admitted values : from 10 up to 20000

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0039	**0x4001/ 0x04**	**- read command : 0x0246 - write command : 0x0247**	**0x04 / 0x03**

Description: 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. Be careful always to keep a value that is higher that the band-stop filter low cut-off frequency.

9.4.5 Band-stop filter low cut-off frequency

- access : R/W
- data type : unsigned16
- default value : 3000
- admitted values : from 10 up to 20000

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x003A	**0x4001/ 0x05**	**- read command : 0x0248 - write command : 0x0249**	**0x04 / 0x04**

Description: 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. Be careful always to keep a value that is lower that the band-stop filter high cut-off frequency.

10 LOGICAL IN/OUT CONFIGURATION

10.1 Principles

eNod4-T device is equipped with 2 logical inputs and 4 logical outputs that are fully configurable.

10.1.1 Logical inputs

Each input can work individually in positive or negative logic. A holding time attached to both 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. Its functioning is similar to description in §7.3.5.
- **zero:** a rising (positive logic) or a falling edge (negative logic) causes a zero function to be triggered Its functioning is similar to description in §7.3.4.
- **cancel tare:** a rising (positive logic) or a falling edge (negative logic) causes the current stored tare to be erased.
- **transmit measurement:** only available in CANopen® and SCMBus/fast SCMBus protocols. 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 chosen level.

10.1.2 Logical outputs

The available functions are:

- **none :** the output has no function
- **motion :** the output is dedicated to copying the stability flag level (see §6).
- **defective measurement** : the output level is given by the logical OR operation between the various defects concerning measurement range (see §6).
- **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 X image :** the output is dedicated to copying a 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 driven by master requests (see §7).

10.2 Settings list

Here is the list of the settings that are involved in the logical inputs/outputs configuration :

Setting	Data type	Access	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
inputs assignment	unsigned16 unsigned8 (CANopen®)	R/W	0x0042	0x4501 / 0x02 0x4501 / 0x03	module 7	0x05 / 0x00
holding time	unsigned16	R/W	0x0043	0x4501 / 0x01	module 7	0x05 / 0x01
outputs 1&2 assignment	unsigned16 unsigned8 (CANopen®)	R/W	0x0044	0x4509 / 0x01 0x4509 / 0x02	module 7	0x06 / 0x00
outputs 3&4 assignment	unsigned16 unsigned8 (CANopen®)	R/W	0x0045	0x4509 / 0x03 0x4509 / 0x04	module 7	0x06 / 0x01
set point 1 high value	integer32	R/W	0x0046	0x4601 / 0x02	module 7	0x06 / 0x02
set point 1 low value	integer32	R/W	0x0048	0x4601 / 0x03	module 7	0x06 /0x03
set point 2 high value	integer32	R/W	0x004A	0x4601 / 0x04	module 7	0x06 / 0x04
set point 2 low value	integer32	R/W	0x004C	0x4601 / 0x05	module 7	0x06 / 0x05

set point 3 high value	integer32	R/W	0x004E	0x4609 / 0x02	module 7	0x06 / 0x06
set point 3 low value	integer32	R/W	0x0050	0x4609 / 0x03	module 7	0x06 / 0x07
set point 4 high value	integer32	R/W	0x0052	0x4609 / 0x04	module 7	0x06 / 0x08
set point 4 low value	integer32	R/W	0x0054	0x4609 / 0x05	module 7	0x06 / 0x09
set points functioning	unsigned16 unsigned8 (CANopen®)	R/W	0x0056	0x4601 / 0x01 0x4609 / 0x01	module 7	0x06 / 0x0A

10.3 Settings description

10.3.1 Logical inputs assignment

- access : R/W
- data type : unsigned16 / unsigned8
- default value : 0x1211 (or 0x12 and 0x11)
- admitted values : see table below

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0042	**0x4501/ 0x02 0x4501 / 0x03**	**- read command : 0x0250 - write command : 0x0251**	**0x05 / 0x00**

Description:

bits	meaning	note
b_3 b_2 b_1 b_0	**input 1 assignment**	
0000	none	the input has no function
0001	tare	equivalent to the functions described in §7
0010	zero	
0011	cancel tare	
0100	transmit measurement*/send TPDO2**	data is transmitted on the bus at every rising or falling edge (depending on the chosen logical)
0101	measurement window*	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
b_4	**input 1 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**	
00	gross	only for SCMBus/fast SCMBus protocols, no effect otherwise
01	net	
10	factory calibrated measurement	

bits	meaning	note
b_{11} b_{10} b_9 b_8 (or b_3 b_2 b_1 b_0 in CANopen®)	**input 2 assignment**	
0000	none	the input has no function
0001	tare	equivalent to the functions described in §7
0010	zero	
0011	cancel tare	
0100	transmit measurement*/send TPDO3**	data is transmitted on the bus at every rising or falling edge (depending on the chosen logical)
0101	measurement window*	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
b_{12} (or b_4 in CANopen®)	**input 2 logical**	
0	negative logic	defines the edge (or level) that triggers input 1 function
1	positive logic	
b_{14} b_{13} (or b_6 b_5 in CANopen®)	**measurement to be transmitted**	
00	gross	only for SCMBus/fast SCMBus protocols, no effect otherwise
01	net	
10	factory calibrated measurement	

- **Note:** the functions with a * only are possible in SCMBus and fast SCMBus protocols.
- **Note 2:** in CANopen® communication protocol, to use the *'send TPDOX'* function, it is necessary to configure the TPDO communication parameters (and particularly the communication type, see §4) and the mapping for the corresponding TPDO.

10.3.2 Holding time

- access : R/W
- data type : unsigned16
- default value : 100
- admitted values : from 0 up to 65535

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0043	**0x4501/ 0x01**	**- read command : 0x0252 - write command : 0x0253**	**0x05 / 0x01**

Description: The holding time corresponds to the minimum required stabilization time of the logical inputs before their activation. If the input level varies within this interval, it is ignored.

10.3.3 Logical outputs 1&2 assignment

- access : R/W
- data type : unsigned16/unsigned8
- default value : 0x1010 (or 0x10 and 0x10)
- admitted values : see table below

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0044	**0x4509/ 0x01 0x4509/0x02**	**- read command : 0x0254 - write command : 0x0255**	**0x06 / 0x00**

Description :

bits	meaning	note
b_3 b_2 b_1 b_0	**output 1 assignment**	
0000	none	the output level does not vary
0001	set point 1	functioning described by the *'set point functioning'* setting and by the *'set point 1 high and low values'*
0010	motion	copies the motion flag of the status bytes (cf. §6)
0011	defective measurement	error flag representing the OR logical operation between the error bits of the status bytes (cf. §6)
0100	input 1 image	copies input 1 level
0101	level on request	output 1 level is driven by the *'OUT1 activation/deactivation'* functional command (cf. §7)
b_4	**output 1 logical**	
0	negative logic	defines the output level when enabled
1	positive logic	
b_{11} b_{10} b_9 b_8 (or b_3 b_2 b_1 b_0 in CANopen®)	**output 2 assignment**	
0000	none	the output level does not vary
0001	set point 2	functioning described by the *'set point functioning'* setting and by the *'set point 2 high and low values'*
0010	motion	copies the motion flag of the status bytes (cf. §6)
0011	defective measurement	error flag representing the OR logical operation between the error bits of the status bytes (cf §6)
0100	input 2 image	copies input 2 level
0101	level on request	output 2 level is driven by the *'OUT2 activation/deactivation'* functional command (cf. §7)
b_{12} (or b_4 in CANopen®)	**output 2 logical**	
0	negative logic	defines the output level when enabled
1	positive logic	defines the output level when enabled

10.3.4 Logical outputs 3&4 assignment

- access : R/W
- data type : unsigned16/unsigned8
- default value : 0x1010 (or 0x10 and 0x10)
- admitted values : see table above

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0045	**0x4509/ 0x03 0x4509/0x04**	**- read command : 0x0256 - write command : 0x0257**	**0x06 / 0x01**

Description : similar to the outputs 1&2 configuration parameter, see previous paragraph (replacing all references to output 1 by output 3 and all references to output 2 by output 4).

10.3.5 Set points functioning

- access : R/W
- data type : unsigned16/unsigned8
- default value : 0x3333 (or 0x33 and 0x33)
- admitted values : see table below

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0056	**0x4601 / 0x01 0x4609 / 0x01**	**- read command : 0x0258 - write command : 0x0259**	**0x06 / 0x0A**

Description:

bits	meaning	note
b_0	**set point 1 commutation mode**	
0	window	only if output 1 assigned to the *'set point'* function
1	hysteresis	
b_2 b_1	**set point 1 comparison measurement**	
00	gross	
01	net	
b_3	reserved (0)	
b_4	**set point 2 commutation mode**	
0	window	only if output 2 assigned to the *'set point'* function
1	hysteresis	
b_6 b_5	**set point 2 comparison measurement**	
00	gross	
01	net	
b_7	reserved (0)	
b_8 (or b_0 in CANopen®)	**set point 3 commutation mode**	
0	window	only if output 3 assigned to the *'set point'* function
1	hysteresis	

b_{10} b_9 (or b_2 b_1 in CANopen®)	**set point 3 comparison measurement**	
00	gross	
01	net	
b_{11} (or b_3 in CANopen®)	reserved (0)	
b_{12} (or b_4 in CANopen®)	**set point 4 commutation mode**	
0	window	only if output 4 assigned to the *'set point'* function
1	hysteresis	
b_{14} b_{13} (or b_6 b_5 in CANopen®)	**set point 4 comparison measurement**	
00	gross	
01	net	
b_{15} (or b_7 in CANopen®)	reserved (0)	

10.3.6 Set points high and low values

- access : R/W
- data type : integer32
- default values : 20000/10000 (set point 1) ; 40000/30000 (set point 2) ; 60000/50000 (set point 3) ; 80000/70000 (set point 4)
- admitted values : from -1000000 to 1000000

⇨ **set point 1 (high – low) :**

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0046 0x0048	0x4601 / 0x02 0x4601 / 0x03	- read commands : 0x045A / 0x045C - write commands : 0x045B / 0x045D	0x06 / 0x02 0x06 / 0x03

⇨ **set point 2 (high – low) :**

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x004A 0x004C	0x4601 / 0x04 0x4601 / 0x05	- read commands : 0x045E / 0x0460 - write commands : 0x045F / 0x0461	0x06 / 0x04 0x06 / 0x05

⇨ **set point 3 (high – low) :**

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x004E 0x0050	0x4609 / 0x02 0x4609 / 0x03	- read commands : 0x0462 / 0x0464 - write commands : 0x0463 / 0x0465	0x06 / 0x06 0x06 / 0x07

⇨ **set point 4 (high – low) :**

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0052 0x0054	0x4609 / 0x04 0x4609 / 0x05	- read commands : 0x0466 / 0x0468 - write commands : 0x0467 / 0x0469	0x06 / 0x08 0x06 / 0x09

Description: each set point is described by its commutation mode (hysteresis/window) and by a couple of values that are constantly compared to the gross or net measurement (depending on the configuration the set point has been given) in order to define the corresponding output logical level. For more details about the set points functioning, please refer to documentation ref. 196702.

11 LEGAL FOR TRADE OPTIONS & PRODUCT IDENTIFICATION

11.1 Principles

The legal for trade options are a set of functions and indicators that are generally used in weighing applications. They have an impact on the device behaviour regarding the metrological constraints and allow tracking any changes concerning its configuration that may affect the measurement determination.
In addition, ***eNod4-T*** is identified by several values that characterize specifically each device.

11.2 Legal for trade settings list

Here is the list of the settings that need to be taken into account for a legal for trade use :

Setting	Data type	Access	ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
legal for trade switch	unsigned8	R/W	0x0004	0x3600 / 0x01	module 6	0x01 / 0x00
legal for trade software version	unsigned8	RO		0x3600 / 0x02		
legal for trade counter	unsigned16	RO	0x0005	0x3600 / 0x03	module 6	0x01 / 0x01
legal for trade checksum	unsigned16	RO	0x0006	0x3600 / 0x04	module 6	
zero functions	unsigned16	R/W	0x0007	0x3500 / 0x00	module 6	0x01 / 0x02
stability criterion	unsigned8	R/W	0x0008	0x3605 / 0x00	module 6	0x01 / 0x03
decimal point position	unsigned8	R/W		0x3700 / 0x02		
unit	visible string	R/W	0x0009	0x3700 / 0x01	module 6	0x01 / 0x04

11.3 Settings description

11.3.1 Legal for trade switch

- access : R/W
- data type : unsigned8
- default value : 0
- admitted values : 0 or 1

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0004 (MSB)	0x3600 / 0x01	- read command : 0x0210 - write command : 0x0211 (MSB)	0x01 / 0x00 (MSB)

Description: this setting activates (b_0 bit set to to 1) or deactivates (b_0 bit set to 0) criteria and parameters related to the use of ***eNod4*** in OIML compliance.
The *'legal for trade'* option activation leads to the following changes :

- ⇨ the *'legal for trade counter'* is incremented every time a storage into EEPROM is requested if one or several metrological settings have been modified (cf. §11.3.3).
- ⇨ a new *'legal for trade checksum'* value is calculated every time a storage into EEPROM is requested if one or several metrological settings have been modified (cf. §11.3.3).
- ⇨ taring is now impossible if gross measurement is negative.
- ⇨ the measurement value variations can not be read during the 15 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

11.3.2 Legal for trade software version

- access : RO
- data type : unsigned8
- default value : 1
- admitted values : /

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0004 (LSB)	0x3600 / 0x02	- read command : 0x0210 (LSB)	0x01 / 0x00 (LSB)

Description: this RO value identifies the version of the part of the software that is dedicated to the metrology and the measurement exploitation.

11.3.3 Legal for trade counter

- access : RO
- data type : unsigned16
- default value : 0
- admitted values : from 0 up to 65535

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0005	0x3600 / 0x03	- read command : 0x0212	0x01 / 0x01 (MSB)

Description: if the *'legal for trade'* option is enabled, the legal for trade counter is incremented every time a storage into EEPROM is requested if at least one (or several) of these settings has been modified:

- ⇨ legal for trade switch
- ⇨ stability criterion
- ⇨ decimal point position
- ⇨ maximum capacity
- ⇨ number of calibration segments
- ⇨ calibration loads 1/2/3
- ⇨ scale interval
- ⇨ span adjusting coefficient
- ⇨ calibration place/place of use g values
- ⇨ sensitivity
- ⇨ A/D conversion rate
- ⇨ filtering configuration (activation option, order and cut-off frequencies)
- ⇨ unit
- ⇨ zero functions

11.3.4 Legal for trade checksum

- access : RO
- data type : unsigned16
- default value : 0
- admitted values : from 0x0000 up to 0xFFFF

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0006	0x3600 / 0x04	- read command : 0x0214	0x01 / 0x01 (LSB)

Description: if the 'legal for trade' option is enabled, a new legal for trade checksum is calculated every time a storage into EEPROM is requested if at least one (or several) of the settings listed above (see §11.3.3) has been modified.

11.3.5 Zero functions

- access : R/W
- data type : unsigned16
- default value : 0
- admitted values : from 0 up to 3

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0007	0x3500 / 0x00	- read command : 0x0216 - write command : 0x0217	0x01 / 0x02

Description: the zero tracking and the initial zero setting can be respectively enabled by setting b_0 bit or b_1 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).

11.3.6 Stability criterion

- access :R/W
- data type : unsigned8
- default value : 0x01
- admitted values : from 0 up to 4

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0008 (LSB)	0x3605 / 0x00	- read command : 0x0218 - write command : 0x0219 (LSB)	0x01 / 0x03 (LSB)

Description: 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.

A/D conversion rate (meas/s)		X
50-Hz rejection	60-Hz rejection	
6.25	7.50	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

For the possible coding values, see table below :

bits	meaning	note
$b_2\, b_1\, b_0$	**stability criterion**	
000	no motion detection	always stable
001	0.25d	d = divisions
010	0.5d	
011	1d	
100	2d	

11.3.7 Decimal point position

- access : R/W
- data type : unsigned8
- default value : 0
- admitted values : from 0 up to 7

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0008 (MSB)	**0x3700 / 0x02**	**- read command : 0x0218 - write command : 0x0219 (MSB)**	**0x01 / 0x03 (MSB)**

Description: although ***eNod4-T*** 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 after the point, the zero value indicates that it is not used.

❑ **Note:** the decimal point is directly integrated to SCMBus protocol frames (see §6).

11.3.8 Unit

- access : R/W
- data type : visible string (4 characters)
- default value : 0x20206B67 (↔ kg)
- admitted values : any 4-characters string

ModBus RTU register offset	CANopen® Index/sub-index	Profibus cyclic IN/OUT	DPV1 acyclic slot/index
0x0009	**0x3700 / 0x01**	**- read command : 0x041A - write command : 0x041B**	**0x01 / 0x04**

Description: it is possible to store an '*unit*' into ***eNod4-T*** so as to design a display related to the application.

❑ **Note:** the unit is directly integrated to SCMBus protocol frames (see §6).

11.4 Product identification access

Depending on the communication protocol, the methods to access the device identification information is different.

11.4.1 ModBus RTU

This communication protocol gives access to a read-only register containing the firmware revision and also a product identification code :

ModBus RTU register offset	Description	Access	Default value	Type
0x0000	firmware revision	RO	0x6073	unsigned16

The 12 less significant bits stands for the firmware version (073_H = 115) whereas the 4 upper bits represent the firmware version (6_H is equivalent to an ***eNod4-T*** firmware version).
The device serial number is also accessible as a read-only setting :

ModBus RTU register offset	Description	Access	Default value	Type
0x0002	serial number	RO	1000001	unsigned32

11.4.2 CANopen®

The device identification settings are a part of the communication profile area (see §4) of the object dictionary.

⇨ **0x1009 : Manufacturer hardware version**

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	manufacturer hardware version	RO	0x32302E31 (1.02)	N	visible string

⇨ **0x100A : Manufacturer software version**

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	manufacturer software version	RO	0x35312E31 (1.15)	N	visible string

⇨ **0x1018 : Identity object :**

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	max. sub-index	RO	0x04	N	unsigned8
0x01	vendor ID	RO	0x00000142	N	unsigned32
0x02	product code	RO	0x000816E0	N	unsigned32
0x03	revision number	RO	0x0001000F	N	unsigned32
0x04	serial number	RO	0x000186A1	N	unsigned32

11.4.3 Profibus DPV1

As a DPV1 compatible device, ***eNod4-T*** supports record 0 of Identification and Maintenance (I&M0). A Profibus master can access its content through standard DPV1 request. It is composed of 64 bytes that contain the following information :

Offset	Content	Size	Default value
0x00	manufacturer specific	10 bytes (char)	eNod4T PRO
0x0A	manufacturer ID	2 bytes	reserved (0)
0x0C	order ID	20 bytes (char)	530145
0x20	serial number	16 bytes (char)	100001
0x30	hardware revision	2 bytes	0x0002
0x32	software revision	4 bytes (char)	V115
0x36	revision counter	2 bytes	0x0001
0x38	profile ID	2 bytes	reserved (0)
0x3A	profile specific type	2 bytes	reserved (0)
0x3C	I&M version	2 bytes	0x0100
0x3E	I&M supported	2 bytes	0x0001

12 OTHER CANOPEN® SPECIFIC OBJECTS

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

12.1 Communication objects

12.1.1 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 se 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

12.1.2 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-T*** answers the SDO abort code 0x06090030.

12.1.3 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 to 0x7E3

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

12.1.4 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-T*** responds to NMT master periodic (period equal to life guard) remote transmit requests.

12.1.5 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-T*** state is set to stopped. ***eNod4-T*** behaviour while stopped can be configured via the object at index 0x4800.

12.1.6 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-T*** is stored at this index. Its value automatically is updated if the node identifier is modified.

12.1.7 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-T*** 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)

12.1.8 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-T*** periodically generates a Heartbeat frame. It is expressed in ms and must be comprised between 1 and 65535.

12.1.9 0x4800 : Safety mode

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

Description: This entry defines ***eNod4-T*** 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.

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

12.1.10 Error behavior

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
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-T*** shall enter by default autonomously the pre-operational mode.
Alternatively, by error class coding in object 0x1029 ***eNod4-T*** can enter in the following states :

- 0: Pre-operational
- 1: No state change
- 2: Stopped. refer to '*Safety mode*' (see § 12.1.9)

12.2 PDO-related objects

12.2.1 *eNod4-T* RPDO default mapping

⇨ **0x1600 : RPDO1 mapping parameters**

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	number of supported objects	RO	1	N	unsigned8
0x01	1st object mapping	RO	0x20030008 (command register)	N	unsigned32

⇨ **0x1601 : RPDO2 mapping parameters**

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	number of supported objects	RO	1	N	unsigned8

0x01	1st object mapping	RO	0x30010120 (calibration load 1)	N	unsigned32

⇨ **0x1602 : RPDO3 mapping parameters**

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	number of supported objects	RO	2	N	unsigned8
0x01	1st object mapping	RO	0x25000020 (zero offset)	N	unsigned32
0x02	2nd object mapping	RO	0x30050120 (span adjusting coefficient)	N	unsigned32

⇨ **0x1603 : RPDO4 mapping parameters**

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

12.2.2 *eNod4-T* TPDO default mapping

⇨ **0x1A00 : TPDO1 mapping parameters**

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	number of supported objects	RO	1	N	unsigned8
0x01	1st object mapping	RO	0x20040008 (response register)	N	unsigned32

⇨ **0x1A01 : TPDO2 mapping parameters**

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	number of supported objects	R/W	2	N	unsigned8
0x01	1st object mapping	R/W	0x50010020 (gross measurement)	N	unsigned32
0x02	2nd object mapping	R/W	0x50030010 (measurement status)	N	unsigned16
0x03	3rd object mapping	R/W	0	N	unsigned32

⇨ **0x1A02 : TPDO3 mapping parameters**

Sub-index	Description	Access	Default value	Mappable (PDO) ?	Type
0x00	number of supported objects	R/W	3	N	unsigned8
0x01	1st object mapping	R/W	0x50000020 (net measurement)	N	unsigned32
0x02	2nd object mapping	R/W	0x51000008 (logical inputs level)	N	unsigned8
0x03	3rd object mapping	R/W	0x52000008 (logical outputs level)	N	unsigned8

- **Note:** TPDO2 and TPDO3 are programmable. TPDO1 is not programmable.
  To set a new mapping, the procedure is as following :
  - Set ***eNod4-T*** in '*pre-operational mode*'.
  - Disable current TPDO mapping setting to zero number of supported objects (sub-index 0x00).
  - Write new mapping.
  - Write in sub-index 0x00 number of mapped objects
  - Save in EEPROM.

APPENDIX A: MODBUS RTU REGISTER MAP

Register address (Hex)	Size in bytes (n)	Type	Name	Access
0000	2	Uint	firmware revision	RO
0001	2	Uint	node number / baud rate	RO
0002	4	Ulong	device serial number	RO
0004	2	Uint	legal for trade switch and version	R/W
0005	2	Uint	legal for trade counter	RO
0006	2	Uint	legal for trade checksum	RO
0007	2	Uint	zero functions	R/W
0008	2	Uint	stability criterion / decimal point position	R/W
0009	4	string	unit	R/W
000B	2		reserved	
000C	4	Ulong	maximum capacity	R/W
000E	2	Uint	number of calibration segments	R/W
000F	4	Ulong	calibration load 1	R/W
0011	4	Ulong	calibration load 2	R/W
0013	4	Ulong	calibration load 3	R/W
0015	4	Ulong	sensor sensitivity	R/W
0017	2	Uint	scale interval	R/W
0018	4	long	zero calibration	RO
001A	4	float	span coefficient 1	RO
001C	4	float	span coefficient 2	RO
001E	4	float	span coefficient 3	RO
0020	4	Ulong	span adjusting coefficient	R/W
0022	4	Ulong	calibration place g value	R/W
0024	4	Ulong	place of use g value	R/W
0026	32		reserved	
0036	2	Uint	A/D conversion rate	R/W
0037	2	Uint	low-pass order / filters activation	R/W
0038	2	Uint	low-pass cut-off frequency	R/W
0039	2	Uint	band-stop high cut-off frequency	R/W
003A	2	Uint	band-stop low cut-off frequency	R/W
003B	6		reserved	
003E	2	Uint	serial protocol	R/W
003F	2	Uint	sampling period	R/W
0040	4		reserved	
0042	2	Uint	logical inputs functioning	R/W
0043	2	Uint	holding time	R/W
0044	2	Uint	outputs 1 & 2 functioning	R/W
0045	2	Uint	outputs 3 & 4 functioning	R/W
0046	4	long	set point 1 high value	R/W
0048	4	long	set point 1 low value	R/W
004A	4	long	set point 2 high value	R/W

004C	4	long	set point 2 low value	R/W
004E	4	long	set point 3 high value	R/W
0050	4	long	set point 3 low value	R/W
0052	4	long	set point 4 high value	R/W
0054	4	long	set point 4 low value	R/W
0056	2	Uint	set points functioning	R/W
0057	76		reserved	
007D	2	Uint	measurement status	RO
007E	4	long	gross measurement	RO
0080	4	long	tare value	RO
0082	4	long	net measurement	RO
0084	4	long	factory calibrated points	RO
0086	20		reserved	
0090	2	Uint	command register	R/W
0091	2	Uint	response register	ROI
0092	4	long	delta zero	R/W
0094	2	Uint	IN/OUT level	RO
0095	4	long	Preset tare value	R/W

APPENDIX B: CRC-16 CALCULATION ALGORITHM