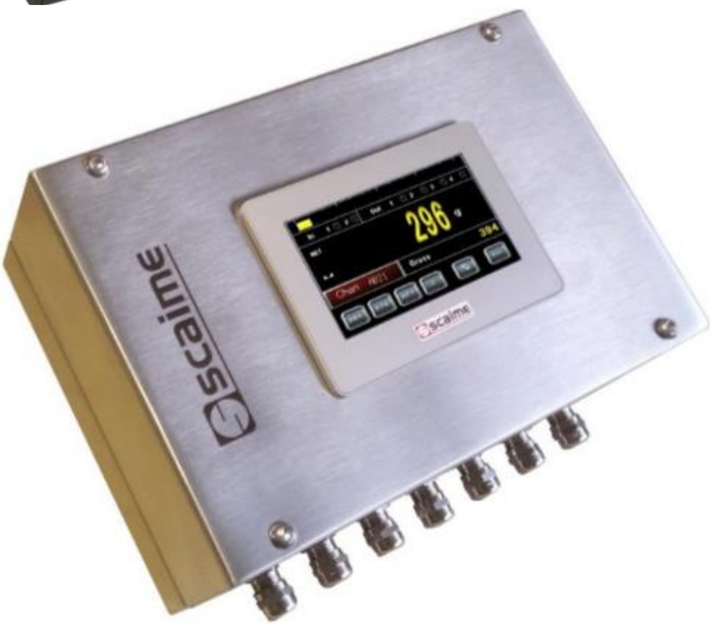
Software User Manual

eNod4-F ETHERNET

Digital Process Transmitter



SCAIME SAS – Technosite Altéa – 294, Rue Georges Charpak – 74100 JUVIGNY - FRANCE
Tél. : +33 (0)4 50 87 78 64 – www.scaime.com

1 ENOD4 PRODUCT RANGE

1.1 General presentation

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

Quick and accurate:

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

Easy to integrate into automated system:

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

1.2 Versions

1.2.1 Communication protocol versions

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

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

1.2.2 IO+ version

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

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

1.3 eNodView Software

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

eNodView features and functions:

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

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

2 COMMUNICATION AND FONCTIONING MODES

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Functioning mode / Serial protocol AUX/USB	0x003E	0x64/10	0x00A8	R: 0x02E8 W: 0x02E9	0x2000 / 0x00	Uint	RW
HMI name	0x0034	0x64/21	0x00E0	/	0x3701 / 0x00	String	RW

2.1 Communication protocols Modbus RTU and SCMBus

Modbus RTU, SCMBus, and fast SCMBus communication protocols are accessible through AUX, USB. Modbus RTU or Profibus only depending on version on DB9 connection.
The protocol can be changed via the « Functioning mode/ serial protocol » register (see below).

bits b9b8	Protocol
00	SCMBus
01	Modbus RTU
11	Fast SCMBus

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

2.2 Functioning mode

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

bits b_1b_0	Functioning mode				
	eNod4-T	eNod4-C	eNod4-D	eNod4-F	eNod4-B
00	Transmitter	Transmitter	Transmitter	Transmitter	Transmitter
01	/	Checkweigher transmitter on request	Dosing by filling	Dosing	Belt scale
10	/	/	Dosing by unfilling	/	Belt weigh feeder

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

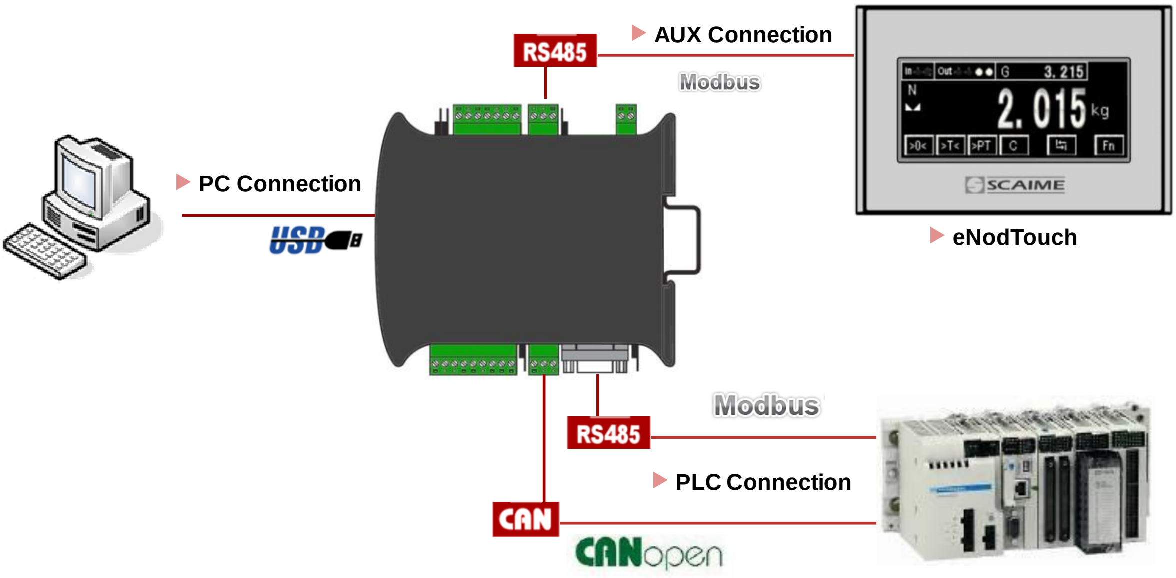
2.3 HMI name

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

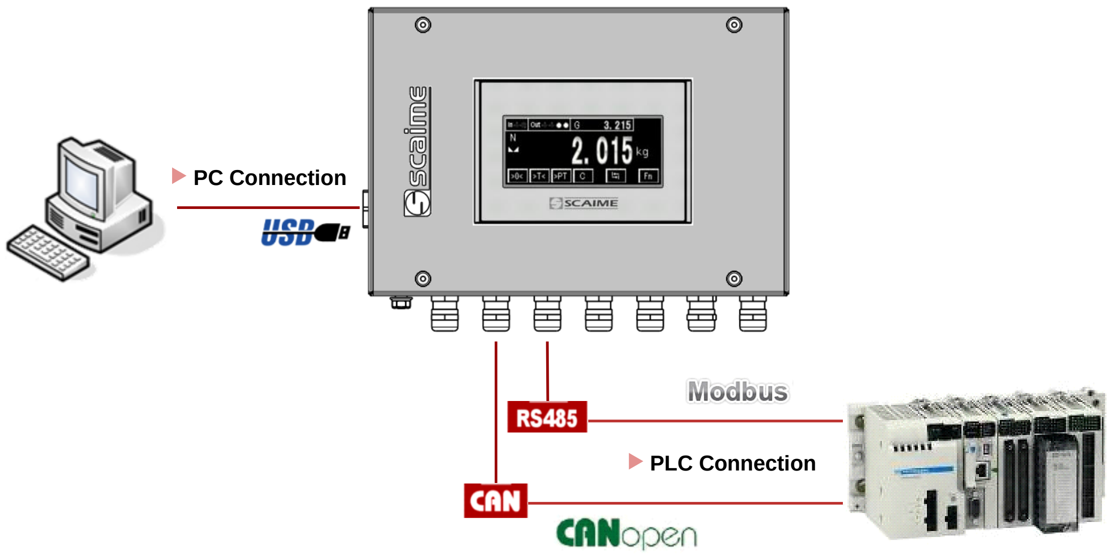
2.4 Simultaneous functioning of communications

2.4.1 Standard version

- **DIN Version**



- **BOX Version**

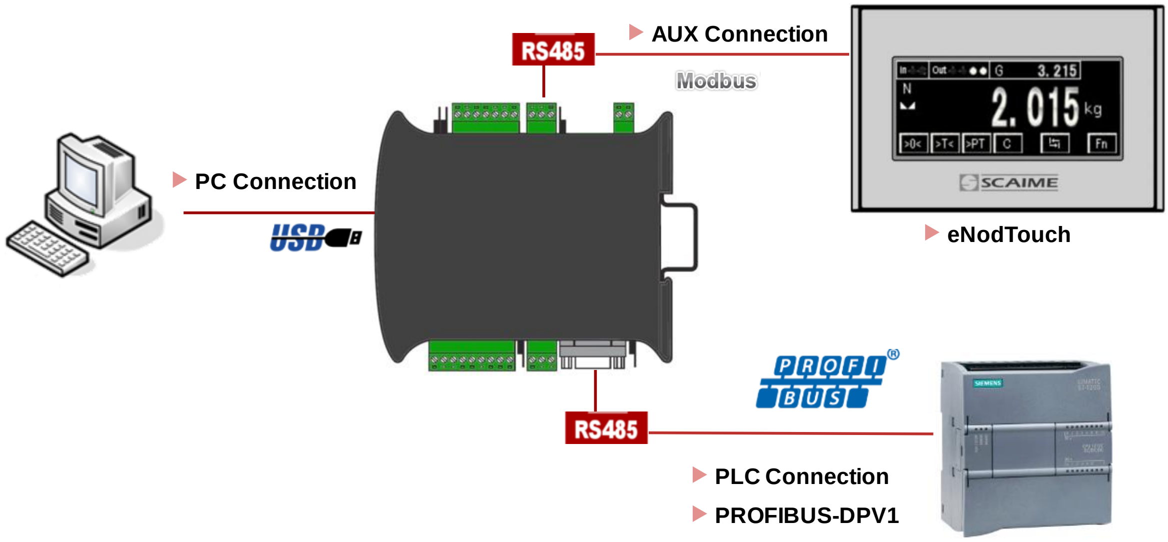


Simultaneous Communication	*RS485 PLC*	*RS485 AUX*	*CAN*
USB	*Yes**	*No*	*Yes**
RS485 PLC		*Yes*	*No*
RS485 AUX			*Yes**

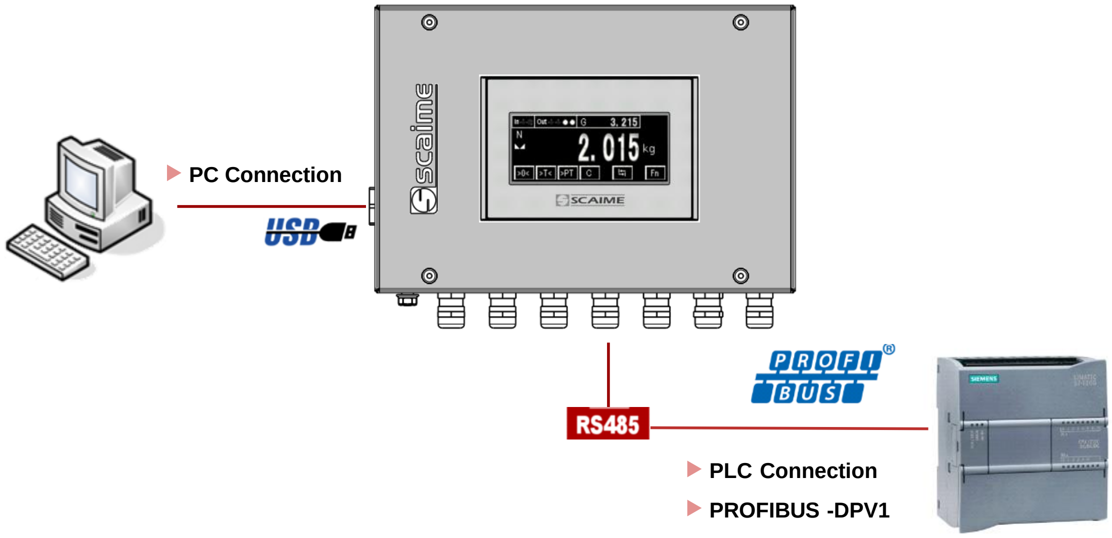
(*)Simultaneous use of CAN or RS485 PLC communication with USB or RS485 AUX can reduce performance of this interface.

2.4.2 Profibus version

- **DIN Version**



- **BOX Version**

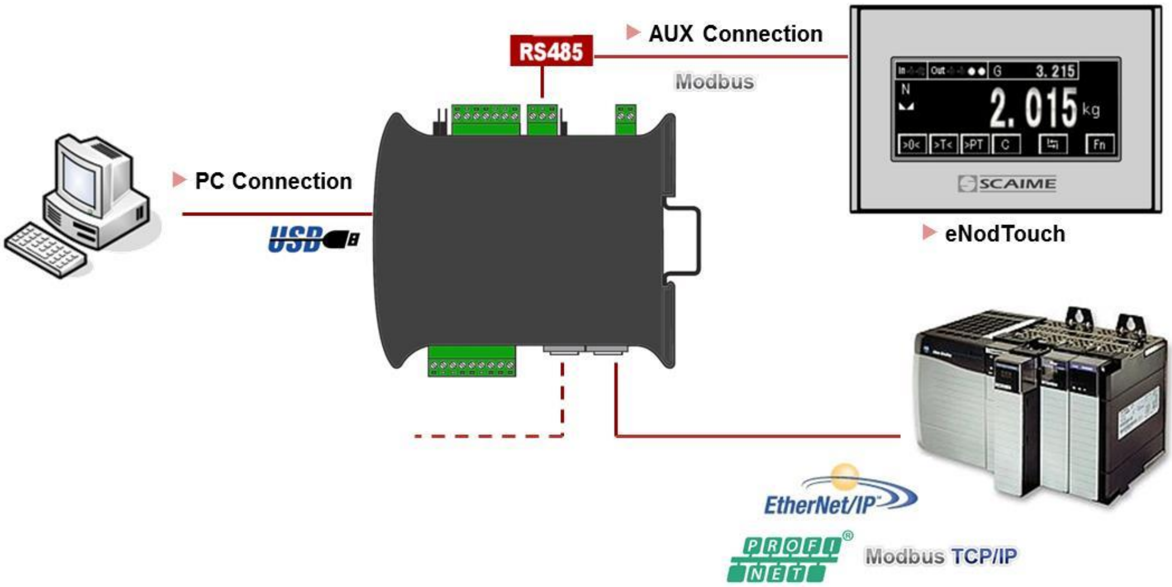


Simultaneous Communication	*Profibus*	*RS485 AUX*
USB	*Yes**	*No*
Profibus		*Yes**

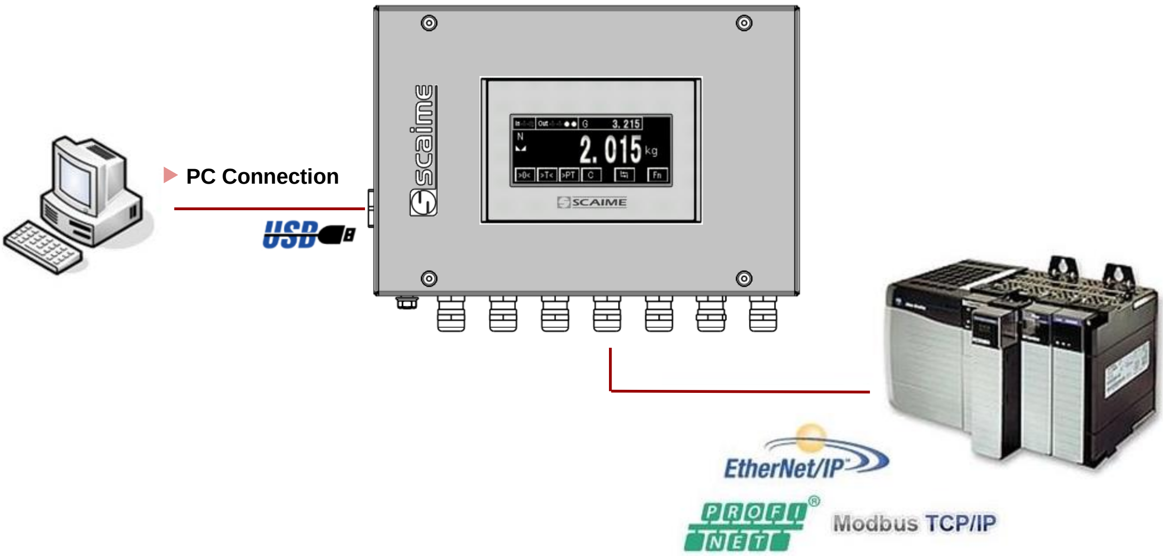
(*)Simultaneous use of Profibus with USB or RS485AUX can reduce performance of this interface.

2.4.3 Ethernet versions

- **DIN Version**



- **BOX Version**



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

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

3 MODBUS RTU

3.1 Physical interfaces

Modbus RTU communication protocol can be used either through ***eNod4*** USB port, AUX port. Modbus RTU or Profibus only depending on version on DB9 connection.
USB port behaves as a full duplex interface whereas the DB9 and AUX ports support half-duplex RS485 communication. Supported baud rates are 9600, 19200, 38400, 57600, and 115200.
For a complete description of the recommendations about ***eNod4*** RS485 connection, please refer to the user manual "characteristics and functioning" of the ***eNod4***.
Note: using ***eNod4*** through USB requires installing first the necessary USB drivers available on the website http://www.scaime.com.

3.2 Byte format

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

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

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

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

(cf. CRC-16 calculation algorithm).

3.3 Modbus RTU supported functions

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

Function	Code
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***.

3.4 Frames structure

During a read or write transaction, the two bytes of a register are transmitted MSB first then LSB.
If a data is coded on **4 bytes** (that means it requires two registers), **the two LSB are stored in the low address register and the two MSB are stored in the high address register.**

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

- request command sent to the slave :

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

- slave response :

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)

3.4.2 Function (06H) – 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*

3.4.3 Function (10H) – 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*

3.4.4 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*
02_H	*illegal data address*	*register address requested out of eNod4 register table*
03_H	*illegal data value*	*forbidden data values for the requested register*
04_H	*eNod4 not ready*	*eNod4 is not ready to answer (for example measurement request during a taring operation)*

3.5 Address and Baud rate

Address Modbus RTU	Meaning	Access	Type
0x0001	*Address and Baud rate*	*RO*	*Uint*

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

3.6 Product identification

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

Address Modbus RTU	Meaning	Access	Type
0x0000	*SW and product version*	*RO*	*Uint*

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

3.7 Measurement transmission

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

3.8 EEPROM error management

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

4 SCMBUS / FAST SCMBUS

4.1 Physical interfaces

SCMBus and fast SCMBus communication protocols can be used either through ***eNod4*** USB port and AUX port.
USB port behaves as a full duplex interface whereas the DB9 and AUX ports support half-duplex RS485 communication. Supported baud rates are 9600, 19200, 38400, 57600, and 115200.
For a complete description of the recommendations about ***eNod4*** RS485 connexion, please refer to the user manual "characteristics and functioning" of the ***eNod4***.
Note : using ***eNod4*** through USB requires installing first the necessary USB drivers available on the website http://www.scaime.com.

4.2 SCMBus and fast SCMBus features

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

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

4.3 Byte format

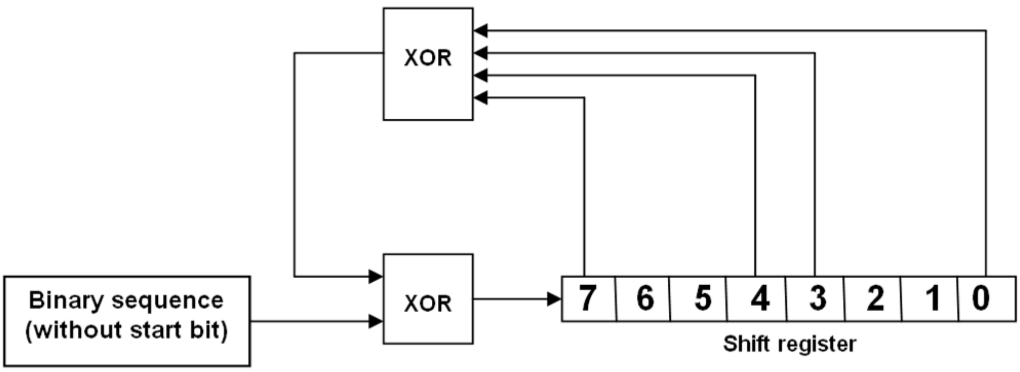
Data transmitted to ***eNod4*** thanks to SCMBus or fast SCMBus communication protocol must respect following format:

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

in SCMBus protocol, data is encoded as ASCII numeral characters (30_H ..... 39_H) and ASCII hexadecimal characters ($3A_H$ ..... $3F_H$).
in fast SCMBus protocol, data is encoded as signed hexadecimal (see frame structure paragraph) below.
SCMBus CRC-8 byte is generated by the following polynomial:

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

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



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

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

4.4 Frames structure

4.4.1 Transmission organization

- frame : ***eNod4*** address first

- byte : lsb first
- multi-bytes data : MSB first

4.4.2 Reading request

- request

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

- SCMBus response

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

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

- Fast SCMBus response

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

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

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

- request :

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

- response (SCMBus and fast SCMBus) :

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

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

4.4.4 Error frame

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

- response (SCMBus and fast SCMBus) :

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

- The error codes are listed below:

Error code	Meaning	Description
FE$_H$	*unknown command*	*requested command is not supported by **eNod4***
FF$_H$	*error during command execution*	*ex. : tare when gross meas.<0*

4.5 Address and Baud rate

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

4.6 Product identification

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

4.7 Measurement transmission

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

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

4.8 Continuous transmission

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

Address SCMBus	*Description*	*Accès*	*Type*
0x003F	*SCMBus Measurement transmission period*	*RW*	*Uint*

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 transmission	*$E1_H$*
start brut measurement transmission	*$E2_H$*
stop continuous transmission	*$E3_H$*

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.

4.9 EEPROM error management

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

5 MODBUS TCP

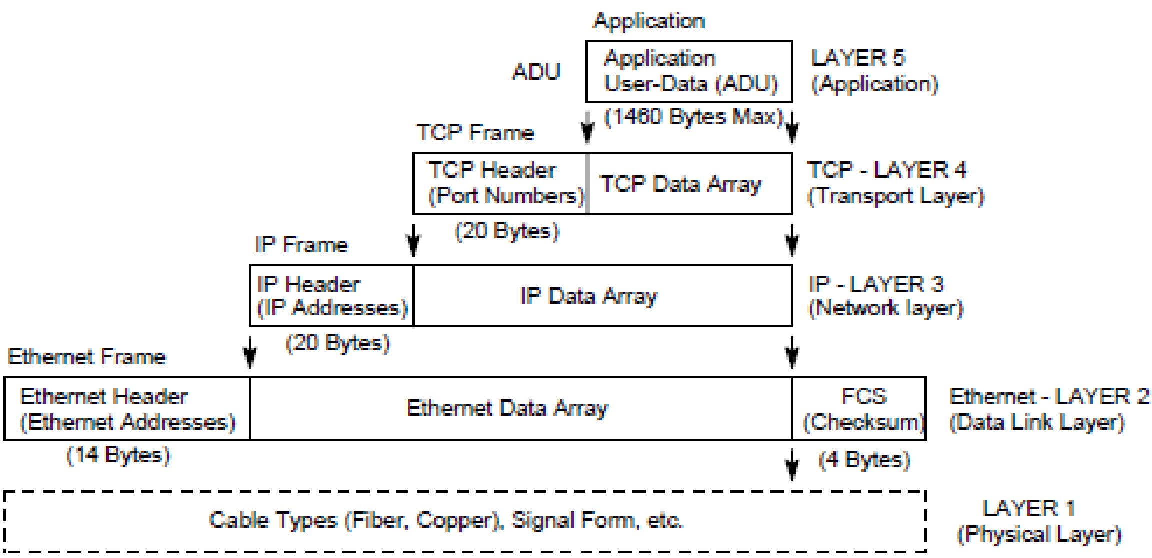
When a configuration change occurs (change of Ethernet parameters, set default params via eNodView or eNodTouch) eNod4 Modbus-TCP absolutely must not be reset or power cycled within 10 seconds after send of the change. This could permanently damage the eNod. MS LED blinks green or red cyclically when in this "damaged" state.

5.1 Physical interface

eNod4 is fitted with an Ethernet interface on RJ45 connectors and is galvanically isolated.
The Auto-Crossover function is supported. Due to this fact the signals RX and TX may be switched on ETH1 and ETH2 interfaces.
Because Modbus TCP (or Modbus TCP/IP) shares the same physical and data link layers of traditional IEEE 802.3 Ethernet, physical interface remains fully compatible with the already installed Ethernet infrastructure of cables, connectors, network interface cards, hubs, and switches.
eNod4 allows topologies in tree, line or star network. It also allows ring-shaped topology since **RSTP (Rapid Spanning Tree Protocol)** has been implemented (***eNod4*** is a simple node and cannot act as network supervisor).
Every ***eNod4*** drives two Ethernet ports and has an internal switch and hub functions, respectively the different circuits which are related to the special features of some Real-Time-Ethernet systems to build up a line structure.

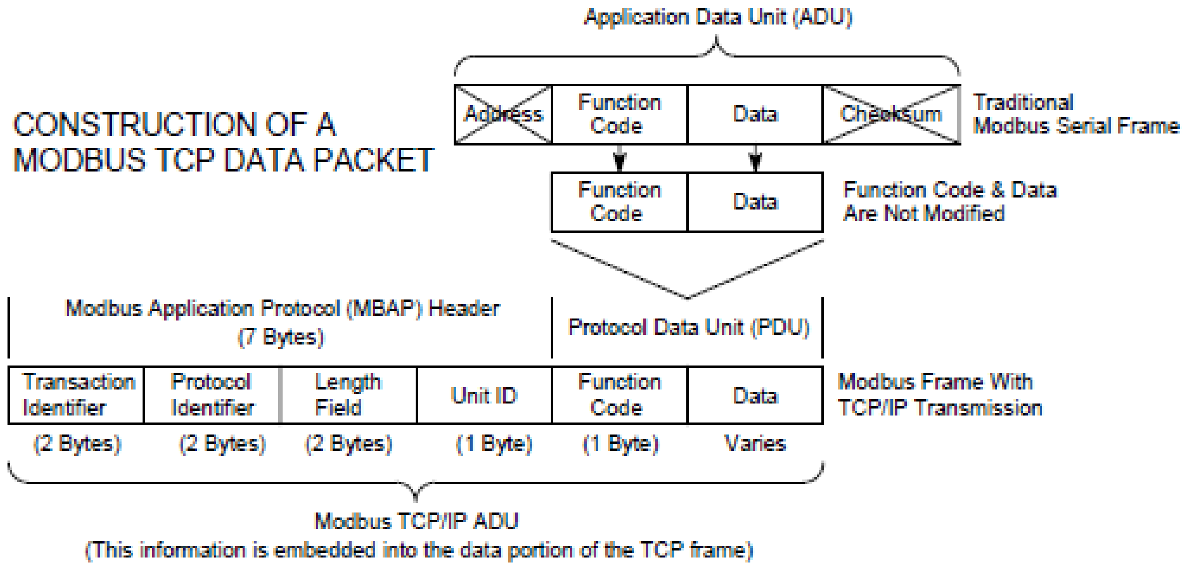
5.2 General information

eNod4 is fitted with an Ethernet communication interface that supports protocols TCP (Transmission Control Protocol) and IP (Internet Protocol). These protocols are used together and are the main transport protocol for the internet.
When Modbus information is sent using these protocols, the data is encapsulated by TCP where additional information is attached and given to IP. IP then places the data in a packet (or datagram) and transmits it on Ethernet network.
Construction of a Modbus TCP data packet and simplified OSI model communication layers representation:



TCP must establish a connection before transferring data, since it is a connection-based protocol.
The Master (or Client in Modbus TCP) establishes a connection with the Slave (or Server) ***eNod4***. The Server ***eNod4*** waits for an incoming connection for the Client. Once a connection is established, the Server ***eNod4*** then responds to the queries from the Client until the Client closes the connection.
Modbus TCP/IP uses well-known specific port 502 to listen and receive Modbus messages over Ethernet.
Note: ***eNod4*** does not support **Modbus RTU over TCP** protocol (simply put, this is a Modbus RTU message transmitted with a TCP/IP wrapper and sent over a network instead of serial lines).

eNod4 supports **Modbus TCP (or Modbus TCP/IP)** protocol: a document ***Modbus Messaging on TCP/IP implementation guide*** provided by Schneider Automation outlines a modified protocol specifically for use over TCP/IP. The official Modbus specification can be found at ***Modbus organization*** (www.modbus.org).
ADU (Application Data Unit) and PDU (Protocol Data Unit): aside from the main differences between serial and network connections stated above, there are few differences in the message content between Modbus TCP and Modbus RTU.
Starting with Modbus RTU frame (**ADU**), the checksum disappears. From now on data integrity is granted by Ethernet Data Link layer. Slave ID address is suppressed and supplanted by an identifier (Unit ID) that is a part of a complementary data header called **MBAP** (Modbus Application Protocol) header. The MBAP header is 7 bytes long.



MBAP header: fields are defined below:

fields	*Length (bytes)*	*Description*	*Client (Master)*	*Server (Slave)*
Transaction Identifier	*2*	*Transaction pairing (request / response Modbus)*	*Initiated by the Client*	*Echoed back by the Server*
Protocol Identifier	*2*	*0 = MODBUS Protocol*	*Initiated by the Client*	*Echoed back by the Server*
Length	*2*	*byte count of the remaining fields (Unit ID + Function Code + Data)*	*Initiated by the Client (request)*	*Initiated by the Server (response)*
Unit Identifier	*1*	*Idendification of a remote server (non TCP/IP or other buses), 0x00 or 0xFF otherwise*	*Initiated by the Client*	*Echoed back by the Server*

Supported functions: identical to Modbus RTU ones.

- Read multiple registers* : **03_H / 04_H**
- Write single register* **06_H**
- Write multiple registers* **10_H**

*1 register = 2 bytes
Maximal number of registers = 123

5.3 Frames structure

- By default and as in Modbus RTU, during a read or write transaction, the two bytes of a register are swapped. The MSB is transmitted first and then the LSB. However it may be possible using ***eNodView*** software to invert the swapping of data in a register.
- 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 Modbus RTU request command example sent to the slave in hexadecimal:

Slave address	*03_H or 04_H*	*First register address*	*N registers*	*CRC16*
1 byte	1 byte	2 bytes	2 bytes	2 bytes
11	03	00 7D	00 03	97 43

- Equivalent request in Modbus TCP:

Transaction Identifier	*Protocol Identifier*	*Message length*	*Unit Identifier*	*03_H or 04_H*	*First register address*	*N registers*
2 bytes	2 bytes	2 bytes	1 byte	1 byte	2 bytes	2 bytes
00 01	00 00	00 06	FF	03	00 7D	00 03

Modbus exception codes: like in Modbus RTU a server ***eNod4*** may generate an exception response to a client request.

- Exception codes table:

Error code	*Exception*	*Description*
01	Illegal Function	The function code received by ***eNod4*** in the query is not allowed or invalid.
02	Illegal Data Address	The data address received in the query is not an allowable address for ***eNod4*** or is invalid.
03	Illegal Data Value	A value contained in the query data field is not an allowable value or out of the limits
06	***eNod4*** Device Busy	***eNod4*** is not ready to answer (for example measurement request during a taring operation).

5.4 Network configuration

Every ***eNod4*** is identified on the network by an IP address, a subnet mask and a default gateway address. Network configuration can only be set using ***eNodView*** software at minimum version V.

IP address: the IP address is comprised of two parts: the network address or Net ID (first part), and the host address or Host ID (last part). This last part refers to a specific machine on the given sub-network identified by the first part. The numbers of bytes of the total four that belong to the network address depend on the Class definition (Class A, B, or C) and this refers to the size of the network.

Class C subnets share the first 3 octets of an IP address, giving 254 possible IP addresses for ***eNod4*** device. Recall that the first 00_H and last FF_H IP addresses are always used as a network number and broadcast address respectively.

eNod4 default local IP* address is **192.168.0.100**

**if IP static configuration set*

Subnet mask: a Subnet Mask is used to subdivide the host portion of the IP address into two or more subnets. The subnet mask will flag the bits of the IP address that belong to the network address, and the remaining bits correspond to the host portion of the address.

The unique subnet to which an ***eNod4*** IP address refers to is recovered by performing a bitwise AND operation between the IP address and the mask itself, with the result being the sub-network address.

eNod4 subnet mask default value is the default Class C subnet mask **255.255.255.0**

Gateway address: a gateway is being used to bridge Ethernet to other networks like a serial sub-network of Modbus RTU devices in order to provide communication compatibility.

The IP address of the default gateway has to be on the same subnet as the local IP address. The value 0.0.0.0 is forbidden. If no gateway is to be defined then this value is to be set to the local IP address of the ***eNod4*** device.

Default gateway address has been set to **192.168.0.254**

DHCP functionality (Dynamic Host Configuration Protocol):

It's a protocol that automates network-parameter assignment and allows an ***eNod4*** device to dynamically configure (without any particular action) an IP address and other information that is needed for network communication.

eNod4 device needs imperatively to be connected on the sub-network to a DHCP server that allocates IP address and also DHCP functionality has to be activated in ***eNod4*** device.

A label affixed on every ***eNod4*** contains 6 bytes of its MAC address (Media Access Control Address) which is a unique identifier assigned to network interfaces for communications on any physical network segment.

In DHCP when the Master of the sub-network attributes an IP address to a Slave (***eNod4*** device), it associates its unique MAC address to the IP address. So the MAC address is the only way for a Master to identify an ***eNod4*** device on the sub-network.
DHCP functionality is not activated by default (set to **static IP configuration**).

5.5 Modbus TCP LED

State of the **NS (Network Status)** bicolor LED is described in the table below:

Color	*State*	*Meaning*
Green	*Blinking 1Hz*	*Device **READY** but not **CONFIGURED** yet*
	Blinking 5Hz	*Device **WAITING** for communication*
	Always on	***CONNECTED** (at least one TCP connection is established)*
Red	*Blinking 2Hz (On/Off rate 25%)*	*Internal Fault detect (like TCP connection lost)*
	Always on	*Communication fatal error*
-	*Always off*	*Device not powered or defective*

State of the **MS (Module Status)** bicolor LED is described in the table below:

Color	State	Meaning
Green	*Blinking*	*Device **WAITING FOR CONFIGURATION***
	Always on	*Device is **OPERATING** correctly*
Red	*Blinking*	*Communication error detected*
	Always on	*Fatal error detected*
Red / Green	*Blinking*	*Autotest at power on*
-	*Always off*	*Device not powered or defective*

State of the **ACT / LINK** ETH1 and ETH2 network RJ45 connector LED:

Color	State	Meaning
LINK *(Eth1 & Eth2) Green*	*Always on*	*A physical connection to the Ethernet exist*
	Always off	*Device not connected to the Ethernet*
ACT *(Eth1 & Eth2) Yellow*	*On*	*The device sends/receives Ethernet frames*
	Always off	*No traffic on the Ethernet*

5.6 Modbus TCP I/O scanning

The exchange of application data at a high refreshment rate is only possible in a specific range of Modbus addresses. Specified 47 Input registers that are exchanged in I/O scanning are defined in the table below:

Register address (Hex)	Size in bytes (n)	Type	Name	Access
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	*4*	*Float*	*instant flow rate*	*RO*
0088	*4*	*Float*	*average flow rate*	*RO*
008A	*4*	*Float*	*Flow rate control output*	*RO*
008C	*2*	*Uint*	*Control output value*	*RO*
008D	*4*	*Ulong*	*Totalizer value (Great WU)*	*RO*
008F	*2*	*Uint*	*Complementary totalizer value*	*RO*
0090	*2*	*Uint*	*command register*	*R/W*
0091	*2*	*Uint*	*response register*	*RO*
0092	*4*	*long*	*Zero offset*	*R/W*
0094	*2*	*Uint*	*Logical I/O level*	*RO*
0095	*4*	*Float*	*Dosing weight deviation*	*RO*
0097	*4*	*long*	*Preset tare*	*RO*
0099	*2*		***Réservé***	
009A	*2*	*Uint*	*Dosing status*	*RO*
009B	*2*	*Uint*	*Dosing errors report*	*RO*
009C	*2*	*Uint*	*Dosing errors counter*	*RO*
009D	*2*	*Uint*	*Last dosing error*	*RO*
009E	*4*	*Float*	*Dosing quality factor*	*RO*
00A0	*4*	*Float*	*Totalization flow rate*	*RO*
00A2	*4*	*Ulong*	*Grand total (Great WU)*	*RO*
00A4	*4*	*Ulong*	*General total (Great WU)*	*RO*
00A6	*4*	*Float*	*Extraction time*	*RO*
00A8	*4*	*Ulong*	***eNod4** 1ms counter**	*RO*

**for possible check of the performances*

6 ETHERNET/IP

When a configuration change occurs (change of Ethernet parameters, set default params via eNodView or eNodTouch, change of address « Name of product » after a reset with option « Use rotary switch in product name ») eNod4 EtherNet/IP absolutely must not be reset or power cycled within 10 seconds after send of the change or reset. This could permanently damage the eNod. MS LED blinks green cyclically when in this "damaged" state.

EtherNet/IP uses Ethernet layer network infrastructure. It is built on the TCP (Transmission Control Protocol) and IP (Internet Protocol) protocols, but the "IP" in the name stands for "**Industrial Protocol**" and not an abbreviation for "Internet Protocol". EtherNet/IP is supported by four independent networking organizations

- ControlNet International (CI),
- The Industrial Ethernet Organization (IEA),
- The Open DeviceNet Vendor Association (ODVA),
- The Industrial Automation Open Network Alliance (IAONA).

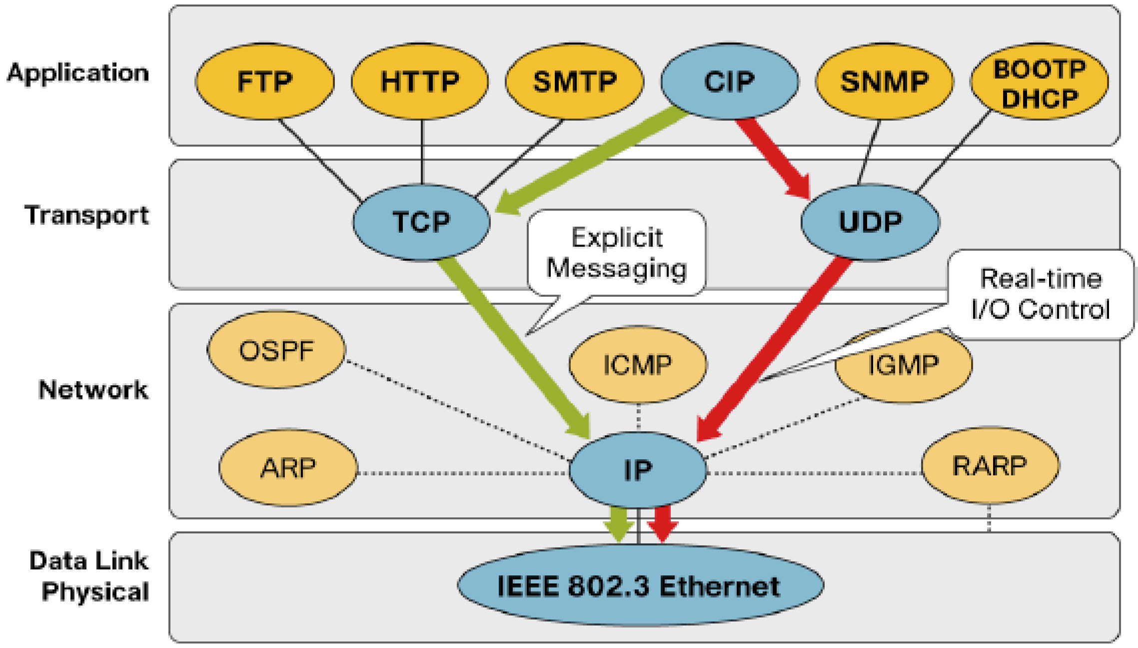
6.1 Physical interface

eNod4 is fitted with two EtherNet ports on RJ45 connectors that are galvanically isolated.
The Auto-Crossover function is supported. Due to this fact the signals RX and TX may be switched on ETH1 and ETH2 interfaces. Auto-negotiation of link parameters applies to 10/100Mbit and full/half duplex operation.
Because EtherNet/IP shares the same physical and data link layers of traditional IEEE 802.3 Ethernet, physical interface remains fully compatible with already installed Ethernet infrastructure (cables, connectors, network interface cards, hubs, and switches).
EtherNet/IP automatically benefits from all further technology enhancements such as Gigabit Ethernet and Wireless technologies.
Tree, line or star network topologies are allowed by ***eNod4***. Ring topology is also supported while Device Level Ring (DLR) protocol is implemented (as ***eNod4*** is not able to act as a ring supervisor, at least one active ring supervisor is required on the DLR network).
Every ***eNod4*** drives two Ethernet ports and has an internal switch and hub functions, respectively the different circuits which are related to the special features of some Real-Time-Ethernet systems to build up a line or ring structure.

6.2 General information

6.2.1 EtherNet/IP "Open standard" protocol

EtherNet/IP shares the same lower four layers of the OSI model common to all Ethernet devices. This makes it fully compatible with existing Ethernet hardware, such as cables, connectors, network interface cards, hubs, and switches.
The application layer protocol is the Control and Information Protocol (CIP™).
eNod4 is fitted with an Ethernet communication interface that supports protocols **TCP** (Transmission Control Protocol), **UDP** (User Datagram Protocol) and **IP** (Internet Protocol). These protocols are used together and are the main transport protocol for the internet. When **CIP™** information is sent using these protocols, the data is encapsulated by TCP or UDP where additional information is attached and given to IP. IP then places the data in a packet (or datagram) and transmits it on **Ethernet** network.



By using TCP/IP, EtherNet/IP is able to send explicit messages, which are used to perform **client-server** type transactions between nodes. Nodes must interpret each message, execute the requested task and generate responses. Uploading and downloading of configuration data like setpoints and applicative parameters uses **explicit (or Class 3) messaging**.
TCP is connection-oriented and use well known TCP port number 44818 (0xAF12) for EtherNet/IP.
For real-time messaging, EtherNet/IP also employs UDP over IP, which allows messages to be unicast (one to one) or multicast (one to a group of destination addresses) in a **producer-consumer** model. This is how CIP™ I/O data transfers called **implicit (or Class1) messaging** is sent on EtherNet/IP. With implicit messaging, the data field contains no protocol information, only real-time I/O data. Since the meaning of the data is pre-defined at the time the connection is established, processing time is minimized during runtime. UDP is connectionless and makes **no guarantee that data will get from one device to another**; however, UDP messages are smaller and can be processed more quickly than explicit messages. As a result, EtherNet/IP uses UDP/IP to transport I/O messages that typically contain time-critical control data. The CIP™ Connection mechanism provides timeout mechanisms that can detect data delivery problems, a capability that is essential for reliable control system performance.
UDP port used is port 2222 (0x08AE).

TCP/IP/MAC Encapsulation (Explicit Messaging)

Ethernet Header (14 Bytes)	IP Header (20 Bytes)	TCP Header (20 Bytes)	Encapsulation Message(s)	CRC

UDP/IP/MAC Encapsulation (Implicit Messaging)

Ethernet Header (14 Bytes)	IP Header (20 Bytes)	UDP Header (8 Bytes)	Encapsulation Message	CRC

The process of opening a connection is called Connection Origination, and the node that initiates the connection establishment request is called a Connection Originator, or just an **Originator** (so called Scanner). Conversely, the node that responds to the establishment request is called a Connection Target, or a **Target** (so called Adapter).

6.2.2 Common Industrial Protocol (CIPTM)

Common Industrial Protocol (CIP™) has implementations based upon Ethernet with EtherNet/IP, but also through DeviceNet (CIP™ over CAN bus) and ControlNet (CIP™ over a dedicated network).
Most controllers (with appropriate network connections) can transfer data from one network type to the other, leveraging existing installations, yet taking advantage of Ethernet.
CIP™ is an object oriented protocol. Each CIP™ **object** has **attributes** (data), **services** (commands) and **behaviors** (reactions to events). Objects are also named **classes**. An object **instance** refers to one implementation of a class. Each instance of a class has the same attributes, but its own particular set of attribute values.
We use attributes to refer to the data of an object. You use methods to operate on the data. Every attribute of an object will have a corresponding method and you invoke a method by sending a service to it. Services are the

communication mechanism between objects. CIP™ object models will use "**get**" and "**set**" messages as the methods to access their data.
The behavior of an object is what the object can do and this behavior is contained within its methods.
An integer ID value is assigned to each object **class**, each **instance** of the same class, each class **attribute** and each class **service.** There is only one assigned instance for ***eNod4*** application-specific classes.
CIP™ provides many standard services for control of network devices and access to their data via implicit and explicit messages. The key thing to remember about implicit messages is that there can be many consumers of a single network packet and this requires UDP, while TCP is instead reserved for point-to-point messages.
CIP™ also includes "device types" for which there are "device profiles". ***eNod4*** does not follow any device profile because functionality is specific. CIP™ already includes a large collection of commonly defined objects or object classes and only two objects referring to Ethernet, TCP/IP Interface Object & Ethernet Link Object.
Additional ***eNod4***-specific objects (EtherNet/IP-compliant) have been defined in order to support the functional requirements of particular applications.
eNod4 EtherNet/IP devices supports the following ODVA commonly defined objects:

- An Identity Object (ID 0x01 class),
- A Connection Manager Object (ID 0x06 class),
- A TCP/IP Interface Object (ID 0xF5 class),
- An Ethernet Link Object (ID 0xF6 class),
- A DLR Object (ID 0x47 class),
- A Quality of Service Object (ID 0x48 class).

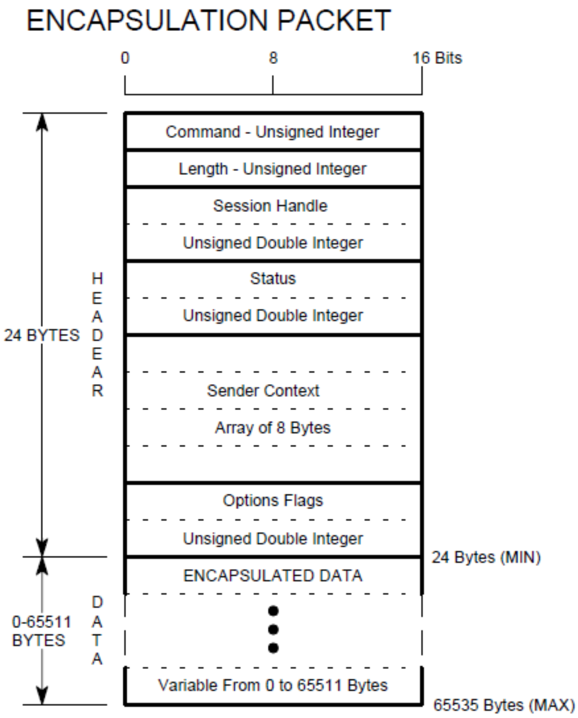
eNod4 application-specific objects are defined below:

- A Metrology and Identification Object (ID 0x64 class),
- A Calibration Object (ID 0x65 class),
- A Filtering Object (ID 0x66 class),
- A Logical Inputs/Outputs Object (ID 0x67 class),
- A Command / Response Object (ID 0x68 class).

Corresponding Class Attributes and Services supported are described in **Appendix**.

6.2.3 CIPTM Encapsulation Format

The CIP™ Encapsulation Message (the data portion of the TCP or UDP frame) includes a 24 byte header followed by its own data (optional) and is limited to a total length of 65535 bytes. This packet takes the following format:



For any data to exchange, the encapsulated data format is **most significant bit (MSB) transmitted first.**
Access to the object model of a device is controlled by one of two objects: the Connection Manager, and the UnConnected Message Manager (**UCMM**). We have already stated that EtherNet/IP is a connection-based network

and that most CIP™ messages are accomplished through connections. CIP™ also allows multiple connections to coexist in a device at any given time.
eNod4 allows up to 4 simultaneous EtherNet/IP connections (sum of explicit and implicit connections).
In addition, it is not possible on the same module to access to different device application-specific Class for multiple explicit connections. For implicit connection, ***eNod4*** accepts 1 exclusive owner and up to 2 listener only.
eNod4 supports only cyclic connection CIP™ trigger.

6.3 Network configuration

Every ***eNod4*** is identified on the network by an IP address, a subnet mask and a default gateway address. Network configuration can only be set using ***eNodView*** software at minimum version V.
IP address: the IP address is comprised of two parts: the network address or Net ID (first part), and the host address or Host ID (last part). This last part refers to a specific machine on the given sub-network identified by the first part. The numbers of bytes of the total four that belong to the network address depend on the Class definition (Class A, B, or C) and this refers to the size of the network.
Class C subnets share the first 3 octets of an IP address, giving 254 possible IP addresses for ***eNod4*** device. Recall that the first 00_H and last FF_H IP addresses are always used as a network number and broadcast address respectively.
eNod4 default local IP* address is **192.168.0.100**
*****if IP static configuration set
Subnet mask: a Subnet Mask is used to subdivide the host portion of the IP address into two or more subnets. The subnet mask will flag the bits of the IP address that belong to the network address, and the remaining bits correspond to the host portion of the address.
The unique subnet to which an ***eNod4*** IP address refers to is recovered by performing a bitwise AND operation between the IP address and the mask itself, with the result being the sub-network address.
eNod4 subnet mask default value is the default Class C subnet mask **255.255.255.0**
Gateway address: a gateway is being used to bridge Ethernet to other networks like a serial sub-network of Modbus RTU devices in order to provide communication compatibility.
The IP address of the default gateway has to be on the same subnet as the local IP address. The value 0.0.0.0 is forbidden. If no gateway is to be defined then this value is to be set to the local IP address of the ***eNod4*** device.
Default gateway address has been set to **192.168.0.254**
DHCP functionality (Dynamic Host Configuration Protocol):
It's a protocol that automates network-parameter assignment and allows an ***eNod4*** device to dynamically configure (without any particular action) an IP address and other information that is needed for network communication.
eNod4 device needs imperatively to be connected on the sub-network to a DHCP server that allocates IP address and also DHCP functionality has to be activated in ***eNod4*** device.
A label affixed on every ***eNod4*** contains 6 bytes of its MAC address (Media Access Control Address) which is a unique identifier assigned to network interfaces for communications on any physical network segment.
In DHCP when the Master of the sub-network attributes an IP address to a Slave(***eNod4*** device), it associates its unique MAC address to the IP address. So the MAC address is the only way for a Master to identify an ***eNod4*** device on the sub-network.
DHCP functionality is not activated by default (set to **static IP configuration**).

6.4 EtherNet/IP LED

State of the **NS (Network Status)** bicolor LED is described in the table below:

Color	State	Meaning
Green	*Blinking*	***NO CONNECTIONS**: device has no connections established, but has obtained an IP address*
	Always on	***CONNECTED** (at least one connection is established)*
Red	*Blinking*	***CONNECTION TIMEOUT**: one or more of the connections in which this device is a target has timed out. This shall be left only if all timed out connections are reestablished or if the device is reset.*
	Always on	***DUPLICATE IP**: the device has detected that its IP address is already in use*
Red / Green	*Blinking*	*Autotest at power on*
-	*Always off*	*Device not powered or defective*

State of the **MS (Module Status)** bicolor LED is described in the table below:

Color	State	Meaning
Green	*Blinking*	***STANDBY**: the device has not been configured*
	Always on	***DEVICE OPERATIONAL**: Device is operating correctly*
Red	*Blinking*	***MINOR FAULT**: the device a detected a recoverable minor fault*
	Always on	***MAJOR FAULT**: the device a detected a non-recoverable major fault*
Red / Green	*Blinking*	*Autotest at power on*
-	*Always off*	*Device not powered or defective*

State of the **ACT / LINK** ETH1 and ETH2 network RJ45 connector LED:

Color	State	Meaning
LINK *(Eth1 & Eth2)* *Green*	*Always on*	*A physical connection to the Ethernet exist*
	Always off	*Device not connected to the Ethernet*
ACT *(Eth1 & Eth2)* *Yellow*	*On*	*The device sends/receives Ethernet frames*
	Always off	*No traffic on the Ethernet*

6.5 I/O scanning / implicit messaging

eNod4 Target (Adapter) depending version could consumes up to two registers (4 bytes without header) of Output data (from the network's point of view) through Assembly Instance 0x64 (100) with a Cyclic transport trigger type and point to point connection type.
With no IO+ option, data exchanged is the command register which is the attribute 1 of device application-specific 0x68 class.

With IO+ option, data's exchanged are the command register and an External value to control analog output which is the attribute 16 of device application-specific 0x67 class.

eNod4 produces Input data (from the network's point of view) through Assembly Instance 0x65 (101) with a Cyclic transport trigger type. Multicast or point to point connection type, connection rate, size and priority are defined when the connection is established by the Originator (Scanner) through the connection manager Object using the *Forward_open* Service (Connection is closed using the *Forward_close* Service).

6.5.1 Produced registers

Find in the table below the specified registers (**80 bytes** without header) that are produced through Assembly Instance 0x65 (101):

Register Modbus Address (Hex)	*Offset in bytes (without header)*	*Type*	*Name*
/	*0*	*long*	***eNod4** 1ms counter**
0094	*4*	*Uint*	*Input / Output levels*
007D	*6*	*Uint*	*Measurement status*
007E	*8*	*long*	*Gross measurement*
0080	*12*	*long*	*Tare value*
0082	*16*	*long*	*Net measurement*
0084	*20*	*long*	*Factory calibrated points*
0090	*24*	*Uint*	*Command register*
0091	*26*	*Uint*	*Response register*
0086	*28*	*Float*	*instant flow rate*
0088	*32*	*Float*	*average flow rate*
008A	*36*	*Float*	*Flow rate control output*
008C	*40*	*Uint*	*Control output value*
00A0	*42*	*Float*	*Totalization flow rate*
008D	*46*	*Ulong*	*Totalizer value (Great WU)*
008F	*50*	*Uint*	*Complementary totalizer value*
00A2	*52*	*Ulong*	*Grand total (Great WU)*
00A4	*56*	*Ulong*	*General total (Great WU))*
009A	*60*	*Uint*	*Dosing status*
009B	*62*	*Uint*	*Dosing errors report*
009C	*64*	*Uint*	*Dosing errors counter*
009D	*66*	*Uint*	*Last dosing error*
0095	*68*	*Float*	*Dosing weight deviation*
009E	*72*	*Float*	*Dosing quality factor*
00A6	*76*	*Float*	*Extraction time*

**for possible check of the performances*

6.5.2 Consumed registers

Find in the table below the specified register **(2 bytes without header)** that is consumed through Assembly Instance 0x64 (100):

Register Modbus Address (Hex)	*Offset in bytes (without header)*	*Type*	*Name*
0090	*0*	*Uint*	*Command register*

The register "Command register" uses the mechanism of ***eNod4*** functional commands defined in another chapter.
Note: "reset" and "Restore default settings" commands cannot be sent via cyclic and acyclic exchanges immediately after a restart of ***eNod4***. To be able to use these commands, it must first be processed another command ("cancel Tare" for example).
Note: The "Command register" data **must be** set to 0x0000 before each new command.

6.5.2.1 IO+ version

Find in the table below the specified register **(4 bytes without header)** that is consumed through Assembly Instance 0x64 (100):
Functional commands register works in same way whatever IO+ version or not.

Register Modbus Address (Hex)	*Offset in bytes (without header)*	*Type*	*Name*
0090	*0*	*Uint*	*Command register*
0032	*2*	*Uint*	*External value to control analog output*

External device (e.g. PLC) could drive ***eNod4*** analog output through the register ***external value to control analog output***. With IO+ option and when analog output is set to ***level on request*** function, ***eNod4*** will pilot the analog output in current or in voltage by the value of ***external value to control analog output*** variable. Analog output value is expressed in 0.01% of maximum current or voltage level.

7 PROFINET IO

When a configuration change occurs (change of Ethernet parameters, set default params via eNodView or eNodTouch, change of address «Name of the station » after a reset with option « Use rotary switch in name of the station») eNod4 Profinet absolutely must not be reset or power cycled within 10 seconds after send of the change or reset. This could permanently damage the eNod. MS LED blinks green cyclically when in this "damaged" state.

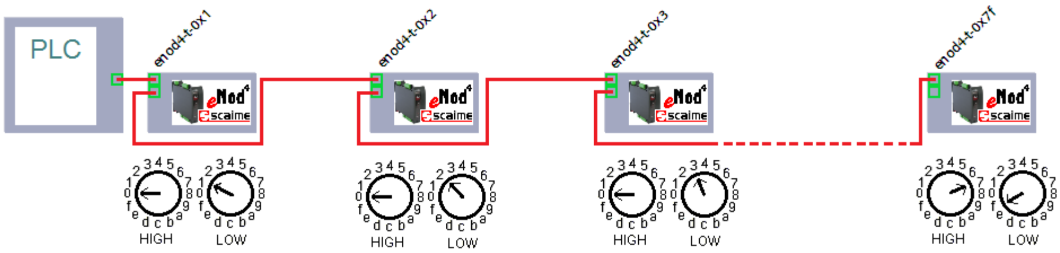
PROFINET is the communication standard created by the PROFIBUS International organization. It allows use of an industrial Ethernet network for real time data exchange between automation components. Whereas PROFINET CBA variant allows splitting intelligence of the application over network components, the PROFINET IO variant allows the exchange of I/O data between an IO-controller (e.g. PLC (Programmable Logic Controller)) that contains the intelligence of the application and IO-devices. ***eNod4 ETH Profinet*** is an IO-device and can exchange data only with one IO-controller.

7.1 Physical interface

eNod4 is fitted with two Ethernet ports on RJ45 connectors that are galvanically isolated. They support the switch or hub functions, specific functions of real time Ethernet systems and facilitate the implementation of line or ring topology.
The function of automatic crossing of emission line and reception line (Auto-Crossover Rx/Tx) on ETH1 and ETH2 interfaces is supported. Auto-negotiation of Ethernet link layer settings applies to the choice of the 10/100Mbit speed as well as Full or Half-Duplex operations.
As PROFINET IO communicates on Ethernet II type frames, ***eNod4*** is compatible with most of the existing network infrastructures (cards, connectors, network, hub and switches).
Each ***eNod4*** has a hardware **MAC** address (Media Access Control address). A label affixed to each ***eNod4*** includes the 6-bytes MAC address. It is a unique identifier of any Ethernet network hardware.

7.2 Network settings

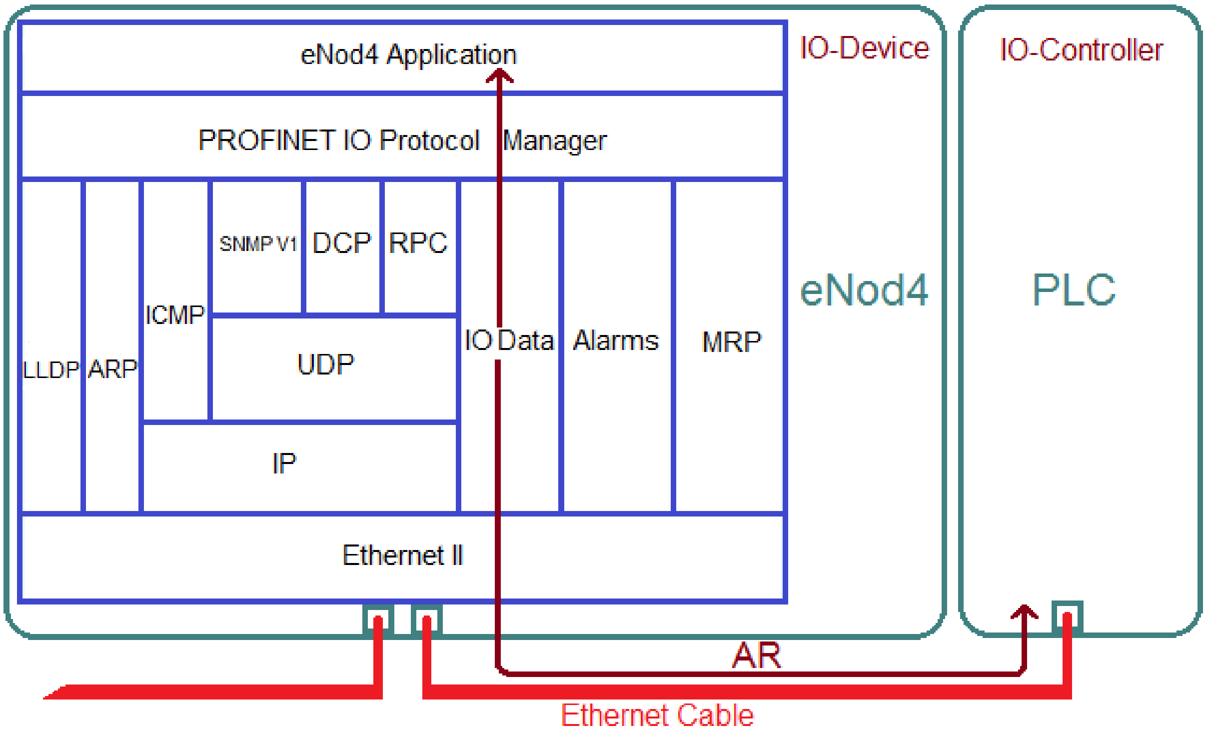
All PROFINET IO network settings and options are configurable using the eNodView software to V version minimum.
IP settings: IP address, subnet mask and default gateway. Default values of these parameters are (192.168.0.100, 255.255.255.0, 192.168.0.254). Configuration of these settings via eNodView is of little interest. Usually it is the IO-Controller which assigns to each IO-Device its IP settings using the name of the station.
Name of the station: The name of the station is the primary key that allows the identification of the PROFINET IO node. So, it must be unique for each node on PROFINET IO subnet. It can only contain lowercase characters, figures, dashes and dots. The default value of this parameter is based upon (configurable option) the rotary switches located in front of ***eNod4***. It is set to:
"enod4-t-0x'address_on_rotary_switches_in_lowercase_hexadecimal'" for ***eNod4***-T.



*PROFINET IO network and names of **eNod4-T** stations in factory configuration. Only rotary switches have been reconfigured.*
Byte order: The byte order defines the order in which the application data are emitted on the network. The two possibilities are "Big Endian" or "Little-Endian". With AA as least significant byte, data of 2 or 4 bytes length are coded

for each possibility in this way: "Big Endian" 2 bytes: AA BB, 4 bytes: AA BB CC DD; "Little Endian" 2 bytes: BB AA, 4 bytes: DD CC BB AA. The default value of this parameter is "Little Endian".

7.3 Definition of protocols roles



*PROFINET IO protocols stack inside **eNod4**.*

Protocols involved in setting up an IO-Device (***eNod4)*** and the establishment and maintenance of a cyclic data connection are described below:

- LLDP (Link Layer Discovery Protocol). The LLDP messages are sent regularly on the network and inform other nodes about the identity of ***eNod4***.
- IP (Internet Protocol) allows routing of packets on the sub network by using IP address.
- ARP (Address Resolution Protocol). This protocol allows the creation of a resolution table of MAC addresses from an IP address. This table will be used in each node when a layer protocol based on IP (which uses an IP address) may wish to send a packet to another node on the Ethernet (MAC address) network.
- ICMP (Internet Control Message Protocol). Allows the 'Ping' command on the ***eNod4***.
- UDP (User Datagram Protocol) allows specification of a port number for an IP packet. The port number is associated with a higher level protocol.
- SNMP V1 (Simple Network Management Protocol) allows the network administrator to manage and oversee the whole network, including ***eNod4***.
- DCP (Discovery and Configuration Protocol). Enables the discovery and configuration of PROFINET nodes. The main functionality is similar to the more commonly used protocol DHCP (unsupported). Main available services are:
  - Identify: Allows an application to identify all PROFINET nodes present on the network, including ***eNod4***.
  - Signal: Allows the user to flash an LED on a specified node to identify the corresponding hardware equipment.

- Set IP (remanent or not). Allows the assignment of IP parameters (IP address, subnet mask, default gateway) for a node. Remanent means that parameters will keep their values after a power cycle, in non-remanent that they will be recovered to their previous values.
- Set Name Of Station (remanent or not). Allows the allocation of the name of the station for a node. Used in remanent, this service disables the option “use rotary switches for name of the station”; to reactivate it you can use eNodView.
- Set Reset Factory Settings: Allows the reset of all settings (application and networks) from ***eNod4*** to their default values. It places the IP settings to (0.0.0.0, 0.0.0.0, 0.0.0.0), turns the current name of the station into an empty field and disables the option to use rotary switches for name of the station.

- RPC (Remote Procedure Call): Allows the management of connections (called **AR (Application relation)** and CR (Communication relation)) for the exchange of cyclic data (IO Data) between the IO-Controller (PLC) and the Device-IO (***eNod4***). Allows also acyclic exchanges (called read/write Records).
- Profinet IO Data: Cyclic PROFINET IO data, these carrying data also contain status informations on the transported data. Compared with other communication standards based on Ethernet, useful cyclic data goes through fewer layers before reaching their destination. For example the IP network layer is not crossed by cyclic data (IO Data).
- Alarms: PROFINET IO alarms are sent by a node whenever a significant event occurs. ***eNod4*** sends an alarm on every appearance and disappearance of diagnostic that reports an application error. Error types corresponding to ***eNod4*** diagnostics are described in the appendix and in the GSDML file. This file can be imported into the engineering software used for the network monitoring.
- MRP (Media Redundancy Protocol): This Protocol allows ring topology. ***eNod4*** acts as a **MRP client** and is not able to act as manager. At least one manager (MRP Manager) is required on the network if the ring topology is desired.

7.4 Main scenario

The main scenario applies to PROFINET IO network; it can be used to diagnose possibly encountered problems on the network.

1. *PROFINET IO network is powered on.*
2. *IO-Devices emit LLDP frames to inform all nodes on the subnet of their presence and identity.*
3. *Network nodes resolve the IP addresses of the stations with which they wish to communicate in peer-to-peer using the ARP protocol.*
4. *With DCP services, IO-Controller identifies IO-Devices involved in its application. It configures their IP settings. ARP tables are updated consequently.*
5. *Using RPC, the IO-Controller opens and configures cyclic connections (AR) for data exchange with IO-Devices and if needed reads and writes application parameters.*
6. *Cyclic data exchanges begin between IO-Devices and the IO-Controller in both directions.*
7. *The application of IO-Controller operates with the data provided by IO-Devices and supplies data to IO-Devices to advance the process of the application.*

7.5 Alternative scenario: control, maintenance, supervision

On point 4 of the main scenario:

4 A. If the network manager wants to control, maintain or supervise the network

4. A.1. The network manager Ping the ***eNod4.***

4. A.2. The network manager consults the network information base of the ***eNod4*** *with SNMP V1.*

7.6 Alternative scenario: eNod4 error application detected

On point 7 of the main scenario:

*7 A. **eNod4** detects an application error*

*7. A.1. **eNod4** sends an alarm of appearance of diagnostic to the IO-Controller which opened and configured a data exchange connection with it.*

7. A.2. The network manager consults diagnostics, determines the cause of the problem and fixes it.

*7. A.3. **eNod4** sends an alarm of disappearance of diagnostic to the IO-Controller which opened and configured a data exchange connection with it.*

7.7 PROFINET IO LEDs

State of the **BF (Bus Fault) labeled NS** (Network Status) bicolor LED is described in the table below:

Color	State	Meaning
Green	*Blinking*	*A data connection is established and the DCP Signal service was initiated via the bus.*
Red	*Blinking*	*No exchange of data.*
	Always on	*Ethernet physical connection low speed detected or no physical connection detected.*
Red/Green	*Blinking*	*Self-test on power up*
-	*Always off*	*No error*

State of the **SF (System Fault) labeled MS** (Module Status) bicolor LED is described in the table below:

Color	State	Meaning
Green	*Blinking*	***STANDBY**: the device has not been configured*
	Always on	***DEVICE OPERATIONAL**: Device is operating correctly*
Red	*Blinking*	***MINOR FAULT**: the device detected a recoverable minor fault*
	Always on	***MAJOR FAULT**: the device detected a non-recoverable major fault*
Red/Green	*Blinking*	*Self-test on power up*
-	*Always off*	*Device not powered or defective*

State of the **ACT / LINK** ETH1 and ETH2 network RJ45 connector LED:

Color	State	Meaning
LINK (Eth1 & Eth2) Green	*Always on*	*A physical connection to the Ethernet exist*
	Always off	*Device not connected to the Ethernet*
ACT (Eth1 & Eth2) Yellow	*On*	*The device sends/receives Ethernet frames*
	Always off	*No traffic on the Ethernet*

7.8 Data arrangement

The provision model of data is very similar to the one used in PROFIBUS DP, this will allow users of ***eNod4*** **Profibus** an easy recycling of their application.

7.8.1 Cyclic data (IO Data)

Cyclic exchanged data are either provided by the IO-Device and consumed by the IO-Controller or provided by the IO-Controller and consumed by the IO-Device.
Data are contained in input or input/output modules (from the point of view of the IO-Controller). These modules are defined in the GSDML file and are presented in a separate chapter.
The designer can select modules that he needs and place them in communication slots. Thus, the slots contain modules. Slots are numbered. Slot 0 is not usable for data exchange, it contains DAP (Device Access Point) informations which defines, among other, which data module can be contained in which slots.

7.8.2 Acyclic data (Records)

Acyclic data are available in read-only or read/write access. They are accessed by using a slot, a sub slot and an **index**.
eNod4 acyclic data are accessible with any slot and sub slot. Indexes for the ***eNod4*** specific application data are presented in appendix.

7.9 PROFINET IO exchange of cyclic data

Acyclic data modules are described in GSDML file. This file can be imported into the engineering tool used for application design. Data modules can be freely plugged into any slot from 1 to 11. This will define the organization of cyclic data in the AR (Application Relation). Unnecessary modules for the application may not be plugged. Inserting data provided by ***eNod4*** automatically implies the insertion of data consumed by ***eNod4*** if the concerned module contains consumed data.

Presentation of provided data in modules:

Module name	Provided size in bytes	Provided Data
Status+Gross Measurement	*6*	*Measurement status (2 bytes)*
		Gross measurement (4 bytes)
I/OLevel+Net+Fact.	*10*	*Logical I/O level (2 bytes)*
		Net Measurement (4 bytes)
		Factory calibrated Meas. (4 bytes)
Flow rates Meas.	*12*	*Instant flow rate (4 bytes)*
		Average flow rate (4 bytes)
		Dosing quality factor (4 bytes)
Output control	*6*	*Flow rate control output (4 bytes)*
		Control output value (2 bytes)
Totalization	*10*	*Totalization flow rate (4 bytes)*
		Totalizer value (Great WU) (4 bytes)
		Complementary totalizer value (2 bytes)
Command/Response Reg	*2*	*Response register*
R/W request Reg.	*6*	*Transaction status (2 bytes)*
		Data read/written (4 bytes)
Status/Errors	*12*	*Dosing status (2 bytes)*
		Dosing errors report (2 bytes)
		Dosing errors counter (2 bytes)
		Last dosing error (2 bytes)
		Dosing weight deviation (4 bytes)
Other totals	*8*	*Grand total (Great WU) (4 bytes)*
		General total (Great WU) (4 bytes)
Ana. Output	*2*	*External value to control analog output*
1 ms counter	*4*	***eNod4** 1ms counter **

**for possible check of performances*

Presentation of consumed data in input/output modules:

Module name	*Consumed size in bytes*	*Consumed Data*
Command/Response Reg	*2*	*Command register*
R/W request Reg.	*6*	*Transaction request (2 bytes)*
		Data to be written (4 bytes)
Ana. Output	*2*	*External value to control analog output*

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

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

The module "R/W request Reg." allows requesting read/write of Record (acyclic data). So this **substitute** read/write of Record via the RPC protocol. The protocol described below (which is the same than the one used on ***eNod4*** Profibus product) allows performing read/write operations:

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

An IO-Controller can transmit a read or write request to ***eNod4*** by writing a specific code (see the codes listed in the appendix) into the transaction request register.

⇨ For a write request, the 4 following OUT bytes can be used so as to enter the new value.
⇨ ***eNod4*** IN are then updated :
- Transaction status is set to 0xFFFF in case of an error otherwise it takes the same value as the one entered in the transaction request word.
- For a read transaction, the value of the requested setting is set 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.

Note: For 2-bytes size data, the data is read/written through the 2 least significant bytes. Ignore the 2 most significant bytes.

Note: The "Transaction request" register **must be** set to 0x0000 before every new transaction.

The module "Ana. Output" allows external device to drive eNod4 analog output *current* or *voltage*. To achieve that, analog output must be configured to "*level on request*" function.

8 ETHERCAT

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
EtherCAT is the communication standard created by the EtherCAT Technology Group (ETG). It allows use of an industrial Ethernet network for the exchange of I/O data between a Master (e.g. PLC (Programmable Logic Controller)) that contains the intelligence of the application and Slaves. **eNod4 ETH EtherCAT** is an EtherCAT Slave.

8.1 Physical interface

eNod4 is fitted with two Ethernet ports on RJ45 connectors that are galvanically isolated. They support the switch or hub functions, specific functions of real time Ethernet systems and facilitate the implementation of line or ring topology (any topology type is possible for EtherCAT networks).
The speed is fixed at 100Mbit and operations at Full-Duplex (EtherCAT requirements).
The "In" port is the one in the center and the "Out" port is the one located at the edge (rear).
As EtherCAT communicates on Ethernet II type frames, **eNod4** is compatible with most of the existing network infrastructures (cards, connectors, network, hub and switches).

8.2 Network settings

The **eNod4** configuration tool, eNodView, is not necessary for the EtherCAT network setup. Nevertheless the identification of **eNod4 ETH EtherCAT** by eNodView is only supported from X version.

Device Identification Value: value of 2 Bytes length that is set locally on the device and allows an Explicit Device Identification of Slaves on EtherCAT network by the Master. This value can be set via rotary switches (ID-Selector) located in front of **eNod4**, the most significant byte is set to zero and the least significant byte is set according to value on rotary switches. This value must be different of zero and a reset must be done after each change to take effect.

8.3 Communication protocol

Data can be exchanged cyclically or acyclically. For acyclic data exchanges **eNod4** make use of a « mailbox » exchange protocol, CoE (CANopen application protocol over EtherCAT), which provides mechanisms to configure cyclic data exchange and parameters access (SDOs). Cyclical data are transmitted within Process Data Objects (PDOs). The data arrangement is described in a following chapter.

8.4 EtherCAT LEDs

State of the **(RUN LED: green, ERR LED: red) labeled NS** (Network Status) bicolor LED is described in the table below:

Color	*State*	*Meaning*
RUN LED Green	***Off***	***INIT:*** *The device is in state INIT.*
	Blinking	***PRE-OPERATIONAL:*** *The device is in PRE-OPERATIONAL state.*
	Single flash	***SAFE-OPERATIONAL:*** *The device is in SAFE-OPERATIONAL state.*
	Always on	***OPERATIONAL:*** *The device is in OPERATIONAL state.*
ERR LED Red	***Blinking***	***Invalid Configuration:*** *General Configuration Error.* *Possible reason: State change commanded by master is impossible due to register or object settings.*
	Single flash	***Local Error:*** *Slave device application has changed the EtherCAT state autonomously.* *Possible reason 1: A host watchdog timeout has occurred.* *Possible reason 2: Synchronization Error, device enters Safe-Operational automatically.*
	Double flash	***Process Data Watchdog Timeout:*** *A process data watchdog timeout has occurred.* *Possible reason: Sync Manager Watchdog timeout.*
Red/Green/Off	***Combinations of red and green: blinking, single and double flash***	*The status of the red and the green LED can be displayed combined. If for example the Ethernet cable is disconnected, then the following combination is displayed: Green single flash (SAFE-OPERATIONAL) and red double flash (Process Data Watchdog Timeout).*

State of the **SF (System Fault) labeled MS** (Module Status) bicolor LED is described in the table below:

Color	*State*	*Meaning*
Green	*Blinking*	***STANDBY****: the device has not been configured*
	Always on	***DEVICE OPERATIONAL****: Device is operating correctly*
Red	*Blinking*	***MINOR FAULT****:* *the device detected a recoverable minor fault*
	Always on	***MAJOR FAULT****:* *the device detected a non-recoverable major fault*
Red/Green	*Blinking*	*Self-test on power up*
-	*Always off*	*Device not powered or defective*

State of the **ACT / LINK** ETH1 and ETH2 network RJ45 connector LED:

Color	*State*	*Meaning*
LINK (Eth1 & Eth2) Green	*Always on*	*A link is established*
	Flashing	*The device sends/receives Ethernet frames*
	Always off	*No link established*
ACT (Eth1 & Eth2) Yellow	*This LED is not used.*	

8.5 Data arrangement

*The provision model of data is very similar to the one used in standard **eNod4 CANopen** (**eNod4 DIN**).*

8.5.1 Acyclic data (Objects)

Acyclic data are available in read-only or read/write access. Each data is an object or a sub-object included in an object. It is accessed using an index and a sub-index if it is a sub-object. Objects are stored in the **eNod4** object dictionary. The index/sub-index for **eNod4** application specific data are presented in the following chapters, they are similar to those used in **standard CANopen eNod4** (**eNod4 DIN**).
All objects of the object dictionary are described in the EtherCAT Slave Information File (ESI), an XML file that can be used by an EtherCAT network configuration tool.

8.5.2 Cyclic data (IO Data)

Data exchanged cyclically are either provided by the EtherCAT Slave and consumed by the Master (TxPDOs, Inputs) or provided by the Master and consumed by the EtherCAT Slave (RxPDOs, Outputs).
The data within PDOs are object dictionary objects (same as for acyclic exchanges). PDOs and their included objects are defined in the ESI file and are described in another chapter.

8.6 EtherCAT exchange of cyclic data

PDOs are described in ESI file. This file can be imported into the application design software. The designer can choose PDOs he needs and assign (enable) or de-assign (disable) them. In **eNod4** PDOs are of fixed size and cannot be remapped (change objects constituting them).

There may be no RxPDO data (Outputs, data consumed by **eNod4**) assigned for the cyclic exchange providing that there is at least one TxPDOs (Inputs, data produced by the **eNod4**) assigned. Similarly there may be no assigned TxPDOs for cyclic data exchange providing that there is at least one assigned RxPDO.

Every PDOs are not default assigned in **eNod4** (see tables below).

It is recommended to de-assign RxPDOs if there are not used in application to avoid writing of inappropriate values in calibration parameters of **eNod4**.

Presentation of data provided in the TxPDOs (Inputs, data produced by the eNod4):

Name	*Size (bytes)*	*Data*	*Default assigned*
TPDO1	*1*	*Response register*	*yes*
TPDO2	*6*	*Gross measurement (4 bytes)*	*yes*
		Measurement status (2 bytes)	
TPDO3	*6*	*Net measurement (4 bytes)*	*yes*
		Logical Inputs level (1 byte)	
		Logical Outputs level (1 byte)	
TPDO4	*24*	*Instant flow rate (4 bytes)*	*yes*
		Average flow rate (4 bytes)	
		Flow rate control output (4 bytes)	
		Control output value (2 bytes)	
		Totalization flow rate (4 bytes)	
		Totalizer value (Great WU) (4 bytes)	
		Complementary totalizer value (2 bytes)	
TPDO5	*16*	*Dosing status (2 bytes)*	*yes*
		Dosing errors report (2 bytes)	
		Dosing errors counter (2 bytes)	
		Last dosing error (2 bytes)	
		Dosing weight deviation (4 bytes)	
		Dosing quality factor (4 bytes)	
TPDO6	*4*	***eNod4** 1ms counter**	*yes*

for possible check of performances

Presentation of data consumed in RxPDO modules (Outputs, data consumed by eNod4):

Name	*Size (bytes)*	*Data*	*Default assigned*
RPDO1	*1*	*Command register*	*yes*
RPDO2	*4*	*Calibration load 1*	*no*
RPDO3	*8*	*Zero offset (4 bytes)*	*no*
		Span adjusting coefficient (4 bytes)	
RPDO4	*8*	*Maximum capacity (4 bytes)*	*no*
		Sensor sensitivity (4 bytes)	
RPDO5	*2*	*External value to control analog output*	*no*

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

Note: The “Command register” data **must be** set to 0x00 before each new command.

The « External value to control analog output» allows writing directly the analog output value. This is only possible when the analog output function assignment is set to « Level on request ».

9 MEASUREMENT AND STATUS

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Measurement status	*0x007D*	*/*	*0x009C*	*R:0x02C8 W: / + See modules list*	*0x5003 / 0x00 (M)*	*Uint*	*RO*
Gross measurement	*0x007E*	*/*	*0x009D*	*R:0x04CA W: / + See modules list*	*0x5001 / 0x00 (M)*	*Long*	*RO*
Tare value	*0x0080*	*/*	*0x0060*	*R:0x0470 W: /*	*0x5004 / 0x01*	*Long*	*RO*
Net measurement	*0x0082*	*/*	*0x009E*	*R:0x04CC W: / + See modules list*	*0x5000 / 0x00 (M)*	*Long*	*RO*
Factory calibrated points	*0x0084*	*/*	*0x009F*	*R:0x04CE W: / + See modules list*	*0x5002 / 0x00*	*Long*	*RO*
Preset Tare	*0x0097*	*0x65/16*	*0x0061*	*R:0x04C4 W:0x04C5*	*0x5004 / 0x02*	*Ulong*	*RW*
Defective measurement debounced time	*0x0A48*	*0x67/17*	*0x005D*	*R:0x0206 W:0x0207*	*0x4509/0x06*	*Uint*	*RW*
Defective measurement alarm activation time	*0x0A49*	*0x67/18*	*0x005E*	*R:0x0208 W:0x0209*	*0x4509/0x07*	*Uint*	*RW*
Sensor input control reference	*0x0A44*	*0x65/17*	*0x0062*	*R: 0x044C W: 0x044D*	*0x5004 / 0x03*	*long*	*RW*
Sensor input control result	*0x0A46*	*0x68/4*	*0x0063*	*R: 0x024E W: /*	*0x5004 / 0x04*	*Int*	*RO*
Sensor input control result max. tolerance	*0x0A47*	*0x65/18*	*0x0064*	*R: 0x020A W: 0x020B*	*0x5004 / 0x05*	*Uint*	*RW*

Note: ***eNod4*** Ethernet, see chapters on I/O scanning or cyclic exchanges for each protocol.

9.1 Measurement transmission

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

9.1.1 Gross measurement

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

9.1.2 Net measurement

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

9.1.3 Tare value

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

9.1.4 Factory calibrated points

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

9.1.5 Preset Tare value

A previous calculated tare can be restored using this variable.

9.1.6 Measurement status

The measurement status contains information on eNod4 measurement parameters.

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

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*	
01	*Defect: sensor input control result out of tolerances OR Sensor input control command in progress OR failed (timeout) OR Sensor input reference command in progress*	*causes a logical output assigned to the 'defective measurement' function to be set active. Causes the analog output assigned to a weight or flow rate image to be set in error mode.*
10	*gross meas.< (- max capacity) OR 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 Note 1*
1	*EEPROM failure*	
b_7		
0	*reserved*	*1 in SCMBus and fast SCMBus, 0 otherwise*
1		

bits	Meaning	Note
b8		
0	*IN1 logical level*	
1		
b9		
0	*IN2 logical level*	
1		
b10		
0	*OUT1 logical level*	
1		
b11		
0	*OUT2 logical level*	
1		
b12		
0	*OUT3 logical level*	
1		
b13		
0	*OUT4 logical level*	
1		
b14		
0	*no tare*	
1	*at least a tare has been processed*	
b15		
0	*reserved*	*1 in SCMBus and fast SCMBus, 0 otherwise*
1		

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

9.2 Weighing diagnosis

9.2.1 Global weighing diagnosis

An internal alarm flag reflects the integrity of the whole measurement chain. It's used to set logical output active or optional analog output in an error mode in order to warn about any defection on the measurement chain (defective measurement).
This variable is set active when at least one of the followings conditions occurs:

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

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

9.2.1.1 Defective measurement debounced time

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

9.2.1.2 Defective measurement alarm activation time

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

9.2.2 Sensor input control

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

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

9.2.2.1 Sensor input control reference

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

9.2.2.2 Sensor input control result

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

9.2.2.3 Sensor input control result max. tolerance

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

10 PROCESSING FUNCTIONAL COMMANDS

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Command register	*0x0090*	*0x68/1*	/	*See modules list*	*0x2003 / 0x00 (M)*	*Uint*	*RW*
Response register	*0x0091*	*0x68/2*	/	*See modules list*	*0x2004 / 0x00 (M)*	*Uint*	*RO*

10.1 Principles

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

the command register : dedicated to accept the functional commands

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

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

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

10.2 Functional commands list

Functional command	Command code	Note
Set to idle (00_H) response register	00_H	See § above
reset*	$D0_H$	
EEPROM storage	$D1_H$	
Restore default settings	$D2_H$	
Zero*	$D3_H$	
Tare*	$D4_H$	
Cancel tare*	$D5_H$	
Cancel current 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 adjustment	$F0_H$	
Dynamic zero acquisition	$F1_H$	
Preset tare*	$F2_H$	
Sensor input reference	EF_H	
Sensor input control	FD_H	
Totalization / errors clear	DF_H	
Dosing / batch start / resume	$E4_H$	
Dosing / batch stop / cancel	$E5_H$	
Suspend batch	$F7_H$	
Emptying hopper (cleaning)	$F4_H$	

Functional command	Command code	Note
Refilling start	*F5H*	
Refilling stop	*F6H*	
Learning cycle on next cycle	*F3H*	
Auto Measurement in two points for flow rate calibration	*C9H*	
Auto Measurement in three points for flow rate calibration	*CAH*	
Flow rate calibration	*F8H*	
PID parameters auto-adjustment	*F9H*	
Clear grand total	*EDH*	
Clear general total	*EEH*	

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

10.3 Functional commands description

10.3.1 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 followed by a reset.

10.3.2 EEPROM storage

eNod4 configuration and calibration are stored in a non-volatile memory (EEPROM). If changes are made in the device configuration, sending to ***eNod4*** the *'EEPROM storage'* functional command will allow ***eNod4*** to keep these modifications after a power shutdown or the reception a *'reset'* functional command.
Moreover the settings listed below need to be stored and will only be taken into account at the next device reboot:

- span adjusting coefficient
- calibration place **g** value
- place of use **g** value
- stability criterion
- legal for trade activation switch
- power-up zero
- A/D conversion rate
- Functioning mode and communication protocol
- Analog output voltage and current

10.3.3 Restore default settings

The *'restore default settings'* command causes ***eNod4*** to be set back to its default configuration. The default configuration corresponds to the one on delivery that means with factory settings. Be careful when using this command, all the default settings are recovered including the stored calibration and the legal for trade indicators.
Note: this functional command is not available in CANopen® communication protocol.

10.3.4 Zero

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

- measurement is stable
- Current gross measurement is within a ±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. This value is not stored in EEPROM.

10.3.5 Tare

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

10.3.6 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'* to be set back to 0.

10.3.7 Cancel last command

This command sets the response register to **00_H** and allows ***eNod4*** to ignore the functional command previously received and not yet issued (for example to exit a sequential procedure like a physical calibration).

10.3.8 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. This calibration must then be saved by sending to ***eNod4*** the '*store calibration'* functional command. Using the *'zero adjustment'* functional command is also recommended so as to completely adapt ***eNod4*** to the application.

10.3.9 Zero adjustment

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

10.3.10 Start physical calibration

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

10.3.11 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 10 seconds, '*calibration zero acquisition*' command is cancelled and an execution error is reported.
Note: In specific cases (silo for example), this step is not mandatory because it is possible to command a "zero adjustment" when the silo is empty.

10.3.12 Segment 1 acquisition

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

10.3.13 Segment 2/3 acquisition

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

10.3.14 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'*.

10.3.15 Logical outputs 1-4 activation/deactivation

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

10.3.16 Zero offset adjustment

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

10.3.17 Dynamic zero acquisition

When receiving a ***dynamic zero acquisition*** command ***eNod4*** calculates the average of successive weight measures during the time specified in ***dynamic zero acquisition time*** parameter. A new zero is then used if the averaged value is within a ±10% (±2% if the legal for trade option is enabled) range of the ***maximum capacity***.
There is no stability criteria required. Dynamic zero acquisition can also be launch by an input assigned to this function. ***Dynamic zero acquisition*** can be interrupted by sending ***cancel current command***.

10.3.18 Preset tare

With this command it is possible to retrieve a tare value defined previously.
Important: ***Preset tare*** parameter value must be stored before sending this command.

10.3.19 Sensor input reference

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

10.3.20 Sensor input control

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

10.3.21 Clear totalization & errors counter

This command triggers the main totalizer reset. The two parts of the totalizer, the main in ***weight unit*** x 1000, and the complementary part in ***weight unit*** are then reset.
The variables ***dosing errors report***, ***dosing errors counter*** and ***last dosing error*** are also reset.

10.3.22 Dosing / batch start / resume

In *batch mode* this command launches or resumes a cycle. A cycle can also be launched by an input assigned to this function. When the command is sent through the ***command register***, ***eNod4*** answers back the control state value through the ***response register***.
This command will not issue if a flow rate calibration has not been previously performed.
When a cycle starts and depending on parameters setting of eNod4, there will be firstly a refilling. Totalizer may be cleared and a learning cycle initiated.
If a batch has been previously suspended, this command allows resuming the cycle and the totalization. The ***dosing status*** variable allows monitoring enod4 state during the dosing cycle.
<u>Attention</u>: an input can only be assigned to ***start/stop dosing*** or ***suspend/resume batch***, with a functioning in a bistable mode and on rising or falling front depending on the selected logic.

10.3.23 Dosing stop / batch cancel

In loss in weight feeder mode, this command allows stopping the running dosing cycle. If a batch is pending, it will be stopped with any possible resumption.
Attention: an input can only be assigned to ***start/stop dosing*** or ***suspend/resume batch***, with a functioning in a bistable mode and on rising or falling front depending on the selected logic.

10.3.24 Suspend batch

In *batch mode* this command suspends dosing. It allows any cleaning operation without clearing a batch. Launching a *Dosing / batch start/* resume command will resume the dosing cycle.
Warning: an input can only be assigned to ***suspend/resume batch***, with a functioning in a bistable mode and on rising or falling front depending on the selected logic.

10.3.25 Emptying hopper (cleaning)

In loss in weight feeder mode, this command allows to automatically emptying the hopper (or silo) in preparation of a cleaning process or for changing the dosing product. Dosing is running and totalization works until the gross weight reaches the ***empty hopper level***.
An input can be assigned to ***start/stop hopper emptying***, with a functioning in a bistable mode and on rising or falling front depending on the selected logic. It's not possible to suspend an emptying.
The command ***start/stop dosing*** or ***suspend/resume batch*** systematically lead to the emptying shutdown.

10.3.26 Refilling start

This command allows refilling activation. Refilling stops when the gross weight reaches the ***refilling high level*** minus the ***refilling inflight value***. Inflight value is not adjusted automatically at each cycle.
Refilling can also be initiated/stopped by an input assigned to ***start/stop refilling***, with a functioning in a bistable mode and on rising or falling front depending on the selected logic.

10.3.27 Refilling stop

This command will stop refilling immediately. Refilling can also be initiated/stopped by an input assigned to ***start/stop refilling***, with a functioning in a bistable mode and on rising or falling front depending on the selected logic.

10.3.28 Learning cycle on next cycle

If ***PID activation***, ***smart refill mode*** and ***acquisition of flow rate refill references on learning cycle*** options are activated, this command will launch a learning cycle after the next refilling phase. This smart refill mode compensates for the flow rate variation due to the product thrust in the hopper or silo so that the flow rate always remains constant throughout the phase. New values for ***reference flow rate control output start refilling*** and ***reference flow rate control output end refilling*** parameters will be determined.

10.3.29 Flow rate calibration

So that ***eNod4*** can carry out an expected flow rate dosing in the best possible conditions, the flow rate calibration is required. This also applies when ***eNod4*** is used both as constant flow rate totalizer and as loss in weight feeder.
From this calibration will depend the accuracy of the flow rate obtained and on the settling time, if a PID controller is activated. This calibration is carried out in minimum two segments by the variable ***segments number for the calibration curve of flow rate***. In case the extraction device has a nonlinear response it is recommended to define maximum segments for the flow rate calibration.
In order to calibrate the flow rate, if the control of extraction device is directly provided by ***eNod4*** through an analog control output in current or voltage, analog output functioning of ***eNod4*** must be allocated to ***level on request*** function first.
For each calibration point of the variable ***control output value,*** read the appropriate *average flow rate.* Then provide each of the ***Calibration of flow rate point n (control output)*** and ***Calibration of flow rate point n (flow rate value)*** matching with control output value***.***
Validate the flow rate calibration by sending ***calibration of flow rate*** command.
Finally allocate in the end the current or voltage analog output of ***eNod4*** to ***flow rate control output*** function***.***

10.3.30 Auto Measurement in two or three points for flow rate calibration

In order to calibrate the flow rate, if the control of extraction device is directly provided by ***eNod4*** through an analog control output in current or voltage, analog output functioning of ***eNod4*** must be allocated to ***level on request*** function first. There are two commands, the first allows automatic measurement of flow rate in two points (35% & 70% of control output). The second allows automatic measurement of flow rate in three points (20%, 50% & 80%).

At the end of procedure, the flow rate calibration is validating automatically.

10.3.31 PID parameters auto-adjustment

The configuration of the PID controller can be made in a totally automatic way. The behavior of PID controller (slow, fast or stable) must be previously configured. You have also to configure ***PID adjustment flow rate*** parameter that will be used for PID auto-adjustment.

So that this controller automatic adjusting device works, both the weight calibration and the flowrate calibration must be previously carried out.

PID parameters auto-adjustment command will cause ***eNod4*** to perform successive dosing cycle sequence to calculate optimized PID coefficients ***Kp, Ti*** and ***Td***. The number of cycle sequence can be fixed by user (parameter).
It is strongly recommended that ***Td*** parameter does not exceeded value **5**.

10.3.32 Clear grand total

Clear great total command allows individual reset of ***Grand total*** (in ***weight unit*** x1000) totalizer.

10.3.33 Clear general total

Clear general total command allows individual reset of ***General total*** (in ***weight unit*** x1000) totalizer.

11 CALIBRATION SETTINGS AND PROCEDURES

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Maximum capacity	0x000C	0x65/1	0x0020	R:0x0420 W:0x0421	0x3002 / 0x00 (M)	Ulong	RW
Number of calibration segments	0x000E	0x65/2	0x0021	R:0x0222 W:0x0223	0x3000 / 0x00	Uint	RW
Calibration load 1	0x000F	0x65/3	0x0022	R:0x0424 W:0x0425	0x3001 / 0x01 (M)	Ulong	RW
Calibration load 2	0x0011	0x65/4	0x0023	R:0x0426 W:0x0427	0x3001 / 0x02	Ulong	RW
Calibration load 3	0x0013	0x65/5	0x0024	R:0x0428 W:0x0429	0x3001 / 0x03	Ulong	RW
Sensor sensitivity	0x0015	0x65/6	0x0025	R:0x042A W:0x042B	0x3004 / 0x00 (M)	Ulong	RW
Scale interval	0x0017	0x65/7	0x0026	R:0x022C W:0x022D	0x3003 / 0x00	Uint	RW
Zero calibration	0x0018	0x65/8	0x0027	R:0x0434 W:0x0435	0x3005 / 0x00	Long	RW
Span coefficient 1	0x001A	0x65/9	0x002B	R:0x0436 W:0x0437	0x3006 / 0x04	Float	RW
Span coefficient 2	0x001C	0x65/10	0x002C	R:0x0438 W:0x0439	0x3006 / 0x05	Float	RW
Span coefficient 3	0x001E	0x65/11	0x002D	R:0x043A W:0x043B	0x3006 / 0x06	Float	RW
Span adjusting coefficient	0x0020	0x65/12	0x0028	R:0x042E W:0x042F	0x3006 / 0x01 (M)	Ulong	RW
Calibration place g value	0x0022	0x65/13	0x0029	R:0x0430 W:0x0431	0x3006 / 0x02	Ulong	RW
Place of use g value	0x0024	0x65/14	0x002A	R:0x0432 W:0x0433	0x3006 / 0x03	Ulong	RW
Zero offset	0x0092	0x65/15	0x002E	R:0x0472 W:0x0473	0x2500 / 0x00 (M)	Long	RW

11.1 Principles

eNod4 is factory calibrated to deliver **500 000 counts for 2mV/V** with a load cell on the **A3** input connector.
The measurement scaling in ***eNod4*** can be adapted to his application by the user. Some settings and the 2 calibration methods allow the user to define his specific span according to his sensors characteristics.

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

11.2 Calibration methods

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

- **Theoretical calibration** involving the sensitivity of the sensor and a user-defined corresponding capacity
- **Physical calibration** involving 1, 2 or 3 know loads (for a load cell)

Both can be achieved thanks to the functional commands.

11.3 Settings description

11.3.1 Maximum capacity

The *'maximum capacity'* stands for the maximum sensor/load cell signal range. When the absolute value of the gross measurement exceeds its value plus 9 divisions, the b_3 bit (positive overloading) or the b_2 bit (negative overloading) of the measurement status is set to 1 (it can activate a logical output if it is assigned to the *'defective measurement'* function).
The zero acquisition (on request or at power-up) is done only if the gross measurement value is contained between a ±10% range of the *'maximum capacity'* (±2% if the *legal for trade* option is active).
The *'maximum capacity'* setting also allows calibrating ***eNod4*** in case of a theoretical calibration in association with the sensor sensitivity. Measurement scaling will be automatically adapted so as to deliver a gross measurement value equivalent to the *'maximum capacity'* for an analog signal corresponding to the sensor sensitivity.
After a theoretical calibration, the maximum capacity can be changed to fit to the application.
Admitted values : from 1 up to 10000000.

11.3.2 Number of calibration segments

The *'number of calibration segments'* defines how many calibration segments are used during the physical calibration procedure. Linear installations only need one segment.
Admitted values : from 1 up to 3.

11.3.3 Calibration loads 1/2/3

Before starting a physical calibration procedure, each calibration segment must be given a corresponding user value (for example, 1000 points for a 1 kg load).
Admitted values : from 1 up to 10000000.

11.3.4 Sensor sensitivity

The *'sensor sensitivity'* setting is used to achieve a theoretical calibration. The stored value for this parameter is the load cell sensitivity in **mV/V**.
The user can adapt the value delivered by ***eNod4*** for the associated signal using the *'maximum capacity'* and the *'sensor sensitivity'*.
This setting is expressed with a 10^{-5} factor (197500 is equivalent to a 1.975 mV/V load cell sensitivity or a 1.975 V input voltage).
Admitted values : from 1 up to 1000000.

11.3.5 Scale interval

The *'scale interval'* is the minimal difference between two consecutive indicated values (either gross or net).
Modification of scale interval is taking into account after a new calibration.
Admitted values : 1/2/5/10/20/50/100

11.3.6 Zero calibration

Zero calibration value corresponds to the A/D converter points measured during the '*zero acquisition*' step of a physical calibration.
For a theoretical calibration this value must be set. It can be set automatically with the '*zero adjustment*' command.
Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).
Admitted values : from 0 up to +-10000000

11.3.7 Span coefficients 1/2/3

These coefficients are computed and written during calibration process. Writing these coefficients could be done if you want to restore a previous calibration.
Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).
Admitted values : different from 0.

11.3.8 Span adjusting coefficient

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

Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).
Admitted values : from 900000 up to 1100000.

11.3.9 Calibration place g value / place of use g value

When the calibration place and the place of use of a measuring chain are different, a deviation can appear due to the difference of g (gravity) between the 2 places.
The ***eNod4*** calculates a ratio applied to the measure which compensates the difference of gravity between the 2 places.
The g value are expressed in 10^{-6} $m.s^{-2}$ (9805470 is equivalent to g = 9.805470 $m.s^{-2}$).
The ***eNodView*** software can help to determine the g value of a place.
Note: To be applied, any modification of this setting must be followed by an EEPROM back up and device reboots (hardware or software).
Admitted values : different from 0.

11.3.10 Zero offset

The *'Zero offset'* value contains the offset in factory calibrated points that can be added/subtracted (if its value is positive or negative) to the zero calibration value when using the *'zero offset'* functional command. Once the command has been successfully achieved, this register is set to 0.
Note: The 'Zero offset' value is not stored into EEPROM memory and is always equal to 0 after a device power-up or a software reset

Admitted values: different from 0.

12 FILTERS

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
A/D conversion rate	0x0036	*0x66/1*	*0x0030*	*R:0x0240 W:0x0241*	*0x4000 / 0x00*	Uint	RW
filters activation	0x0037 LSB	*0x66/2 LSB*	*0x0031 LSB*	*R:0x0242 LSB W:0x0243 LSB*	*0x4001 / 0x01 (byte)*	Byte	RW
Low-pass order	0x0037 MSB	*0x66/2 MSB*	*0x0031 MSB*	*R:0x0242 MSB W:0x0243 MSB*	*0x4001 / 0x02 (byte)*	Byte	RW
Low-pass cut-off frequency	0x0038	*0x66/3*	*0x0032*	*R:0x0244 W:0x0245*	*0x4001 / 0x03*	Uint	RW
Depth of moving average filter on weight	0x0039	*0x66/4*	*0x0033*	*R:0x0246 W:0x0247*	*0x4001 / 0x04*	Uint	RW
Tolerance of clipping filter on instant flow rates	0x003A	*0x66/5*	*0x0034*	*R:0x0248 W:0x0249*	*0x4001 / 0x05*	Uint	RW
Average flow rate determination depth	0x0059	*0x69/3*	*0x0072*	*R:0x0278 W:0x0279*	*0x4700 / 0x03*	Uint	RW

12.1 Principles

eNod4 contains 4 filtering layers that are user-configurable :

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

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

12.2 Settings description

12.2.1 A/D conversion rate

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

b_4	Rejection	
0	*60 Hz*	
1	*50 Hz*	
$b_3\ b_2\ b_1\ b_0$	A/D conversion rate (measures/s)	
	50-Hz rejection	60-Hz rejection
0000	*100*	*120*
0001	*50*	*60*
0010	*25*	*30*
0011	*12.5*	*15*
0100	*6.25*	*7.5*
1001	*1600**	*1920**
1010	*800**	*960**
1011	*400**	*480**
1100	*200*	*240*

***** values that can only be selected in transmitter mode.

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

12.2.2 Filters activation & order

This setting allows to define what filters are enabled in ***eNod4*** signal processing chain.
Note: the filters activation & order setting can be accessed through a 16-bits register except in CANopen® communication protocol where this word is divided into 2 8-bits registers:

b_1	Meaning
0	*self-adaptive filter disabled*
1	*self-adaptive filter enabled*
$b_{10}\ b_9\ b_8$	
000	*low-pass filter disabled*
010	*2nd order low-pass filter*
011	*3rd order low-pass filter*
100	*4th order low-pass filter*

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

12.2.3 Low-pass filter cut-off frequency

This register contains the low-pass filter cut-off frequency expressed in Hz and multiplied by 100. That means that 690 is equivalent to 6.90 Hz. The value must be compliant with the table shown below.
Admitted values: from 10 up to 20000.

12.2.4 Limitations

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

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

(*) values of A/D conversion rate that are accessible in feeder mode.

12.2.5 Depth of moving average filter on weights

A moving average filter on weight can be set in cascade after previous filters. This filter is used to smooth the weight value in case of random interferences. If enable, this filter computes the mean of a specified last number of measures which are output of the previous filters. The number of measures is defined by ***depth of moving average filter on weights*** parameter. A high filter depth will give a better stability, with a longer response time.

Filter depth admitted values: 0(disabled) up to 128.

12.2.6 Tolerance of clipping filter on instant flow rates

When eNod4 is used in loss in weight feeder, it's possible to operate a clipping filter on the successive values of flow rate. Clipping range is defined in 0.1% of the nominal flow rate. This filter will substitute flow rate values outside the clipping range by the meaning of precedent flow rate values. It allows to suppress short asynchronous physical disturbances on weight measures due to the environment. The filter is disabled when parameter is cleared.

12.2.7 Average flow rate determination depth

This parameter specifies the depth of the moving average filter on successive instant flow rate values in order to produce the average flow rate. The average flow rate is only produced for display and generally it is appropriate to set average flow rate determination depth parameter to the maximal value 128. The filter is disabled when parameter is cleared.

13 CONFIGURATION OF INPUT/OUTPUT

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Logical input 1 functioning	0x0042 LSB	0x67/1 LSB	0x0040 LSB	R:0x0250 LSB W: 0x0251 LSB	0x4501 / 0x02	Byte	RW
Logical input 2 functioning	0x0042 MSB	0x67/1 MSB	0x0040 MSB	R: 0x0250 MSB W: 0x0251 MSB	0x4501 / 0x03	Byte	RW
Logical input 3 functioning (IO+ version)	0x0041 LSB	0x67/14 LSB	0x0042 LSB	R:0x026A LSB W: 0x026B LSB	0x4501 / 0x04	Byte	RW
Logical input 4 functioning (IO+ version)	0x0041 MSB	0x67/14 MSB	0x0042 MSB	R:0x026A MSB W: 0x026B MSB	0x4501 / 0x05	Byte	RW
holding time	0x0043	0x67/2	0x0041	R:0x0252 W:0x0253	0x4501 / 0x01	Uint	RW
Analog output functioning (IO+ version)	0x0040	0x67/15	0x005B	R:0x026C W:0x026D	0x4509 / 0x05	Uint	RW
External value to control analog output (IO+ version)	0x0032	0x67/16	0x005C	R:0x023C W:0x023D + See modules list	0x5050 / 0x00 (M)	Uint	RW
Output 1 functioning	0x0044 LSB	0x67/3 LSB	0x0050 LSB	R:0x0254 LSB W: 0x0255 LSB	0x4509 / 0x01	Byte	RW
Output 2 functioning	0x0044 MSB	0x67/3 MSB	0x0050 MSB	R:0x0254 MSB W: 0x0255 MSB	0x4509 / 0x02	Byte	RW
Output 3 functioning	0x0045 LSB	0x67/4 LSB	0x0051 LSB	R:0x0256 LSB W: 0x0257 LSB	0x4509 / 0x03	Byte	RW
Output 4 functioning	0x0045 MSB	0x67/4 MSB	0x0051 MSB	R:0x0256 MSB W: 0x0257 MSB	0x4509 / 0x04	Byte	RW
Weight quantity per pulse on logical output	0x0057	0x69/29	0x008C	R:0x02AE W:0x02AF	0x4707 / 0x00	Uint	RW
Set point 1 high value	0x0046	0x67/5	0x0052	R:0x045A W:0x045B	0x4601 / 0x02	Long	RW
Set point 1 low value	0x0048	0x67/6	0x0053	R:0x045C W:0x045D	0x4601 / 0x03	Long	RW
Set point 2 high value	0x004A	0x67/7	0x0054	R:0x045E W:0x045F	0x4601 / 0x04	Long	RW
Set point 2 low value	0x004C	0x67/8	0x0055	R:0x0460 W:0x0461	0x4601 / 0x05	Long	RW

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Set point 3 high value	*0x004E*	*0x67/9*	*0x0056*	*R:0x0462 W:0x0463*	*0x4609 / 0x02*	*Long*	*RW*
Set point 3 low value	*0x0050*	*0x67/10*	*0x0057*	*R:0x0464 W:0x0465*	*0x4609 / 0x03*	*Long*	*RW*
Set point 4 high value	*0x0052*	*0x67/11*	*0x0058*	*R:0x0466 W:0x0467*	*0x4609 / 0x04*	*Long*	*RW*
Set point 4 low value	*0x0054*	*0x67/12*	*0x0059*	*R:0x0468 W:0x0469*	*0x4609 / 0x05*	*Long*	*RW*
1&2 Set points functioning	*0x0056 LSB*	*0x67/13 LSB*	*0x005A LSB*	*R:0x0258 LSB W: 0x0259 LSB*	*0x4601 / 0x01*	*Byte*	*RW*
3&4 Set points functioning	*0x0056 MSB*	*0x67/13 MSB*	*0x005A MSB*	*R:0x0258 MSB W: 0x0259 MSB*	*0x4609 / 0x01*	*Byte*	*RW*
Logical input level	*0x0094 LSB*	*0x68/3*	*/*	*See modules list*	*0x5100 / 0x00 (M)*	*Byte*	*RO*
Logical ouput level	*0x0094 MSB*	*0x68/4*	*/*	*See modules list*	*0x5200 / 0x00 (M)*	*Byte*	*RO*

13.1 Principles

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

13.1.1 Logical inputs

Each input can work individually in positive or negative logic. A holding time (de-bounced time) attached to all inputs can be configured.
Available functions see hereunder table:

Function	Operating mode	
	transmitter	**Loss in weight feeder**
None	●	●
Tare	●	●
Cancel tare	●	●
Zero	●	●
Transmit measurement (in SCMBus/fast SCMBus, in CANopen TPDO2 sending if input 1, TPDO3 sending if input 2)	●	
Continuous measurement transmit (SCMBus/fast SCMBus only)*	●	
Dynamic zero	●	●
Start/Stop refilling**		●
Start/Stop emptying hopper**		●
Start/Stop dosing**		●
Suspend/Resume batch**		●
Emergency stop		●
Reset totalization and dosing error counter		●
Sensor input control	●	●

* command operates on logical level
** these commands operates on front in a bistable functioning mode. For example for the Start/Stop refilling command, the first front initiates refilling, the second stops it.
Note: Most of functions set to logical inputs can also be sent by '*functional commands*'; for a precise description see § 'functional commands'.

- **None :** the input has no function
- **Tare :** a rising (positive logic) or a falling edge (negative logic) causes a tare function to be triggered .
- **Cancel tare :** a rising (positive logic) or a falling edge (negative logic) causes the current stored tare to be erased.
- **Zero :** a rising (positive logic) or a falling edge (negative logic) causes a zero function to be triggered.
- **Transmit measurement** : only available in CANopen® and SCMBus/fast SCMBus protocols and in transmitter mode only. A rising (positive logic) or a falling edge (negative logic) triggers a measurement transmission.
- **Continuous transmit measurement** : only available in SCMBus/fast SCMBus protocols. Measurements are continuously transmitted at a rate defined by the ***measurement transmission period*** parameter while the input is maintained at the chosen level.
- **Dynamic zero** : this input activation will cause ***eNod4*** to perform zero function. This zeroing acquisition is not conditioned by stability criterion and consists in a moving average of measures during an adjustable parameter ***dynamic zero acquisition time***. The new current zero is not stored permanently and can be canceled by a reset of the module.
- **Start/Stop refilling :** this command operates on front in a bistable functioning mode. In dosing mode, the first rising or falling edge (according to the configured logic) on this input causes a refilling procedure. The second one will stop it.

- **Start/Stop emptying hopper :** this command operates on front in a bistable functioning mode. In dosing mode, the first rising or falling edge (according to the configured logic) on this input causes an emptying procedure. The second one will stop it.
- **Start/Stop dosing :** this command operates on front in a bistable functioning mode. In dosing mode, the first rising or falling edge (according to the configured logic) on this input starts dosing. The second one stops it.
- **Suspend/Resume batch :** this command operates on front in a bistable functioning mode. In dosing mode, the first rising or falling edge (according to the configured logic) on this input suspends the batch. The second one resumes it.
- **Emergency stop :** a rising (positive logic) or a falling edge (negative logic) triggers an emergency stop which immediately stops the dosing, the totalization and eventually the refilling.
- **Reset totalization and dosing error counter :** a rising (positive logic) or a falling edge (negative logic) triggers the main totalizer reset. Both parts of the totalizer, the main in ***weight unit*** x1000 and the complementary part in ***weight unit*** are then reset.
  The ***dosing errors counter*** variable is also reset.
- **Sensor input control :** a rising (positive logic) or a falling edge (negative logic) triggers a test routine of the sensor input and produces a test result.

13.1.2 Analog output (IO+ version)

An optional analog board in *current* and *voltage* might be used with ***eNod4*** to provide IO+ version. This must be asked when ordering ***eNod4*** product.
Voltage output might be set either 0-5V or 0-10V, and the current output to 4-20mA, 0-24mA , 0-20mA or 4-20mA with alarm at 3.6mA. Both output (current and voltage) might separately be enable. Settings are effective after ***eNod4*** reset.
Analog output affectation function is common to both *current* and *voltage* output and might be assigned to followings:

function	*Operating mode*	
	transmitter	***loss in weight feeder***
none	●	●
gross measurement	●	●
net measurement	●	●
level on request	●	●
flow rate control output		●
instant flow rate	●	●
average flow rate	●	●

- **None :** analog outputs have no function.
- **Gross measurement :** analog outputs can be assigned to gross measurement copy. Maximal level value is related to ***Maximum Capacity*** parameter and works in mono-quadrant functioning. Bi-quadrant option can only be applied to gross measurement copy. When this option is activated, the lowest value of current and voltage levels corresponds to -***MC*** and the highest value to +***MC***.
- **Net measurement :** analog outputs can be assigned to net measurement copy. Maximal value is related to ***Maximum Capacity*** parameter and works in mono-quadrant functioning only. The highest value of current and voltage levels corresponds to +***MC*** in only one quadrant.
- **Level on request :** analog outputs are driven by master requests through the ***external value to control analog output*** variable (in 0.01% of the full scale of current or voltage analog outputs).
- **Flow rate control output :** PID controller output data (activated or not) drives current or voltage outputs generally coupled to the extraction device. Data is expressed in 0.01% of the high value of current or

voltage analog outputs. Maximal level output value corresponds to the maximal flowrate (see § flow rate calibration).

- **Instant flow rate :** analog outputs can be assigned to instant flow rate copy. Maximal level output value corresponds to the maximal flowrate (see § flow rate calibration).
- **Average flow rate :** analog outputs can be assigned to average flow rate copy. Maximal level output value corresponds to the maximal flowrate (see § flow rate calibration).

When analog output is assigned to “Gross measurement”, “Net measurement”, “Instant flow rate**”** or **“**Average flow rate” its value jumps to a special error value when the internal alarm flag described in “Weighing diagnosis” § in the MEASUREMENT AND STATUS § is activated. This allows to warn about defection of the measurement chain.
The error value on analog output is defined depending on voltage or current settings as described in following table:

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

* no error detection possible in this setting

13.1.3 Logical outputs

Each output can work individually in positive or negative logic
The available functions are:

function	Operating mode	
	transmitter	***Loss in weight feeder***
none	●	●
set point	●	●
motion	●	●
defective measurement	●	●
input image	●	●
level on request	●	●
cycle in progress		●
External totalizer		●
Flow rate alarm		●
Batch complete		●
Refilling in progress		●
Extraction activated		●
Volumetric phase		●

- **None :** the output has no function
- **Setpoint :** 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).
- **Motion :** the output is dedicated to copying the stability flag level.
- **Defective measurement :** the output level is set when the internal alarm flag described in "Weighing diagnosis" § in the MEASUREMENT AND STATUS § is activated. This allows to warn about defection of the measurement chain. Flowrate, refilling, empty or full hopper or too long batch alarms are not considered as defective measurements.
- **Input image :** the output is dedicated to copying a logical input level (outputs 1 and 3 correspond to inputs 1 and 3, outputs 2 and 4 correspond to input 2 and 4).
- **Level on request :** the input level is driven by master requests.
- **Cycle in progress :** in dosing mode, indicates that a dosing cycle is running.
- **External totalizer :** this output is dedicated to deliver pulses for external totalizer device.
  If a logical output is assigned to ***external totalizer*** function, ***eNod4*** will send a pulse every time totalization result will reach a multiple of weight value defined in ***weight quantity per pulse on logical output*** parameter expressed in ***weight unit*** x1000.
- **Flowrate alarm :** the output is dedicated to copying dosing alarms. Alarms might be flow rate, control output level, refilling, batch too long, empty hopper, full hopper or external totalizer overflow.
- **Batch complete :** in dosing mode, indicates that a batch is complete.
- **Refilling in progress :** indicates that a refilling process is running.
- **Extraction activated :** in dosing mode, indicates that the extraction device is activated.
- **Volumetric phase :** in dosing mode, indicates volumetric or gravimetric phases.

13.2 Settings description

13.2.1 Logical inputs assignment

The following table describes the possible assignments.

bits	*meaning*	*note*
$b_3 b_2 b_1 b_0$	***input 1 (and input 3 if IO+ version) assignment***	
0000	*none*	*the input has no function*
0001	*tare*	*equivalent to the functions described in § functional commands*
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)* ***Transmitter mode only***
0101	*Continuous transmit measurement**	*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* ***measurement transmission period*** *setting*
0110	*Dynamic zero*	*Same like equivalent functional command describe in § functional commands*
0111	*Start / Stop refilling***	*Not directly equivalent compared to functional commands*
1000	*Start / Stop hopper emptying***	*Not directly equivalent compared to functional commands*
1001	*Start/Stop dosing***	*Not directly equivalent compared to functional commands*
1010	*Suspend / Resume batch***	*Not directly equivalent compared to functional commands*
1011	*Emergency stop*	*There is no associated functional command*
1100	*Clear totalization and errors counter*	*Same like equivalent functional command describe in § functional commands*
1101	*Sensor input control*	*Same like equivalent functional command describe in § functional commands*
b_4	***input 1&3 logical***	
0	*negative logic*	*defines the edge (or level) that triggers input 1 function*
1	*positive logic*	
$b_6 b_5$	***measurement to be transmitted***	

bits	meaning	note
00	*gross*	*only for SCMBus/fast SCMBus protocols, no effect otherwise*
01	*net*	
10	*factory calibrated measurement*	

bits	meaning	note
b11 b10 b9 b8 (or b3 b2 b1 b0 in CANopen®)	***input 2 (and input 4 if IO+ version) assignment***	
0000	*none*	*the input has no function*
0001	*tare*	*equivalent to the functions described in § functional commands*
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)* ***Transmitter mode only***
0101	*Continuous transmit measurement**	*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* ***measurement transmission period*** *setting*
0110	*Dynamic zero*	*Same like equivalent functional command describe in § functional commands*
0111	*Start / Stop refilling***	*Not directly equivalent compared to functional commands*
1000	*Start / Stop hopper emptying***	*Not directly equivalent compared to functional commands*
1001	*Start/Stop dosing***	*Not directly equivalent compared to functional commands*
1010	*Suspend / Resume batch***	*Not directly equivalent compared to functional commands*
1011	*Emergency stop*	*There is no associated functional command*
1100	*Clear totalization and errors counter*	*Same like equivalent functional command describe in § functional commands*
1101	*Sensor input control*	*Same like equivalent functional command describe in § functional commands*
b_{12} (or b_4 in CANopen®)	***input 2&4 logical***	

bits	*meaning*	*note*
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*	

** these commands operates on front in a bistable functioning mode. For example for the Start/Stop refilling command, the first front initiates refilling, the second stops it.

Note 1: 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) and the mapping for the corresponding TPDO.

13.2.2 Holding time (debounced time)

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

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

The following tables describe the possible assignments.

bits	*meaning*	*note*
b3 b2 b1 b0	***analog output(s) assignment***	
0000	*none*	*the output level does not vary*
0001	*copy gross weight*	*Adjustable polarity*
0010	*copy net weight*	
0011	*level on request*	*parameter **External value to control analog output** will drive analog output*
0100	*flow rate control output*	*in loss in weight feeder mode (for extraction device control)*
0101	*copy instantaneous flow rate*	
0110	*copy average flow rate*	
b4	***polarity***	
0	*unipolar*	*could be set only with gross measurement*
1	*bipolar*	
$b_7\ b_6\ b_5$	***output voltage settings***	
000	*disable*	
001	*0 V - 5 V*	
010	*0 V - 10 V*	
$b_{10}\ b_9\ b_8$	***output current settings***	
000	*disable*	
001	*4 mA - 20 mA*	
010	*0 mA - 20 mA*	
011	*0 mA - 24 mA*	
100	*4 mA - 20 mA with alarm at 3.6 mA*	*voltage output is inactive (High-Z state)*

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

If an external device (e.g. PLC) would like to control extraction command through ***eNod4*** analog output, so that output must be set on ***level on request*** function. In this configuration ***eNod4*** will copy ***external value to control analog output*** parameter on analog output in current and voltage.
The external value parameter is expressed in 0.01% of full scale of analog output current or voltage.

13.2.5 Logical outputs 1&2 assignment

The following table describes the possible assignments.

bits	*meaning*	*note*
b3 b2 b1 b0	***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. § status register)*
0011	*Defective measurement*	*reflect the internal alarm flag described in "Weighing diagnosis" § in the MEASUREMENT AND STATUS §*
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. § functional commands)*
0110	*Cycle in progress*	*In loss in weight feeder mode*
0111	*External totalizer*	*In loss in weight feeder mode*
1000	*Flowrate alarm*	*In loss in weight feeder mode*
1001	*Batch complete*	*In loss in weight feeder mode*
1010	*Refilling in progress*	*In loss in weight feeder mode*
1011	*Extraction activated*	*In loss in weight feeder mode*
1100	*Volumetric phase*	*In loss in weight feeder mode*
b4	***output 1 logical***	
0	*negative logic*	*defines the output level when enabled*
1	*positive logic*	
b11 b10 b9 b8 (or b3 b2 b1 b0 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. § status register)*
0011	*defective measurement*	*reflect the internal alarm flag described in "Weighing diagnosis" § in the MEASUREMENT AND STATUS §*
0100	*input 2 image*	*copies input 2 level*

bits	***meaning***	***note***
0101	*level on request*	*output 2 level is driven by the 'OUT2 activation/deactivation' functional command (cf. § functional commands)*
0110	*Cycle in progress*	*In loss in weight feeder mode*
0111	*External totalizer*	*In loss in weight feeder mode*
1000	*Flowrate alarm*	*In loss in weight feeder mode*
1001	*Batch complete*	*In loss in weight feeder mode*
1010	*Refilling in progress*	*In loss in weight feeder mode*
1011	*Extraction activated*	*In loss in weight feeder mode*
1100	*Volumetric phase*	*In loss in weight feeder mode*
b12 (or b4 in CANopen®)	***output 2 logical***	
0	*negative logic*	*defines the output level when enabled*
1	*positive logic*	*defines the output level when enabled*

13.2.6 Logical outputs 3&4 assignment

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).

13.2.7 Weight quantity per pulse on logical output

When a logical output is assigned to ***external totalizer***, a pulse is generated every time the totalization increases a multiple of a ***weight quantity per pulse on logical output***. The maximum pulse frequency is 10 Hz and pulse duration is fixed at 50ms. The output pulse is not rounded (i.e. the pulse is issued only once ***weight quantity per pulse on logical output*** is strictly exceeded).

13.2.8 Set points functioning

The following table describes the possible assignments.

bits	*meaning*	*note*
b0	***set point 1 commutation mode***	
0	*window*	*only if output 1 assigned to the 'set point' function*
1	*hysteresis*	
b2 b1	***set point 1 comparison measurement***	
00	*gross*	
01	*net*	
10	*Sensor input control result*	
b3	***reserved (0)***	
b4	***set point 2 commutation mode***	
0	*window*	*only if output 2 assigned to the 'set point' function*
1	*hysteresis*	
b6 b5	***set point 2 comparison measurement***	
00	*gross*	
01	*net*	
10	*Sensor input control result*	
b7	***reserved (0)***	
b8 (or b0 in CANopen®)	***set point 3 commutation mode***	
0	*window*	*only if output 3 assigned to the 'set point' function*
1	*hysteresis*	
b10 b9 (or b2 b1 in CANopen®)	***set point 3 comparison measurement***	
00	*gross*	
01	*net*	
10	*Sensor input control result*	
b11 (or b3 in CANopen®)	***reserved (0)***	
b12 (or b4 in CANopen®)	***set point 4 commutation mode***	
0	*window*	*only if output 4 assigned to the 'set point' function*
1	*hysteresis*	

bits	meaning	note
b14 b13 (or b6 b5 in CANopen®)	***set point 4 comparison measurement***	
00	*gross*	
01	*net*	
10	*Sensor input control result*	
b15 (or b7 in CANopen®)	***reserved (0)***	

13.2.9 Set points high and low values

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 or to dosing result or dosing running total (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 ***eNod4*** "characteristics and functioning".
Admitted values: from -1000000 to 1000000.

13.3 Input/output level

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

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

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

14 LEGAL FOR TRADE OPTIONS

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Legal for trade version	*0x0004 LSB*	*0x64/3*	*0x0010 LSB*	*R: 0x0210 LSB W: /*	*0x3600 / 0x02*	*Byte*	*RO*
Legal for trade switch	*0x0004 MSB*	*0x64/4*	*0x0010 MSB*	*R: 0x0210 MSB W: 0x0211 MSB*	*0x3600 / 0x01*	*Byte*	*RW*
Legal for trade counter	*0x0005*	*0x64/5*	*0x0011*	*R: 0x0212 W: /*	*0x3600 / 0x03*	*Uint*	*RO*
Legal for trade checksum	*0x0006*	*0x64/6*	*0x0012*	*R: 0x0214 W: /*	*0x3600 / 0x04*	*Uint*	*RO*
Zero functions	*0x0007*	*0x64/7*	*0x0013*	*R:0x0216 W:0x0217*	*0x3501 / 0x01*	*Byte*	*RW*
Stability criterion	*0x0008 LSB*	*0x64/8 LSB*	*0x0014 LSB*	*R:0x0218 LSB W: 0x0219 LSB*	*0x3605 / 0x00*	*Byte*	*RW*
decimal point position	*0x0008 MSB*	*0x64/8 MSB*	*0x0014 MSB*	*R:0x0218 MSB W: 0x0219 MSB*	*0x3700 / 0x02*	*Byte*	*RW*
Weight unit	*0x0009*	*0x64/9*	*0x0015*	*R:0x041A W:0x041B*	*0x3700 / 0x01*	*String*	*RW*
Flow rate time unit	*0x005A*	*0x64/11*	*0x0016*	*R:0x021E W:0x021F*	*0x3700 / 0x03*	*Uint*	*RW*

14.1 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 behavior regarding the metrological requirements and track every configuration change that may affect the measurement determination.

14.2 Settings description

14.2.1 Legal for trade switch

This setting activates (**b_0 bit set 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.
- 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.
- taring is now impossible if gross measurement is negative.
- the measurement value variations cannot 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

14.2.2 Legal for trade software version

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

14.2.3 Legal for trade counter

If the *'legal for trade'* option is enabled, the legal for trade counter is incremented every time a backup 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)
- weight unit
- flow rate time unit
- zero functions

14.2.4 Legal for trade checksum

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

14.2.5 Zero functions

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).

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

14.2.6 Stability criterion

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

Bits b2 b1 b0	*Stability criterion*	*Note*
000	*no motion detection (always stable)*	
001	*0,25d*	*1d = 1 scale interval*
010	*0,5d*	
011	*1d*	
100	*2d*	

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

14.2.7 Decimal point position

Although ***eNod4*** measurements are integer values it is however possible to store a '*decimal point position'* so as to design a display related to the application. Its value represents the number of decimal digits. If the variable is set to Zero, it means that decimal point is not used.
Decimal point position has a consistency and influences flow rate values.
Note: the decimal point is directly integrated to SCMBus protocol frames (see § SCMBus).
Admitted values: from 0 up to 7.

14.2.8 Weight unit

It is possible to store the display weight unit into the ***eNod4***.
This weight unit is a combination of 4 characters and data has no consistency. There is no automatic calculation to adjust flowrate values neither flowrate calibration values.
Note: the unit is directly integrated to SCMBus protocol frames (see § SCMBus).

14.2.9 Flow rate time unit

It is possible to store the display *flow rate time unit* into the ***eNod4***.
Flow rate time unit is a combination of 2 characters and there is no automatic calculation to adjust flow rate value if modified.

In loss in weight feeder mode, following values are permitted for *flow rate time unit* parameter:

- 0x2073 for seconds (**s**)
- 0x6D6E for minutes(**mn**)
- 0x2068 for hours (**h**)

In opposition to the ***weight unit*** behaviour, the ***flowrate unit*** is consistent for instant and average flowrate determination. However and when the time unit is modified, there is no automatic redetermination of calibration flowrate values.

14.2.10 Save Tare and Zero in non-volatile memory

There are two options for saving TARE or ZERO request value in non-volatile memory. These options are accessible through the "dosing cycle options" Object.

15 LOSS IN WEIGHT FEEDER

15.1 Settings list

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Instant flow rate	0x0086	/	0x00BF	R: 0x0400 W: / + See modules list	0x5005 / 0x01	float	RO
Instant flow rate determination time	0x0058	0x69/1	0x0070	R: 0x0274 W: 0x0275	0x4700 / 0x01	Uint	RW
Instant flow rate correction factor	0x002C	0x69/2	0x0071	R: 0x0476 W: 0x0477	0x4700 / 0x02	Ulong	RW
Average flow rate	0x0088	/	0x00B0	R: 0x046E W: / + See modules list	0x5005 / 0x02	float	RO
Nominal flow rate	0x007A	0x69/28	0x008B	R: 0x04AC W: 0x4AD	0x4706 / 0x00	float	RW
Dosing weight deviation	0x0095	/	0x00B9	R: 0x04FA W: / + See modules list	0x5007 / 0x05	float	RO
Flow rate control output	0x008A	/	0x00B1	R: 0x04EA W: / + See modules list	0x5005 / 0x03	float	RO
Control output value	0x008C	/	0x00B2	R: 0x02EC W: / + See modules list	0x5005 / 0x04	Uint	RO
Refilling low level	0x0060	0x69/8	0x0077	R: 0x0482 W: 0x0483	0x4701 / 0x05	Ulong	RW
Refilling high level	0x0062	0x69/9	0x0078	R: 0x0484 W: 0x0485	0x4701 / 0x06	Ulong	RW
Refilling inflight value	0x0064	0x69/10	0x0079	R: 0x0486 W: 0x0487	0x4701 / 0x07	long	RW
Cycle and alarm options	0x005F	0x69/7	0x0076	R: 0x0280 W: 0x0281	0x4701 / 0x04	Uint	RW
Fixed flow rate during refilling	0x0066	0x69/11	0x007A	R: 0x0488 W: 0x0489	0x4701 / 0x08	float	RW
End of refill and cycle start stabilization time	0x005E	0x69/6	0x0075	R: 0x027E W: 0x027F	0x4701 / 0x03	Uint	RW
Weight to totalize (Great WU)	0x0075	0x69/25	0x0088	R: 0x04A6 W: 0x04A7	0x4705 / 0x01	Ulong	RW

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Complementary weight to totalize	*0x0077*	*0x69/26*	*0x0089*	*R: 0x02A8 W: 0x02A9*	*0x4705 / 0x02*	*Uint*	*RW*
Weigth to totalize inflight value	*0x0078*	*0x69/27*	*0x008A*	*R: 0x04AA W: 0x04AB*	*0x4705 / 0x03*	*long*	*RW*
Learning cycle end of refill delay	*0x005C*	*0x69/4*	*0x0073*	*R: 0x027A W: 0x027B*	*0x4701 / 0x01*	*Uint*	*RW*
Learning cycle flow rates reference determination time	*0x005D*	*0x69/5*	*0x0074*	*R: 0x027C W: 0x027D*	*0x4701 / 0x02*	*Uint*	*RW*
Empty hopper level	*0x0068*	*0x69/14*	*0x007D*	*R: 0x048E W: 0x048F*	*0x4702 / 0x01*	*Ulong*	*RW*
Full hopper level	*0x0A3E*	*0x69/57*	*0x00A9*	*R: 0x0404 W: 0x0405*	*0x4702 / 0x02*	*Ulong*	*RW*
Min permissible instant flow rate	*0x006E*	*0x69/18*	*0x0081*	*R: 0x0298 W: 0x0299*	*0x4704 / 0x01*	*Uint*	*RW*
Max permissible instant flow rate	*0x006F*	*0x69/19*	*0x0082*	*R: 0x029A W: 0x029B*	*0x4704 / 0x02*	*Uint*	*RW*
Min permissible flow rate control output	*0x0070*	*0x69/20*	*0x0083*	*R: 0x029C W: 0x029D*	*0x4704 / 0x03*	*Uint*	*RW*
Max permissible flow rate control output	*0x0071*	*0x69/21*	*0x0084*	*R: 0x029E W: 0x029F*	*0x4704 / 0x04*	*Uint*	*RW*
Inhibit time of flow rates alarms at start	*0x0072*	*0x69/22*	*0x0085*	*R: 0x02A0 W: 0x02A1*	*0x4704 / 0x05*	*Uint*	*RW*
Inhibit time of flow rates alarms in service	*0x0073*	*0x69/23*	*0x0086*	*R: 0x02A2 W: 0x02A3*	*0x4704 / 0x06*	*Uint*	*RW*
Maximum time for refilling start	*0x006B*	*0x69/15*	*0x007E*	*R: 0x0290 W: 0x0291*	*0x4703 / 0x01*	*Uint*	*RW*
Time interval for weight variation control on refilling	*0x006C*	*0x69/16*	*0x007F*	*R: 0x0294 W: 0x0295*	*0x4703 / 0x02*	*Uint*	*RW*
Maximum refilling time	*0x006D*	*0x69/17*	*0x0080*	*R: 0x0296 W: 0x0297*	*0x4703 / 0x03*	*Uint*	*RW*
Maximum batch time	*0x0074*	*0x69/24*	*0x0087*	*R: 0x02A4 W: 0x02A5*	*0x4704 / 0x07*	*Uint*	*RW*
Totalizer value (Great WU)	*0x008D*	*/*	*0x00B4*	*R: 0x04F0 W: / + See modules list*	*0x5006 / 0x01 (M)*	*Ulong*	*RO*
Complementary totalizer value	*0x008F*	*/*	*0x00B5*	*R: 0x02F2 W: / + See modules list*	*0x5006 / 0x02 (M)*	*Uint*	*RO*

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Dosing status	*0x009A*	*/*	*0x00BA*	*R: 0x02FC W: / + See modules list*	*0x5007 / 0x01 (M)*	*Uint*	*RO*
Dosing errors report	*0x009B*	*/*	*0x00BB*	*R: 0x02FE W: / + See modules list*	*0x5007 / 0x02 (M)*	*Uint*	*RO*
Dosing errors counter	*0x009C*	*/*	*0x00BC*	*R: 0x023E W: / + See modules list*	*0x5007 / 0x03 (M)*	*Uint*	*RO*
Last dosing error	*0x009D*	*/*	*0x00BD*	*R: 0x024A W: / + See modules list*	*0x5007 / 0x04 (M)*	*Uint*	*RO*
Dosing quality factor	*0x009E*	*/*	*0x00B3*	*R: 0x04EE W: / + See modules list*	*0x5005 / 0x05 (M)*	*float*	*RO*
Totalization flow rate	*0x00A0*	*/*	*0x00B8*	*R: 0x04F8 W: / + See modules list*	*0x5006 / 0x03 (M)*	*float*	*RO*
Grand total (Great WU)	*0x00A2*	*/*	*0x00B6*	*R: 0x04F4 W: / + See modules list*	*0x5006 / 0x04*	*Ulong*	*RO*
General total (Great WU)	*0x00A4*	*/*	*0x00B7*	*R: 0x04F6 W: / + See modules list*	*0x5006 / 0x05*	*Ulong*	*RO*
Extraction time	*0x00A6*	*/*	*0x00BE*	*R: 0x0402 W: /*	*0x5007 / 0x06*	*float*	*RO*
Calibration of flow rate point 1 (control output value)	*0x0A00*	*0x69/32*	*0x008F*	*R: 0x02B4 W: 0x02B5*	*0x470A / 0x01*	*Uint*	*RW*
Calibration of flow rate point 2 (control output value)	*0x0A01*	*0x69/33*	*0x0090*	*R: 0x02B6 W: 0x02B7*	*0x470A / 0x02*	*Uint*	*RW*
Calibration of flow rate point 3 (control output value)	*0x0A02*	*0x69/34*	*0x0091*	*R: 0x02B8 W: 0x02B9*	*0x470A / 0x03*	*Uint*	*RW*
Calibration of flow rate point 4 (control output value)	*0x0A03*	*0x69/35*	*0x0092*	*R: 0x02BA W: 0x02BB*	*0x470A / 0x04*	*Uint*	*RW*
Calibration of flow rate point 5 (control output value)	*0x0A04*	*0x69/36*	*0x0093*	*R: 0x02BC W: 0x02BD*	*0x470A / 0x05*	*Uint*	*RW*
Calibration of flow rate point 6 (control output value)	*0x0A05*	*0x69/37*	*0x0094*	*R: 0x02BE W: 0x02BF*	*0x470A / 0x06*	*Uint*	*RW*

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Calibration of flow rate point 7 (control output value)	*0x0A06*	*0x69/38*	*0x0095*	*R: 0x02C0* *W: 0x02C1*	*0x470A / 0x07*	*Uint*	*RW*
Calibration of flow rate point 8 (control output value)	*0x0A07*	*0x69/39*	*0x0096*	*R: 0x02C2* *W: 0x02C3*	*0x470A / 0x08*	*Uint*	*RW*
Calibration of flow rate point 9 (control output value)	*0x0A08*	*0x69/40*	*0x0097*	*R: 0x02C6* *W: 0x02C7*	*0x470A / 0x09*	*Uint*	*RW*
Calibration of flow rate point 10 (control output value)	*0x0A09*	*0x69/41*	*0x0098*	*R: 0x02C8* *W: 0x02C9*	*0x470A / 0x0A*	*Uint*	*RW*
Calibration of flow rate point 1 (flow rate value)	*0x0A0A*	*0x69/42*	*0x0099*	*R: 0x04CA* *W: 0x04CB*	*0x470B / 0x01*	*float*	*RW*
Calibration of flow rate point 2 (flow rate value)	*0x0A0C*	*0x69/43*	*0x009A*	*R: 0x04CC* *W: 0x04CD*	*0x470B / 0x02*	*float*	*RW*
Calibration of flow rate point 3 (flow rate value)	*0x0A0E*	*0x69/44*	*0x009B*	*R: 0x04CE* *W: 0x04CF*	*0x470B / 0x03*	*float*	*RW*
Calibration of flow rate point 4 (flow rate value)	*0x0A10*	*0x69/45*	*0x009C*	*R: 0x04D0* *W: 0x04D1*	*0x470B / 0x04*	*float*	*RW*
Calibration of flow rate point 5 (flow rate value)	*0x0A12*	*0x69/46*	*0x009D*	*R: 0x04D2* *W: 0x04D3*	*0x470B / 0x05*	*float*	*RW*
Calibration of flow rate point 6 (flow rate value)	*0x0A14*	*0x69/47*	*0x009E*	*R: 0x04D4* *W: 0x04D5*	*0x470B / 0x06*	*float*	*RW*
Calibration of flow rate point 7 (flow rate value)	*0x0A16*	*0x69/48*	*0x009F*	*R: 0x04D6* *W: 0x04D7*	*0x470B / 0x07*	*float*	*RW*
Calibration of flow rate point 8 (flow rate value)	*0x0A18*	*0x69/49*	*0x00A0*	*R: 0x04D8* *W: 0x04D9*	*0x470B / 0x08*	*float*	*RW*
Calibration of flow rate point 9 (flow rate value)	*0x0A1A*	*0x69/50*	*0x00A1*	*R: 0x04DA* *W: 0x04DB*	*0x470B / 0x09*	*float*	*RW*
Calibration of flow rate point 10 (flow rate value)	*0x0A1C*	*0x69/51*	*0x00A2*	*R: 0x04DC* *W: 0x04DD*	*0x470B / 0x0A*	*float*	*RW*
Segments number for the calibration curve of flow rate	*0x0A1E*	*0x69/31*	*0x008E*	*R: 0x02B2* *W: 0x02B3*	*0x4709 / 0x00*	*Uint*	*RW*
Reference flow rate control output start refilling	*0x0A31*	*0x69/12*	*0x007B*	*R: 0x048A* *W: 0x048B*	*0x4701 / 0x09*	*float*	*RW*
Reference flow rate control output end refilling	*0x0A33*	*0x69/13*	*0x007C*	*R: 0x048C* *W: 0x048D*	*0x4701 / 0x0A*	*float*	*RW*

Name	Modbus address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Kp	0x0A35	0x69/52	0x00A3	R: 0x04DE W: 0x04DF	0x470C / 0x01	float	RW
Ti	0x0A37	0x69/53	0x00A4	R: 0x04E0 W: 0x04E1	0x470C / 0x02	Ulong	RW
Td	0x0A39	0x69/54	0x00A5	R: 0x04E2 W: 0x04E3	0x470C / 0x03	Ulong	RW
PID behavior	0x0A3B	0x69/55	0x00A6	R: 0x02E4 W: 0x02E5	0x470C / 0x04	Uint	RW
PID adjustment flow rate	0x0A3C	0x69/56	0x00A7	R: 0x04E6 W: 0x04E7	0x470C / 0x05	float	RW
Number of cycles for adjustment PID	0x0A50	0x69/58	0x00E1	/	0x470C / 0x06	Uint	RW
Cut-off frequency for Totalization flow rate (x100)	0x0A51	0x69/59	0x00E2	/	0x4001/ 0x06	Uint	RW
Time unit for totalization flow rate	0x0A52	0x69/60	0x00E3	/	0x3700 / 0x04	Uint	RW
Weight unit for totalization flow rate	0x0A53	0x69/61	0x00E4	/	0x3700 / 0x05	Ulong	RW
Dynamic zero acquisition time	0x005B	0x69/30	0x008D	R:0x02B0 W:0x02B1	0x4708 / 0x00	Uint	RW

15.2 Settings description

- **Instant flowrate:** loss in weight during a period of time expressed in ***weight unit*** per ***flow rate time unit***.
- **Instant flow rate determination time:** time in seconds used for ***instant flow rate*** determination.
- **Instant flow rate correction factor:** correction factor coefficient expressed in 10^{-6}, so 1000000 corresponds to a correction factor coefficient of 1.00000 applied on the ***instant flow rate***.
- **Average flow rate:** flow rate output of moving average filter on successive values of instant flowrate.
- **Nominal flow rate:** target of flow rate expressed in ***weight unit*** per ***flow rate time unit***.
- **Dosing weight deviation:** weight deviation between the totalization of weight during 1 second and what should have been totalized at nominal flowrate.
- **Flow rate control output:** this is the flow rate to control the extraction device, expressed in ***weight unit*** per ***flow rate time unit***. It can take either a fixed value of flow rate, or the output of PID controller, if enabled, or a variable flowrate function of weight value in the hopper (or silo) during refilling phase (in smart refilling mode).
- **Control output value:** control value of the analog output expressed in 0.01% of maximum output current or voltage.
- **Refilling low level:** gross weight level below which a refilling has to operate if the ***auto refilling*** option is activated, cf § refilling.
- **Refilling high level:** gross weight level above which a refilling has to stop when there is no ***refilling inflight value***.
- **Refilling inflight value:** refilling stops when the gross weight will reach ***refilling high level*** - ***refilling inflight value.***
- **Cycle and alarm options:** this register defines cycle and alarms functioning, cf § alarms.
  See following table:

bits set to 1	Meaning
$b_7 \dots b_0$	**Cycle options**
b_0	*Batch dosing*
b_1	*Clear totalization when starting a new batch cycle*
b_2	*Auto refilling*
b_3	*Smart refill mode*
b_4	*Acquisition of flow rate refill references on learning cycle*
b_5	*PID activation*
b_6	*End of batch refill suppression*
b_7	**Reserved (0)**
$b_{15} \dots b_8$	**Alarm options**
B_8	*Stop dosing and refilling / suspend batch on alarm*
B_9	*Refilling to last value of flow rate*
B_{10}	*Save Tare in non-volatile memory*
B_{11}	*Save Zero in non-volatile memory*
$b_{15} \dots b_{12}$	**Reserved (0)**

- **Fixed flow rate during refilling:** flow rate fixed value used during refilling phase when the option ***smart refill mode*** is deactivated, cf § refilling.
- **End of refill stabilization time:** delay in seconds that ensures that the instant flow rate has reached the flow rate target before starting instant flow rate determination phase. This period should be the lowest possible to reduce this volumetric phase to a minimum, cf § refilling.
- **Weight to totalize (Great WU):** in batch mode, the main part of batch target value in ***weight unit*** x 1000, cf § totalization.
- **Complementary weight to totalize:** in batch mode, the complementary part of batch target value in ***weight unit***, cf § totalization.
- **Weight to totalize inflight value:** in batch mode, inflight value target in ***weight unit*** to apply to the weight to totalize, cf § totalization.
- **Learning cycle end of refill delay:** delay in seconds that ensures the average flow rate to be null before activation of the extraction device. It is better to choose a long period to be sure there is no more fall of material and also the average flow rate to be stabilized to zero before activation of the extraction device, cf § learning cycle end of refill delay.
- **Learning cycle flow rates reference determination time:** time in seconds during which the flow rate referencing is carrying out, which will be used to drive the extraction during refilling when the ***smart refill mode*** option is activated, cf § learning cycle flow rates reference determination time.
- **Empty hopper level:** gross weight level under which the dosing stops and an empty hopper alarm occurs.
- **Full hopper level:** gross weight level over which the dosing stops and a full hopper alarm occurs (dosing and refilling are stopped).
- **Min permissible instant flow rate:** low limit of flow rate value under which a flow rate alarm occurs, cf § flow rate / control output alarms. Expressed in 0.1% of nominal flow rate.
- **Max permissible instant flow rate:** high limit of flow rate value over which a flow rate alarm occurs, cf § flow rate / control output alarms. Expressed in 0.1% of nominal flow rate.
- **Min permissible flow rate control output:** low limit of flow rate control output value under which a flow rate control output occurs, cf § flow rate / control output alarms. Expressed in 0.1% of nominal flow rate.

- **Max permissible flow rate control output:** high limit of flow rate output control output value over which a flow rate control output occurs, cf § flow rate / control output alarms. Expressed in 0.1% of nominal flow rate.
- **Inhibit time of flow rates alarms at start:** time throughout no flow rate alarm or flowrate control alarm will occur when a cycle is starting to allow dosing stabilization, cf § flow rate / control output alarms. Expressed in seconds.
- **Inhibit time of flow rates alarms in service:** time throughout flow rate or flow rate control output values should be kept outside the limit specified for an alarm to occur and after the ***Inhibit time of flow rates alarms at start***, cf § flow rate / control output alarms. Expressed in seconds.
- **Maximum time for refilling start:** maximal delay to the gross weight to be higher than its value at refilling start. The aim is to detect no material fall when activating the refilling whether it is a manual or automatic control (lack of material, clogging, valve or control failure), cf § refilling alarms. Expressed in seconds.
- **Time interval for weight variation control on refilling:** time during which a weight variation should be positive. The aim is to detect any malfunction during the refilling phase (lack of material or clogging), cf § refilling alarms. Expressed in seconds.
- **Maximum refilling time:** maximal time for the refilling phase to complete, cf § refilling alarms. Expressed in seconds.
- **Totalizer value (Great WU)):** the main totalization result in ***weight unit*** x 1000, cf § totalization.
- **Complementary totalizer value:** the complementary totalization result in ***weight unit***., cf § totalization.
- **Dosing status:** this register describes dosing cycle phases and additional information on the feeder functioning. See following table:

bits set to 1	***Meaning***
b_7 … b_0	***Dosing steps***
b_0	*Stop*
b_1	*Refilling*
b_2	*Learning cycle end of refill delay*
b_3	*End of refilling stabilization*
b_4	*Flow rate determination*
b_5	*Dosing*
b_6	*Batch suspended*
b_7	*Hopper emptying*
b_8	***Reserved (0)***
b_{15} … b_9	***Additional informations***
b_9	*Learning cycle*
b_{10}	*Flow rate calibrated*
b_{11}	*Volumetric dosing*
b_{12}	*Extraction activated*
b_{13}	*Refilling in progress*
b_{14}	*Dosing cycle in progress*
b_{15}	*Batch complete*

- **Dosing errors report:** this register describes exclusively the status of alarms that can occur, cf § alarms. See coding definition in the table below:

$b_{15} \dots b_0$ bits set to 1	Meaning
b_0	*Instant flow rate > Max permissible instant flow rate*
b_1	*Instant flow rate < Min permissible instant flow rate*
b_2	*Flow rate control output > Max permissible flow rate control output*
b_3	*Flow rate control output < Min permissible flow rate control output*
b_4	*No weight variation on refilling*
b_5	*Too long refilling time*
b_6	*Empty hopper*
b_7	*External totalizer output overflow*
b_8	*Too long dosing batch time*
b_9	*Full hopper*
$b_{15} \dots b_{10}$	*Reserved (0)*

- **Dosing errors counter:** this register counts the dosing errors that may occur during dosing cycles. This counter can only be cleared by ***totalization / errors clear*** command, cf § alarms.
- **Last dosing error:** the last dosing error that occurs is stored in this register. Coding definition is equal to the variable ***dosing errors report***. This register can only be cleared by ***totalization / errors clear*** command, cf § functional commands.
- **Dosing quality factor:** every second a standard deviation of the ***dosing weight deviation*** successive values is calculated. Lower is the value, more constant is the flow rate.
- **Totalization flow rate:** information of flow rate that reflects the flow rate used for totalization and can be used for external display purpose (remote display or HMI). It has its own unity and filter.
- **Grand total (Great WU):** high level totalization in ***weight unit*** x 1000. This totalization can be cleared separately by the ***clear grand total*** command. The data of this totalizer is being permanently backed up internally, cf § totalization.
- **General total (Great WU):** high level totalization in ***weight unit*** x 1000. This totalization can be cleared separately by the ***clear general total*** command. The data of this totalizer is being permanently backed up internally, cf § totalization.
- **Extraction time:** time measurement of extraction activation, expressed in seconds. This value is cleared each time the extraction is activated. This parameter may be used for the evaluation of the flow rate accuracy of the loss in weight feeder.
- **Calibration of flow rate point n (control output value):** expressed in 0.01% of maximum output current or voltage. Up to 10 calibration points can be configured, see flow rate calibration, cf § flow rate calibration.
- **Calibration of flow rate point n (flow rate value):** expressed in ***weight unit*** per ***flow rate time unit***. Up to 10 calibration points can be configured, see flow rate calibration, cf § flow rate calibration.
- **Segments number for the calibration curve of flow rate:** when the flow rate of an extraction device has a nonlinear response in function of the flow rate output control, up to 9 segments can be configured, see flow rate calibration, cf § flow rate calibration.
- **Reference flow rate control output start refilling:** reference flow rate value corresponding to the flowrate control output that is applied at the beginning of refilling phase in smart refill mode. In this mode, the flow rate output control value is a function of the gross weight of the hopper or silo. This parameter can be determined automatically by a learning cycle, cf § refilling.
- **Reference flow rate control output end refilling:** reference flow rate value corresponding to the flowrate control output that is applied at the end of refilling phase in smart refill mode. In this mode, the flow rate output control value is a function of the gross weight of the hopper or silo. This parameter can be determined automatically by a learning cycle, cf § refilling.
- **Kp :** proportional coefficient of the PID controller that drives the extraction device. An increase of this

parameter will degrade stability, reduce the rise time and increase the overshoot.

- **T_i :** integration time constant of the PID controller that drives the extraction device, expressed in ms. An increase of this parameter will reduce the steady state error but will degrade stability and increase time rise and overshoot.
- **Td :** derivate time constant of the PID controller that drives the extraction device. Adding some derivate can improve time rise and overshoot. The great majority of extraction devices don't accept derivate correction or a very low value for the derivate term.
- **PID behavior: *eNod4*** is fitted with an automatic adjustment device of the PID parameters. Slow, Fast or stable behavior can be selected by the ***PID behavior*** variable that takes **2**, **1** or **0** values respectively.
- **PID adjustment flow rate:** flow rate value that will be used for the automatic adjustment of PID parameters. In addition to the ***PID behavior*** variable, the variable has to be defined previously to the ***PID parameters auto-adjustment*** command execution.
- **Number of cycles for PID adjustment :** It’s the number of cycles during which PID coefficients are calculated. If this number is large, sequence will be longer but computing accuracy will be greater. We advise to fix this number at 10.

16 MODBUS RTU/ MODBUS TCP REGISTERS

Chapter	Name	Modbus RTU / TCP address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Protocol	Program version	0x0000	0x64/1	0xAFF0 (I&M0)	/	0x100A / 0x00	Uint	RO
Protocol	Address /	0x0001	0x64/2	/	/	/	Byte	RO
Protocol	baudrate						Byte	RO
Protocol	Serial number (not managed)	0x0002	/	0xAFF0 (I&M0)	/	0x1018 / 0x04	Ulong	RO
Metrology	Legal for trade version	0x0004 LSB	0x64/3	0x0010 LSB	R: 0x0210 LSB W: /	0x3600 / 0x02	Byte	RO
Metrology	Legal for trade switch	0x0004 MSB	0x64/4	0x0010 MSB	R: 0x0210 MSB W: 0x0211 MSB	0x3600 / 0x01	Byte	RW
Metrology	Legal for trade counter	0x0005	0x64/5	0x0011	R: 0x0212 W: /	0x3600 / 0x03	Uint	RO
Metrology	Legal for trade checksum	0x0006	0x64/6	0x0012	R: 0x0214 W: /	0x3600 / 0x04	Uint	RO
Metrology	Zero functions	0x0007	0x64/7	0x0013	R:0x0216 W:0x0217	0x3501 / 0x01	Uint	RW
Metrology	Stability criterion	0x0008 LSB	0x64/8 LSB	0x0014 LSB	R:0x0218 LSB W: 0x0219 LSB	0x3605 / 0x00	Byte	RW
Metrology	decimal point position	0x0008 MSB	0x64/8 MSB	0x0014 MSB	R:0x0218 MSB W: 0x0219 MSB	0x3700 / 0x02	Byte	RW
Metrology	Weight unit	0x0009	0x64/9	0x0015	R:0x041A W:0x041B	0x3700 / 0x01	String	RW
	Reserved (2 bytes)	0x000B						
Calibration	Maximum capacity	0x000C	0x65/1	0x0020	R:0x0420 W:0x0421	0x3002 / 0x00 (M)	Ulong	RW
Calibration	Number of calibration segments	0x000E	0x65/2	0x0021	R:0x0222 W:0x0223	0x3000 / 0x00	Uint	RW
Calibration	Calibration load 1	0x000F	0x65/3	0x0022	R:0x0424 W:0x0425	0x3001 / 0x01 (M)	Ulong	RW
Calibration	Calibration load 2	0x0011	0x65/4	0x0023	R:0x0426 W:0x0427	0x3001 / 0x02	Ulong	RW
Calibration	Calibration load 3	0x0013	0x65/5	0x0024	R:0x0428 W:0x0429	0x3001 / 0x03	Ulong	RW
Calibration	Sensor sensitivity	0x0015	0x65/6	0x0025	R:0x042A W:0x042B	0x3004 / 0x00 (M)	Ulong	RW
Calibration	Scale interval	0x0017	0x65/7	0x0026	R:0x022C W:0x022D	0x3003 / 0x00	Uint	RW
Calibration	Zero calibration	0x0018	0x65/8	0x0027	R:0x0434 W:0x0435	0x3005 / 0x00	Long	RW
Calibration	Span coefficient 1	0x001A	0x65/9	0x002B	R:0x0436 W:0x0437	0x3006 / 0x04	Float	RW
Calibration	Span coefficient 2	0x001C	0x65/10	0x002C	R:0x0438 W:0x0439	0x3006 / 0x05	Float	RW

Chapter	Name	Modbus RTU / TCP address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Calibration	**Span coefficient 3**	0x001E	0x65/11	0x002D	R:0x043A W:0x043B	0x3006 / 0x06	Float	RW
Calibration	**Span adjusting coefficient**	0x0020	0x65/12	0x0028	R:0x042E W:0x042F	0x3006 / 0x01 (M)	Ulong	RW
Calibration	**Calibration place g value**	0x0022	0x65/13	0x0029	R:0x0430 W:0x0431	0x3006 / 0x02	Ulong	RW
Calibration	**Place of use g value**	0x0024	0x65/14	0x002A	R:0x0432 W:0x0433	0x3006 / 0x03	Ulong	RW
	Reserved (12 bytes	0x0026						
Dosing	**Instant flow rate correction factor**	0x002C	0x69/2	0x0071	R: 0x0476 W: 0x0477	0x4700 / 0x02	Ulong	RW
	Réservé (8 octets)	0x002D						
Inputs/ outputs	**External value to control analog output (IO+ version)**	0x0032	0x67/16	0x005C	R:0x023C W:0x023D + See modules list	0x5050 / 0x00 (M)	Uint	RW
	Reserved (2 bytes)	0x0033						
HMI	**HMI name**	0x0034	0x64/21	0x00E0	/	0x3701 / 0x00	String	RW
Filtering	**A/D conversion rate**	0x0036	0x66/1	0x0030	R:0x0240 W:0x0241	0x4000 / 0x00	Uint	RW
Filtering	**filters activation**	0x0037 LSB	0x66/2 LSB	0x0031 LSB	R:0x0242 LSB W:0x0243 LSB	0x4001 / 0x01 (byte)	Byte	RW
Filtering	**Low-pass order**	0x0037 MSB	0x66/2 MSB	0x0031 MSB	R:0x0242 MSB W:0x0243 MSB	0x4001 / 0x02 (byte)	Byte	RW
Filtering	**Low-pass cut-off frequency**	0x0038	0x66/3	0x0032	R:0x0244 W:0x0245	0x4001 / 0x03	Uint	RW
Filtering	**Depth of moving average filter on weight**	0x0039	0x66/4	0x0033	R:0x0246 W:0x0247	0x4001 / 0x04	Uint	RW
Filtering	**Tolerance of clipping filter on instant flow rates**	0x003A	0x66/5	0x0034	R:0x0248 W:0x0249	0x4001 / 0x05	Uint	RW
	Reserved (6 bytes)	0x003B						
Protocol and functioning mode	**Functioning mode / Serial protocol AUX/USB**	0x003E	0x64/10	0x00A8	R: 0x02E8 W: 0x02E9	0x2000 / 0x00	Uint	RW
	Reserved (2 bytes)	0x003F						
Inputs/ outputs	**Analog output functioning (IO+ version)**	0x0040	0x67/15	0x005B	R:0x026C W:0x026D	0x4509 / 0x05	Uint	RW
Inputs/ outputs	**Logical input 3 functioning (IO+ version)**	0x0041 LSB	0x67/14 LSB	0x0042 LSB	R:0x026A LSB W: 0x026B LSB	0x4501 / 0x04	byte	RW
Inputs/ outputs	**Logical input 4 functioning (IO+ version)**	0x0041 MSB	0x67/14 MSB	0x0042 MSB	R:0x026A MSB W: 0x026B MSB	0x4501 / 0x05	byte	RW

Chapter	Name	Modbus RTU / TCP address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Inputs/ outputs	***Logical input 1 functioning***	*0x0042 LSB*	*0x67/1 LSB*	*0x0040 LSB*	*R:0x0250 LSB W: 0x0251 LSB*	*0x4501 / 0x02*	*byte*	*RW*
Inputs/ outputs	***Logical input 2 functioning***	*0x0042 MSB*	*0x67/1 MSB*	*0x0040 MSB*	*R: 0x0250 MSB W: 0x0251 MSB*	*0x4501 / 0x03*	*byte*	*RW*
Inputs/ outputs	***holding time***	*0x0043*	*0x67/2*	*0x0041*	*R:0x0252 W:0x0253*	*0x4501 / 0x01*	*Uint*	*RW*
Inputs/ outputs	***Output 1 functioning***	*0x0044 LSB*	*0x67/3 LSB*	*0x0050 LSB*	*R:0x0254 LSB W: 0x0255 LSB*	*0x4509 / 0x01*	*byte*	*RW*
Inputs/ outputs	***Output 2 functioning***	*0x0044 MSB*	*0x67/3 MSB*	*0x0050 MSB*	*R:0x0254 MSB W: 0x0255 MSB*	*0x4509 / 0x02*	*byte*	*RW*
Inputs/ outputs	***Output 3 functioning***	*0x0045 LSB*	*0x67/4 LSB*	*0x0051 LSB*	*R:0x0256 LSB W: 0x0257 LSB*	*0x4509 / 0x03*	*byte*	*RW*
Inputs/ outputs	***Output 4 functioning***	*0x0045 MSB*	*0x67/4 MSB*	*0x0051 MSB*	*R:0x0256 MSB W: 0x0257 MSB*	*0x4509 / 0x04*	*byte*	*RW*
Inputs/ outputs	***Set point 1 high value***	*0x0046*	*0x67/5*	*0x0052*	*R:0x045A W:0x045B*	*0x4601 / 0x02*	*Long*	*RW*
Inputs/ outputs	***Set point 1 low value***	*0x0048*	*0x67/6*	*0x0053*	*R:0x045C W:0x045D*	*0x4601 / 0x03*	*Long*	*RW*
Inputs/ outputs	***Set point 2 high value***	*0x004A*	*0x67/7*	*0x0054*	*R:0x045E W:0x045F*	*0x4601 / 0x04*	*Long*	*RW*
Inputs/ outputs	***Set point 2 low value***	*0x004C*	*0x67/8*	*0x0055*	*R:0x0460 W:0x0461*	*0x4601 / 0x05*	*Long*	*RW*
Inputs/ outputs	***Set point 3 high value***	*0x004E*	*0x67/9*	*0x0056*	*R:0x0462 W:0x0463*	*0x4609 / 0x02*	*Long*	*RW*
Inputs/ outputs	***Set point 3 low value***	*0x0050*	*0x67/10*	*0x0057*	*R:0x0464 W:0x0465*	*0x4609 / 0x03*	*Long*	*RW*
Inputs/ outputs	***Set point 4 high value***	*0x0052*	*0x67/11*	*0x0058*	*R:0x0466 W:0x0467*	*0x4609 / 0x04*	*Long*	*RW*
Inputs/ outputs	***Set point 4 low value***	*0x0054*	*0x67/12*	*0x0059*	*R:0x0468 W:0x0469*	*0x4609 / 0x05*	*Long*	*RW*
Inputs/ outputs	***1&2 Set points functioning***	*0x0056 LSB*	*0x67/13 LSB*	*0x005A LSB*	*R:0x0258 LSB W: 0x0259 LSB*	*0x4601 / 0x01*	*byte*	*RW*
Inputs/ outputs	***3&4 Set points functioning***	*0x0056 MSB*	*0x67/13 MSB*	*0x005A MSB*	*R:0x0258 MSB W: 0x0259 MSB*	*0x4609 / 0x01*	*byte*	*RW*
Inputs/ outputs	***Weight quantity per pulse on logical output***	*0x0057*	*0x69/29*	*0x008C*	*R:0x02AE W:0x02AF*	*0x4707 / 0x00*	*Uint*	*RW*
Dosing	***Instant flow rate determination time***	*0x0058*	*0x69/1*	*0x0070*	*R: 0x0274 W: 0x0275*	*0x4700 / 0x01*	*Uint*	*RW*
Filtering	***Average flow rate determination depth***	0x0059	*0x69/3*	*0x0072*	*R:0x0278 W:0x0279*	*0x4700 / 0x03*	*Uint*	*RW*

Chapter	Name	Modbus RTU / TCP address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Metrology	**Flow rate time unit**	0x005A	0x64/11	0x0016	R:0x021E W:0x021F	0x3700 / 0x03	Uint	RW
Measures	**Dynamic zero acquisition time**	0x005B	0x69/30	0x008D	R:0x02B0 W:0x02B1	0x4708 / 0x00	Uint	RW
Dosing	**Learning cycle end of refill delay**	0x005C	0x69/4	0x0073	R: 0x027A W: 0x027B	0x4701 / 0x01	Uint	RW
Dosing	**Learning cycle flow rates reference determination time**	0x005D	0x69/5	0x0074	R: 0x027C W: 0x027D	0x4701 / 0x02	Uint	RW
Dosing	**End of refill and cycle start stabilization time**	0x005E	0x69/6	0x0075	R: 0x027E W: 0x027F	0x4701 / 0x03	Uint	RW
Dosing	**Cycle and alarm options**	0x005F	0x69/7	0x0076	R: 0x0280 W: 0x0281	0x4701 / 0x04	Uint	RW
Dosing	**Refilling low level**	0x0060	0x69/8	0x0077	R: 0x0482 W: 0x0483	0x4701 / 0x05	Ulong	RW
Dosing	**Refilling high level**	0x0062	0x69/9	0x0078	R: 0x0484 W: 0x0485	0x4701 / 0x06	Ulong	RW
Dosing	**Refilling inflight value**	0x0064	0x69/10	0x0079	R: 0x0486 W: 0x0487	0x4701 / 0x07	long	RW
Dosing	**Fixed flow rate during refilling**	0x0066	0x69/11	0x007A	R: 0x0488 W: 0x0489	0x4701 / 0x08	float	RW
Dosing	**Empty hopper level**	0x0068	0x69/14	0x007D	R: 0x048E W: 0x048F	0x4702 / 0x01	Ulong	RW
	Reserved (2 bytes)	0x006A						
Dosing	**Maximum time for refilling start**	0x006B	0x69/15	0x007E	R: 0x0290 W: 0x0291	0x4703 / 0x01	Uint	RW
Dosing	**Time interval for weight variation control on refilling**	0x006C	0x69/16	0x007F	R: 0x0294 W: 0x0295	0x4703 / 0x02	Uint	RW
Dosing	**Maximum refilling time**	0x006D	0x69/17	0x0080	R: 0x0296 W: 0x0297	0x4703 / 0x03	Uint	RW
Dosing	**Min permissible instant flow rate**	0x006E	0x69/18	0x0081	R: 0x0298 W: 0x0299	0x4704 / 0x01	Uint	RW
Dosing	**Max permissible instant flow rate**	0x006F	0x69/19	0x0082	R: 0x029A W: 0x029B	0x4704 / 0x02	Uint	RW
Dosing	**Min permissible flow rate control output**	0x0070	0x69/20	0x0083	R: 0x029C W: 0x029D	0x4704 / 0x03	Uint	RW

Chapter	Name	Modbus RTU / TCP address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Dosing	***Max permissible flow rate control output***	*0x0071*	*0x69/21*	*0x0084*	*R: 0x029E* *W: 0x029F*	*0x4704 / 0x04*	*Uint*	*RW*
Dosing	***Inhibit time of flow rates alarms at start***	*0x0072*	*0x69/22*	*0x0085*	*R: 0x02A0* *W: 0x02A1*	*0x4704 / 0x05*	*Uint*	*RW*
Dosing	***Inhibit time of flow rates alarms in service***	*0x0073*	*0x69/23*	*0x0086*	*R: 0x02A2* *W: 0x02A3*	*0x4704 / 0x06*	*Uint*	*RW*
Dosing	***Maximum batch time***	*0x0074*	*0x69/24*	*0x0087*	*R: 0x02A4* *W: 0x02A5*	*0x4704 / 0x07*	*Uint*	*RW*
Dosing	***Weight to totalize (Great WU)***	*0x0075*	*0x69/25*	*0x0088*	*R: 0x04A6* *W: 0x04A7*	*0x4705 / 0x01*	*Ulong*	*RW*
Dosing	***Complementary weight to totalize***	*0x0077*	*0x69/26*	*0x0089*	*R: 0x02A8* *W: 0x02A9*	*0x4705 / 0x02*	*Uint*	*RW*
Dosing	***Weigth to totalize inflight value***	*0x0078*	*0x69/27*	*0x008A*	*R: 0x04AA* *W: 0x04AB*	*0x4705 / 0x03*	*long*	*RW*
Dosing	***Nominal flow rate***	*0x007A*	*0x69/28*	*0x008B*	*R: 0x04AC* *W: 0x4AD*	*0x4706 / 0x00*	*float*	*RW*
	Reserved (2 bytes)	*0x007C*						
Measures	***Measurement status***	*0x007D*	*/*	*/*	*See modules list*	*0x5003 / 0x00 (M)*	*Uint*	*RO*
Measures	***Gross measurement***	*0x007E*	*/*	*/*	*See modules list*	*0x5001 / 0x00 (M)*	*Long*	*RO*
Measures	***Tare value***	*0x0080*	*/*	*0x0060*	*R:0x0470* *W: /*	*0x5004 / 0x01 (M)*	*Long*	*RO*
Measures	***Net measurement***	*0x0082*	*/*	*/*	*See modules list*	*0x5000 / 0x00 (M)*	*Long*	*RO*
Measures	***Factory calibrated points***	*0x0084*	*/*	*/*	*See modules list*	*0x5002 / 0x00 (M)*	*Long*	*RO*
Dosing	***Instant flow rate***	*0x0086*	*/*	*0x00BF*	*R: 0x0400* *W: /* *+ See modules list*	*0x5005 / 0x01 (M)*	*float*	*RO*
Dosing	***Average flow rate***	*0x0088*	*/*	*0x00B0*	*R: 0x046E* *W: /* *+ See modules list*	*0x5005 / 0x02 (M)*	*float*	*RO*
Dosing	***Flow rate control output***	*0x008A*	*/*	*0x00B1*	*R: 0x04EA* *W: /* *+ See modules list*	*0x5005 / 0x03 (M)*	*float*	*RO*

Chapter	Name	Modbus RTU / TCP address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Dosing	**Control output value**	0x008C	/	0x00B2	R: 0x02EC W: / + See modules list	0x5005 / 0x04 (M)	Uint	RO
Dosing	**Totalizer value (Great WU)**	0x008D	/	0x00B4	R: 0x04F0 W: / + See modules list	0x5006 / 0x01 (M)	Ulong	RO
Dosing	**Complementary totalizer value**	0x008F	/	0x00B5	R: 0x02F2 W: / + See modules list	0x5006 / 0x02 (M)	Uint	RO
Commands	**Command register**	0x0090	0x68/1	/	See modules list	0x2003 / 0x00 (M)	Uint	RW
Commands	**Response register**	0x0091	0x68/2	/	See modules list	0x2004 / 0x00 (M)	Uint	RO
Calibration	**Zero offset**	0x0092	0x65/15	0x002E	R:0x0472 W:0x0473	0x2500 / 0x00 (M)	Long	RW
Inputs/ outputs	**Logical input level**	0x0094 LSB	/	/	See modules list	0x5100 / 0x00 (M)	Byte	RO
Inputs/ outputs	**Logical ouput level**	0x0094 MSB	/	/	See modules list	0x5200 / 0x00 (M)	Byte	RO
Dosing	**Dosing weight deviation**	0x0095	/	0x00B9	R: 0x04FA W: / + See modules list	0x5007 / 0x05 (M)	float	RO
Measures	**Preset Tare**	0x0097	0x65/16	0x0061	R:0x04C4 W:0x04C5	0x5004 / 0x02 (M)	Ulong	RW
	Reserved (2 bytes)	0x0099						
Dosing	**Dosing status**	0x009A	/	0x00BA	R: 0x02FC W: / + See modules list	0x5007 / 0x01 (M)	Uint	RO
Dosing	**Dosing errors report**	0x009B	/	0x00BB	R: 0x02FE W: / + See modules list	0x5007 / 0x02 (M)	Uint	RO
Dosing	**Dosing errors counter**	0x009C	/	0x00BC	R: 0x023E W: / + See modules list	0x5007 / 0x03 (M)	Uint	RO

Chapter	Name	Modbus RTU / TCP address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Dosing	***Last dosing error***	*0x009D*	*/*	*0x00BD*	*R: 0x024A* *W: /* *+ See modules list*	*0x5007 / 0x04 (M)*	*Uint*	*RO*
Dosing	***Dosing quality factor***	*0x009E*	*/*	*0x00B3*	*R: 0x04EE* *W: /* *+ See modules list*	*0x5005 / 0x05 (M)*	*float*	*RO*
Dosing	***Totalization flow rate***	*0x00A0*	*/*	*0x00B8*	*R: 0x04F8* *W: /* *+ See modules list*	*0x5006 / 0x03 (M)*	*float*	*RO*
Dosing	***Grand total (Great WU)***	*0x00A2*	*/*	*0x00B6*	*R: 0x04F4* *W: /* *+ See modules list*	*0x5006 / 0x04*	*Ulong*	*RO*
Dosing	***General total (Great WU)***	*0x00A4*	*/*	*0x00B7*	*R: 0x04F6* *W: /* *+ See modules list*	*0x5006 / 0x05*	*Ulong*	*RO*
Dosing	***Extraction time***	*0x00A6*	*/*	*0x00BE*	*R: 0x0402* *W: /*	*0x5007 / 0x06*	*float*	*RO*
	Reserved (16 bytes)	*0x00AE*						

*************************** *MODBUS ADDRESS TABLE JUMP* ************************************

Chapter	Name	Modbus RTU / TCP address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Dosing	***Calibration of flow rate point 1 (control output value)***	*0x0A00*	*0x69/32*	*0x008F*	*R: 0x02B4* *W: 0x02B5*	*0x470A / 0x01*	*Uint*	*RW*
Dosing	***Calibration of flow rate point 2 (control output value)***	*0x0A01*	*0x69/33*	*0x0090*	*R: 0x02B6* *W: 0x02B7*	*0x470A / 0x02*	*Uint*	*RW*
Dosing	***Calibration of flow rate point 3 (control output value)***	*0x0A02*	*0x69/34*	*0x0091*	*R: 0x02B8* *W: 0x02B9*	*0x470A / 0x03*	*Uint*	*RW*
Dosing	***Calibration of flow rate point 4 (control output value)***	*0x0A03*	*0x69/35*	*0x0092*	*R: 0x02BA* *W: 0x02BB*	*0x470A / 0x04*	*Uint*	*RW*
Dosing	***Calibration of flow rate point 5 (control output value)***	*0x0A04*	*0x69/36*	*0x0093*	*R: 0x02BC* *W: 0x02BD*	*0x470A / 0x05*	*Uint*	*RW*
Dosing	***Calibration of flow rate point 6 (control output value)***	*0x0A05*	*0x69/37*	*0x0094*	*R: 0x02BE* *W: 0x02BF*	*0x470A / 0x06*	*Uint*	*RW*
Dosing	***Calibration of flow rate point 7 (control output value)***	*0x0A06*	*0x69/38*	*0x0095*	*R: 0x02C0* *W: 0x02C1*	*0x470A / 0x07*	*Uint*	*RW*
Dosing	***Calibration of flow rate point 8 (control output value)***	*0x0A07*	*0x69/39*	*0x0096*	*R: 0x02C2* *W: 0x02C3*	*0x470A / 0x08*	*Uint*	*RW*
Dosing	***Calibration of flow rate point 9 (control output value)***	*0x0A08*	*0x69/40*	*0x0097*	*R: 0x02C6* *W: 0x02C7*	*0x470A / 0x09*	*Uint*	*RW*
Dosing	***Calibration of flow rate point 10 (control output value)***	*0x0A09*	*0x69/41*	*0x0098*	*R: 0x02C8* *W: 0x02C9*	*0x470A / 0x0A*	*Uint*	*RW*
Dosing	***Calibration of flow rate point 1 (flow rate value)***	*0x0A0A*	*0x69/42*	*0x0099*	*R: 0x04CA* *W: 0x04CB*	*0x470B / 0x01*	*float*	*RW*

Chapter	Name	Modbus RTU / TCP address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Dosing	*Calibration of flow rate point 2 (flow rate value)*	*0x0A0C*	*0x69/43*	*0x009A*	*R: 0x04CC* *W: 0x04CD*	*0x470B / 0x02*	*float*	*RW*
Dosing	*Calibration of flow rate point 3 (flow rate value)*	*0x0A0E*	*0x69/44*	*0x009B*	*R: 0x04CE* *W: 0x04CF*	*0x470B / 0x03*	*float*	*RW*
Dosing	*Calibration of flow rate point 4 (flow rate value)*	*0x0A10*	*0x69/45*	*0x009C*	*R: 0x04D0* *W: 0x04D1*	*0x470B / 0x04*	*float*	*RW*
Dosing	*Calibration of flow rate point 5 (flow rate value)*	*0x0A12*	*0x69/46*	*0x009D*	*R: 0x04D2* *W: 0x04D3*	*0x470B / 0x05*	*float*	*RW*
Dosing	*Calibration of flow rate point 6 (flow rate value)*	*0x0A14*	*0x69/47*	*0x009E*	*R: 0x04D4* *W: 0x04D5*	*0x470B / 0x06*	*float*	*RW*
Dosing	*Calibration of flow rate point 7 (flow rate value)*	*0x0A16*	*0x69/48*	*0x009F*	*R: 0x04D6* *W: 0x04D7*	*0x470B / 0x07*	*float*	*RW*
Dosing	*Calibration of flow rate point 8 (flow rate value)*	*0x0A18*	*0x69/49*	*0x00A0*	*R: 0x04D8* *W: 0x04D9*	*0x470B / 0x08*	*float*	*RW*
Dosing	*Calibration of flow rate point 9 (flow rate value)*	*0x0A1A*	*0x69/50*	*0x00A1*	*R: 0x04DA* *W: 0x04DB*	*0x470B / 0x09*	*float*	*RW*
Dosing	*Calibration of flow rate point 10 (flow rate value)*	*0x0A1C*	*0x69/51*	*0x00A2*	*R: 0x04DC* *W: 0x04DD*	*0x470B / 0x0A*	*float*	*RW*
Dosing	*Segments number for the calibration curve of flow rate*	*0x0A1E*	*0x69/31*	*0x008E*	*R: 0x02B2* *W: 0x02B3*	*0x4709 / 0x00*	*Uint*	*RW*
	Reserved (36 bytes)	*0x0A1F*						
Dosing	*Reference flow rate control output start refilling*	*0x0A31*	*0x69/12*	*0x007B*	*R: 0x048A* *W: 0x048B*	*0x4701 / 0x09*	*float*	*RW*
Dosing	*Reference flow rate control output end refilling*	*0x0A33*	*0x69/13*	*0x007C*	*R: 0x048C* *W: 0x048D*	*0x4701 / 0x0A*	*float*	*RW*
Dosing	*Kp*	*0x0A35*	*0x69/52*	*0x00A3*	*R: 0x04DE* *W: 0x04DF*	*0x470C / 0x01*	*float*	*RW*

Chapter	Name	Modbus RTU / TCP address	EtherNet/IP Class/ Attribute (hex/dec)	Profinet Record Index	Profinet cyclic Req Code	EtherCAT index/sub-index	Type	Access
Dosing	***Ti***	*0x0A37*	*0x69/53*	*0x00A4*	*R: 0x04E0* *W: 0x04E1*	*0x470C / 0x02*	*Ulong*	*RW*
Dosing	***Td***	*0x0A39*	*0x69/54*	*0x00A5*	*R: 0x04E2* *W: 0x04E3*	*0x470C / 0x03*	*Ulong*	*RW*
Dosing	***PID behavior***	*0x0A3B*	*0x69/55*	*0x00A6*	*R: 0x02E4* *W: 0x02E5*	*0x470C / 0x04*	*Uint*	*RW*
Dosing	***PID adjustment flow rate***	*0x0A3C*	*0x69/56*	*0x00A7*	*R: 0x04E6* *W: 0x04E7*	*0x470C / 0x05*	*float*	*RW*
Dosing	***Full hopper level***	*0x0A3E*	*0x69/57*	*0x00A9*	*R: 0x0404* *W: 0x0405*	*0x4702 / 0x02*	*Ulong*	*RW*
	Reserved (8 bytes)	*0x0A40*						
Measures	***Sensor input control reference***	*0x0A44*	*0x65/17*	*0x0062*	*R: 0x044C* *W: 0x044D*	*0x5004 / 0x03*	*long*	*RW*
Measures	***Sensor input control result***	*0x0A46*	*0x68/4*	*0x0063*	*R: 0x024E* *W: /*	*0x5004 / 0x04*	*Int*	*RO*
Measures	***Sensor input control result max. tolerance***	*0x0A47*	*0x65/18*	*0x0064*	*R: 0x020A* *W: 0x020B*	*0x5004 / 0x05*	*Uint*	*RW*
Measures & Inputs/ outputs	***Defective measurement debounced time***	*0x0A48*	*0x67/17*	*0x005D*	*R:0x0206* *W:0x0207*	*0x4509/0x06*	*Uint*	*RW*
Measures & Inputs/ outputs	***Defective measurement alarm activation time***	*0x0A49*	*0x67/18*	*0x005E*	*R:0x0208* *W:0x0209*	*0x4509/0x07*	*Uint*	*RW*
Dosing	***Number of cycles for adjustment PID***	*0x0A50*	*0x69/58*	*0x00E1*	*/*	*0x470C / 0x06*	*Uint*	*RW*
Measures	***Cut-off frequency for totalization flow rate (x100)***	*0x0A51*	*0x69/59*	*0x00E2*	*/*	*0x4001/ 0x06*	*Uint*	*RW*
Measures	***Time unit for totalization flow rate***	*0x0A52*	*0x69/60*	*0x00E3*	*/*	*0x3700 / 0x04*	*Uint*	*RW*
Measures	***Weight unit for totalization flow rate***	*0x0A53*	*0x69/61*	*0x00E4*	*/*	*0x3700 / 0x05*	*Ulong*	*RW*
	Reserved (8 bytes)	*0x0A4A*						

17 PROFINET IO REGISTERS

Chapter	Name	Access	Size in bytes	Standard read / Write parameters via Profinet RPC Record Index	Substitute for Read Parameters via Profinet Cyclic Transaction Req	Substitute for Write Parameters via Profinet Cyclic Transaction Req
Metrology	Legal for trade switch and version	RW	2	0x0010	0x0210	0x0211
Metrology	Legal for trade counter	RO	2	0x0011	0x0212	/
Metrology	Legal for trade checksum	RO	2	0x0012	0x0214	/
Metrology	Zero functions	RW	2	0x0013	0x0216	0x0217
Metrology	Stability criterion / decimal point position	RW	2	0x0014	0x0218	0x0219
Metrology	Weight unit	RW	4	0x0015	0x041A	0x041B
Metrology	Flow rate time unit	RW	4	0x0016	0x021E	0x021F
Calibration	Maximum capacity	RW	4	0x0020	0x0420	0x0421
Calibration	Number of calibration segments	RW	2	0x0021	0x0222	0x0223
Calibration	Calibration load 1	RW	4	0x0022	0x0424	0x0425
Calibration	Calibration load 2	RW	4	0x0023	0x0426	0x0427
Calibration	Calibration load 3	RW	4	0x0024	0x0428	0x0429
Calibration	Sensor sensitivity	RW	4	0x0025	0x042A	0x042B
Calibration	Scale interval	RW	2	0x0026	0x022C	0x022D
Calibration	Zero calibration	RW	4	0x0027	0x0434	0x0435
Calibration	Span adjusting coefficient	RW	4	0x0028	0x042E	0x042F
Calibration	Calibration place g value	RW	4	0x0029	0x0430	0x0431
Calibration	Place of use g value	RW	4	0x002A	0x0432	0x0433
Calibration	Span coefficient 1	RW	4	0x002B	0x0436	0x0437
Calibration	Span coefficient 2	RW	4	0x002C	0x0438	0x0439
Calibration	Span coefficient 3	RW	4	0x002D	0x043A	0x043B
Calibration	Zero offset	RW	4	0x002E	0x0472	0x0473
Filtering	A/D conversion rate	RW	2	0x0030	0x0240	0x0241
Filtering	Low-pass order / filters activation	RW	2	0x0031	0x0242	0x0243
Filtering	Low-pass cut-off frequency	RW	2	0x0032	0x0244	0x0245
Filtering	Depth of moving average filter on weights	RW	2	0x0033	0x0246	0x0247

Chapter	Name	Access	Size in bytes	Standard read / Write parameters via Profinet RPC Record Index	Substitute for Read Parameters via Profinet Cyclic Transaction Req	Substitute for Write Parameters via Profinet Cyclic Transaction Req
Filtering	**Tolerance of clipping filter on instant flow rates**	RW	2	0x0034	0x0248	0x0249
Inputs/ outputs	**Logical inputs 1 & 2 functioning**	RW	2	0x0040	0x0250	0x0251
Inputs/ outputs	**holding time**	RW	2	0x0041	0x0252	0x0253
Inputs/ outputs	**Logical inputs 3 & 4 functioning (optional)**	RW	2	0x0042	0x026A	0x026B
Inputs/ outputs	**Outputs 1 & 2 functioning**	RW	2	0x0050	0x0254	0x0255
Inputs/ outputs	**Outputs 3 & 4 functioning**	RW	2	0x0051	0x0256	0x0257
Inputs/ outputs	**Set point 1 high value**	RW	4	0x0052	0x045A	0x045B
Inputs/ outputs	**Set point 1 low value**	RW	4	0x0053	0x045C	0x045D
Inputs/ outputs	**Set point 2 high value**	RW	4	0x0054	0x045E	0x045F
Inputs/ outputs	**Set point 2 low value**	RW	4	0x0055	0x0460	0x0461
Inputs/ outputs	**Set point 3 high value**	RW	4	0x0056	0x0462	0x0463
Inputs/ outputs	**Set point 3 low value**	RW	4	0x0057	0x0464	0x0465
Inputs/ outputs	**Set point 4 high value**	RW	4	0x0058	0x0466	0x0467
Inputs/ outputs	**Set point 4 low value**	RW	4	0x0059	0x0468	0x0469
Inputs/ outputs	**Set points functioning**	RW	2	0x005A	0x0258	0x0259
Inputs/ outputs	**Analog output functioning (optional)**	RW	2	0x005B	0x026C	0x026D
Inputs/ outputs	**External value to control analog output**	RW	2	0x005C	0x023C	0x023D

Chapter	Name	Access	Size in bytes	Standard read / Write parameters via Profinet RPC Record Index	Substitute for Read Parameters via Profinet Cyclic Transaction Req	Substitute for Write Parameters via Profinet Cyclic Transaction Req
Inputs/ outputs	***Defective measurement debounced time***	*RW*	*2*	*0x005D*	*0x0206*	*0x0207*
Inputs/ outputs	***Defective measurement alarm activation time***	*RW*	*2*	*0x005E*	*0x0208*	*0x0209*
Measures	***Tare value***	*RO*	*4*	*0x0060*	*0x0470*	*/*
Measures	***Preset tare value***	*RW*	*4*	*0x0061*	*0x04C4*	*0x04C5*
Measures	***Sensor input control reference***	*RW*	*4*	*0x0062*	*0x044C*	*0x044D*
Measures	***Sensor input control result***	*RO*	*2*	*0x0063*	*0x024E*	*/*
Measures	***Sensor input control result max. tolerance***	*RW*	*2*	*0x0064*	*0x020A*	*0x020B*
Dosing	***Instant flow rate determination time***	*RW*	*2*	*0x0070*	*0x0274*	*0x0275*
Dosing	***Instant flow rate correction factor***	*RW*	*4*	*0x0071*	*0x0476*	*0x0477*
Dosing	***Average flow rate determination depth***	*RW*	*2*	*0x0072*	*0x0278*	*0x0279*
Dosing	***Learning cycle end of refill delay***	*RW*	*2*	*0x0073*	*0x027A*	*0x027B*
Dosing	***Learning cycle flow rates reference determination time***	*RW*	*2*	*0x0074*	*0x027C*	*0x027D*
Dosing	***End of refill and cycle start stabilization time***	*RW*	*2*	*0x0075*	*0x027E*	*0x027F*
Dosing	***Cycle and alarm options***	*RW*	*2*	*0x0076*	*0x0280*	*0x0281*
Dosing	***Refilling low level***	*RW*	*4*	*0x0077*	*0x0482*	*0x0483*
Dosing	***Refilling high level***	*RW*	*4*	*0x0078*	*0x0484*	*0x0485*
Dosing	***Refilling inflight value***	*RW*	*4*	*0x0079*	*0x0486*	*0x0487*
Dosing	***Fixed flow rate during refilling***	*RW*	*4*	*0x007A*	*0x0488*	*0x0489*
Dosing	***Reference flow rate control output start refilling***	*RW*	*4*	*0x007B*	*0x048A*	*0x048B*
Dosing	***Reference flow rate control output end refilling***	*RW*	*4*	*0x007C*	*0x048C*	*0x048D*
Dosing	***Empty hopper level***	*RW*	*4*	*0x007D*	*0x048E*	*0x048F*
Dosing	***Maximum time for refilling start***	*RW*	*2*	*0x007E*	*0x0290*	*0x0291*
Dosing	***Time interval for weight variation control on refilling***	*RW*	*2*	*0x007F*	*0x0294*	*0x0295*

Chapter	Name	Access	Size in bytes	Standard read / Write parameters via Profinet RPC Record Index	Substitute for Read Parameters via Profinet Cyclic Transaction Req	Substitute for Write Parameters via Profinet Cyclic Transaction Req
Dosing	***Maximum refilling time***	*RW*	*2*	*0x0080*	*0x0296*	*0x0297*
Dosing	***Min permissible instant flow rate***	*RW*	*2*	*0x0081*	*0x0298*	*0x0299*
Dosing	***Max permissible instant flow rate***	*RW*	*2*	*0x0082*	*0x029A*	*0x029B*
Dosing	***Min permissible flow rate control output***	*RW*	*2*	*0x0083*	*0x029C*	*0x029D*
Dosing	***Max permissible flow rate control output***	*RW*	*2*	*0x0084*	*0x029E*	*0x029F*
Dosing	***Inhibit time of flow rates alarms at start***	*RW*	*2*	*0x0085*	*0x02A0*	*0x02A1*
Dosing	***Inhibit time of flow rates alarms in service***	*RW*	*2*	*0x0086*	*0x02A2*	*0x02A3*
Dosing	***Maximum batch time***	*RW*	*2*	*0x0087*	*0x02A4*	*0x02A5*
Dosing	***Weight to totalize (Great WU)***	*RW*	*4*	*0x0088*	*0x04A6*	*0x04A7*
Dosing	***Complementary weight to totalize***	*RW*	*2*	*0x0089*	*0x02A8*	*0x02A9*
Dosing	***Weigth to totalize inflight value***	*RW*	*4*	*0x008A*	*0x04AA*	*0x04AB*
Dosing	***Nominal flow rate***	*RW*	*4*	*0x008B*	*0x04AC*	*0x04AD*
Dosing	***Weight quantity per pulse on logical output***	*RW*	*2*	*0x008C*	*0x02AE*	*0x02AF*
Measures	***Dynamic zero acquisition time***	*RW*	*2*	*0x008D*	*0x02B0*	*0x02B1*
Dosing	***Segments number for the calibration curve of flow rate***	*RW*	*2*	*0x008E*	*0x02B2*	*0x02B3*
Dosing	***Calibration of flow rate point 1 (control output value)***	*RW*	*2*	*0x008F*	*0x02B4*	*0x02B5*
Dosing	***Calibration of flow rate point 2 (control output value)***	*RW*	*2*	*0x0090*	*0x02B6*	*0x02B7*
Dosing	***Calibration of flow rate point 3 (control output value)***	*RW*	*2*	*0x0091*	*0x02B8*	*0x02B9*
Dosing	***Calibration of flow rate point 4 (control output value)***	*RW*	*2*	*0x0092*	*0x02BA*	*0x02BB*
Dosing	***Calibration of flow rate point 5 (control output value)***	*RW*	*2*	*0x0093*	*0x02BC*	*0x02BD*
Dosing	***Calibration of flow rate point 6 (control output value)***	*RW*	*2*	*0x0094*	*0x02BE*	*0x02BF*
Dosing	***Calibration of flow rate point 7 (control output value)***	*RW*	*2*	*0x0095*	*0x02C0*	*0x02C1*
Dosing	***Calibration of flow rate point 8 (control output value)***	*RW*	*2*	*0x0096*	*0x02C2*	*0x02C3*

Chapter	Name	Access	Size in bytes	Standard read / Write parameters via Profinet RPC Record Index	Substitute for Read Parameters via Profinet Cyclic Transaction Req	Substitute for Write Parameters via Profinet Cyclic Transaction Req
Dosing	**Calibration of flow rate point 9 (control output value)**	RW	2	0x0097	0x02C6	0x02C7
Dosing	**Calibration of flow rate point 10 (control output value)**	RW	2	0x0098	0x02C8	0x02C9
Dosing	**Calibration of flow rate point 1 (flow rate value)**	RW	4	0x0099	0x04CA	0x04CB
Dosing	**Calibration of flow rate point 2 (flow rate value)**	RW	4	0x009A	0x04CC	0x04CD
Dosing	**Calibration of flow rate point 3 (flow rate value)**	RW	4	0x009B	0x04CE	0x04CF
Dosing	**Calibration of flow rate point 4 (flow rate value)**	RW	4	0x009C	0x04D0	0x04D1
Dosing	**Calibration of flow rate point 5 (flow rate value)**	RW	4	0x009D	0x04D2	0x04D3
Dosing	**Calibration of flow rate point 6 (flow rate value)**	RW	4	0x009E	0x04D4	0x04D5
Dosing	**Calibration of flow rate point 7 (flow rate value)**	RW	4	0x009F	0x04D6	0x04D7
Dosing	**Calibration of flow rate point 8 (flow rate value)**	RW	4	0x00A0	0x04D8	0x04D9
Dosing	**Calibration of flow rate point 9 (flow rate value)**	RW	4	0x00A1	0x04DA	0x04DB
Dosing	**Calibration of flow rate point 10 (flow rate value)**	RW	4	0x00A2	0x04DC	0x04DD
Dosing	**Kp**	RW	4	0x00A3	0x04DE	0x04DF
Dosing	**Ti**	RW	4	0x00A4	0x04E0	0x04E1
Dosing	**Td**	RW	4	0x00A5	0x04E2	0x04E3
Dosing	**PID behavior**	RW	2	0x00A6	0x02E4	0x02E5
Dosing	**PID adjustment Flow Rate**	RW	4	0x00A7	0x04E6	0x04E7
Dosing	**Functioning mode / Serial protocol**	RW	2	0x00A8	0x02E8	0x02E9
Dosing	**Full hopper level**	RW	4	0x00A9	0x0404	0x0405
Dosing	**Average flow rate**	RO	4	0x00B0	0x046E	/
Dosing	**Flow rate control output**	RO	4	0x00B1	0x04EA	/
Dosing	**Control output value**	RO	2	0x00B2	0x02EC	/

Chapter	Name	Access	Size in bytes	Standard read / Write parameters via Profinet RPC Record Index	Substitute for Read Parameters via Profinet Cyclic Transaction Req	Substitute for Write Parameters via Profinet Cyclic Transaction Req
Dosing	**Dosing quality factor**	RO	4	0x00B3	0x04EE	/
Dosing	**Totalizer value (Great WU)**	RO	4	0x00B4	0x04F0	/
Dosing	**Complementary totalizer value**	RO	2	0x00B5	0x02F2	/
Dosing	**Grand total (Great WU)**	RO	4	0x00B6	0x04F4	/
Dosing	**General total (Great WU)**	RO	4	0x00B7	0x04F6	/
Dosing	**Totalization flow rate**	RO	4	0x00B8	0x04F8	/
Dosing	**Dosing weight deviation**	RO	4	0x00B9	0x04FA	/
Dosing	**Dosing status**	RO	2	0x00BA	0x02FC	/
Dosing	**Dosing errors report**	RO	2	0x00BB	0x02FE	/
Dosing	**Dosing errors counter**	RO	2	0x00BC	0x023E	/
Dosing	**Last dosing error**	RO	2	0x00BD	0x024A	/
Dosing	**Extraction time**	RO	4	0x00BE	0x0402	/
Dosing	**Instant flow rate**	RO	4	0x00BF	0x0400	/
Dosing	**Number of cycles for adjustment PID**	RW	2	0x00E1	/	/
Dosing	**Cut-off frequency for totalization flow rate (x100)**	RW	2	0x00E2	/	/
Dosing	**Time unit for totalization flow rate**	RW	2	0x00E3	/	/
Dosing	**Weight unit for totalization flow rate**	RW	4	0x00E4	/	/
HMI	**HMI name**	RW	4	0x00E0	/	/

Error Type	*Diagnostic Name*	*Diagnostic Help*
4197	*Input analog signal out of the A/D conversion range (negative quadrant)*	*Possible Cause: Short circuit on sensor connection.*
4198	*Input analog signal out of the A/D conversion range (positive quadrant)*	*Possible Cause: Short circuit on sensor connection.*
4199	*Gross meas. < (- max capacity)*	*Cause: The value of the gross measurement exceeds the opposed maximum capacity minus 9 divisions.*
4200	*Gross meas. > (max capacity)*	*Cause: The value of the gross measurement exceeds the maximum capacity plus 9 divisions.*
4201	*Default EEPROM*	*Cause: Error of checksum while reading EEPROM after reset.*

18 ETHERNET/IP REGISTERS TABLE

Chapter	Name	EtherNet/IP Class	EtherNet/IP Attribute (dec)	Type	Service
	Class 0x64 (100d) / Instance 1				**Get Attribute All**
Modbus	**Firmware revision**	0x64	1	Uint	Get Attribute Single
Modbus	**Node number / baud rate**	0x64	2	Uint	Get Attribute Single
Metrology	**Legal for trade version**	0x64	3	Byte	Get Attribute Single
Metrology	**Legal for trade switch**	0x64	4	Byte	Get Attribute Single / Set Attribute Single
Metrology	**Legal for trade counter**	0x64	5	Byte	Get Attribute Single
Metrology	**Legal for trade checksum**	0x64	6	Uint	Get Attribute Single
Metrology	**Zero functions**	0x64	7	Uint	Get Attribute Single / Set Attribute Single
Metrology	**Stability criterion**	0x64	8 LSB	Byte	Get Attribute Single / Set Attribute Single
Metrology	**decimal point position**	0x64	8 MSB	Byte	Get Attribute Single / Set Attribute Single
Metrology	**Unit**	0x64	9	String	Get Attribute Single / Set Attribute Single
Protocols and modes	**Functioning mode**	0x64	10	Byte	Get Attribute Single / Set Attribute Single
Dosing	**Flow rate time unit**	0x64	11	Uint	Get Attribute Single / Set Attribute Single
HMI	**HMI name**	0x64	21	String	Get Attribute Single / Set Attribute Single
	0x65 (101d) / Instance 1				**Get Attribute All / Set Attribute All**
Calibration	**Maximum capacity**	0x65	1	Ulong	Get Attribute Single / Set Attribute Single
Calibration	**Number of calibration segments**	0x65	2	Uint	Get Attribute Single / Set Attribute Single
Calibration	**Calibration load 1**	0x65	3	Ulong	Get Attribute Single / Set Attribute Single
Calibration	**Calibration load 2**	0x65	4	Ulong	Get Attribute Single / Set Attribute Single
Calibration	**Calibration load 3**	0x65	5	Ulong	Get Attribute Single / Set Attribute Single
Calibration	**Sensor sensitivity**	0x65	6	Ulong	Get Attribute Single / Set Attribute Single
Calibration	**Scale interval**	0x65	7	Uint	Get Attribute Single / Set Attribute Single
Calibration	**Zero calibration**	0x65	8	Long	Get Attribute Single / Set Attribute Single
Calibration	**Span coefficient 1**	0x65	9	Float	Get Attribute Single / Set Attribute Single

Chapter	Name	EtherNet/IP Class	EtherNet/IP Attribute (dec)	Type	Service
Calibration	***Span coefficient 2***	*0x65*	*10*	*Float*	*Get Attribute Single / Set Attribute Single*
Calibration	***Span coefficient 3***	*0x65*	*11*	*Float*	*Get Attribute Single / Set Attribute Single*
Calibration	***Span adjusting coefficient***	*0x65*	*12*	*Ulong*	*Get Attribute Single / Set Attribute Single*
Calibration	***Calibration place g value***	*0x65*	*13*	*Ulong*	*Get Attribute Single / Set Attribute Single*
Calibration	***Place of use g value***	*0x65*	*14*	*Ulong*	*Get Attribute Single / Set Attribute Single*
Calibration	***Zero offset***	*0x65*	*15*	*Long*	*Get Attribute Single / Set Attribute Single*
Measures	***Preset tare value***	*0x65*	*16*	*Long*	*Get Attribute Single / Set Attribute Single*
Measures	***Sensor input control reference***	*0x65*	*17*	*Long*	*Get Attribute Single / Set Attribute Single*
Measures	***Sensor input control result max. tolerance***	*0x65*	*18*	*Uint*	*Get Attribute Single / Set Attribute Single*
	0x66 (102d) / Instance 1				***Get Attribute All / Set Attribute All***
Filtering	***A/D conversion rate***	*0x66*	*1*	*Uint*	*Get Attribute Single / Set Attribute Single*
Filtering	***filters activation***	*0x66*	*2 LSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Filtering	***Low-pass order***	*0x66*	*2 MSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Filtering	***Low-pass cut-off frequency***	*0x66*	*3*	*Uint*	*Get Attribute Single / Set Attribute Single*
Filtering	***Depth of moving average filter on weights***	*0x66*	*4*	*Uint*	*Get Attribute Single / Set Attribute Single*
Filtering	***Tolerance of clipping filter on instant flow rates***	*0x66*	*5*	*Uint*	*Get Attribute Single / Set Attribute Single*
	Class 0x67 (103d) / Instance 1				***Get Attribute All / Set Attribute All***
Inputs/ outputs	***Logical input 1 functioning***	*0x67*	*1 LSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Logical input 2 functioning***	*0x67*	*1 MSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***holding time***	*0x67*	*2*	*Uint*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Output 1 functioning***	*0x67*	*3 LSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Output 2 functioning***	*0x67*	*3 MSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Output 3 functioning***	*0x67*	*4 LSB*	*Byte*	*Get Attribute Single / Set Attribute Single*

Chapter	Name	EtherNet/IP Class	EtherNet/IP Attribute (dec)	Type	Service
Inputs/ outputs	***Output 4 functioning***	*0x67*	*4 MSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Set point 1 high value***	*0x67*	*5*	*Long*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Set point 1 low value***	*0x67*	*6*	*Long*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Set point 2 high value***	*0x67*	*7*	*Long*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Set point 2 low value***	*0x67*	*8*	*Long*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Set point 3 high value***	*0x67*	*9*	*Long*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Set point 3 low value***	*0x67*	*10*	*Long*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Set point 4 high value***	*0x67*	*11*	*Long*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Set point 4 low value***	*0x67*	*12*	*Long*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***1&2 Set points functioning***	*0x67*	*13 LSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***3&4 Set points functioning***	*0x67*	*13 MSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Logical input 3 functioning (optional)***	*0x67*	*14 LSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Logical input 4 functioning (optional)***	*0x67*	*14 MSB*	*Byte*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Analog output functioning (optional)***	*0x67*	*15*	*Uint*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***External value to control analog output***	*0x67*	*16*	*Uint*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Defective measurement debounced time***	*0x67*	*17*	*Uint*	*Get Attribute Single / Set Attribute Single*
Inputs/ outputs	***Defective measurement alarm activation time***	*0x67*	*18*	*Uint*	*Get Attribute Single / Set Attribute Single*
	0x68 (104d) / Instance 1				***Get Attribute All***
Commands	***command register***	*0x68*	*1*	*Uint*	*Get Attribute Single / Set Attribute Single*
Commands	***response register***	*0x68*	*2*	*Uint*	*Get Attribute Single*

Chapter	Name	EtherNet/ IP Class	EtherNet/ IP Attribute (dec)	Type	Service
Inputs/ outputs	***Input / output levels***	*0x68*	*3*	*Uint*	*Get Attribute Single*
Measures	***Sensor input control result***	*0x68*	*4*	*Int*	*Get Attribute Single*
	0x69 (105d) / Instance 1				***Get Attribute All / Set Attribut All***
Dosing	***Instant flow rate determination time***	*0x69*	*1*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Instant flow rate correction factor***	*0x69*	*2*	*Ulon g*	*Get Attribute Single / Set Attribute Single*
Dosing	***Average flow rate determination depth***	*0x69*	*3*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Learning cycle end of refill delay***	*0x69*	*4*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Learning cycle flow rates reference determination time***	*0x69*	*5*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***End of refill and cycle start stabilization time***	*0x69*	*6*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Cycle and alarm options***	*0x69*	*7*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Refilling low level***	*0x69*	*8*	*Ulon g*	*Get Attribute Single / Set Attribute Single*
Dosing	***Refilling high level***	*0x69*	*9*	*Ulon g*	*Get Attribute Single / Set Attribute Single*
Dosing	***Refilling inflight value***	*0x69*	*10*	*Long*	*Get Attribute Single / Set Attribute Single*
Dosing	***Fixed flow rate during refilling***	*0x69*	*11*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Reference flow rate control output start refilling***	*0x69*	*12*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Reference flow rate control output end refilling***	*0x69*	*13*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Empty hopper level***	*0x69*	*14*	*Ulon g*	*Get Attribute Single / Set Attribute Single*
Dosing	***Maximum time for refilling start***	*0x69*	*15*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Time interval for weight variation control on refilling***	*0x69*	*16*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Maximum refilling time***	*0x69*	*17*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Min permissible instant flow rate***	*0x69*	*18*	*Uint*	*Get Attribute Single / Set Attribute Single*

Chapter	Name	EtherNet/ IP Class	EtherNet/ IP Attribute (dec)	Type	Service
Dosing	***Max permissible instant flow rate***	*0x69*	*19*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Min permissible flow rate control output***	*0x69*	*20*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Max permissible flow rate control output***	*0x69*	*21*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Inhibit time of flow rates alarms at start***	*0x69*	*22*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Inhibit time of flow rates alarms in service***	*0x69*	*23*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Maximum batch time***	*0x69*	*24*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Weight to totalize (Great WU)***	*0x69*	*25*	*Ulon g*	*Get Attribute Single / Set Attribute Single*
Dosing	***Complementary weight to totalize***	*0x69*	*26*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Weigth to totalize inflight value***	*0x69*	*27*	*Long*	*Get Attribute Single / Set Attribute Single*
Dosing	***Nominal flow rate***	*0x69*	*28*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Weight quantity per pulse on logical output***	*0x69*	*29*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Dynamic zero acquisition time***	*0x69*	*30*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Segments number for the calibration curve of flow rate***	*0x69*	*31*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 1 (control output value)***	*0x69*	*32*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 2 (control output value)***	*0x69*	*33*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 3 (control output value)***	*0x69*	*34*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 4 (control output value)***	*0x69*	*35*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 5 (control output value)***	*0x69*	*36*	*Uint*	*Get Attribute Single / Set Attribute Single*

Chapter	Name	EtherNet/IP Class	EtherNet/IP Attribute (dec)	Type	Service
Dosing	***Calibration of flow rate point 6 (control output value)***	*0x69*	*37*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 7 (control output value)***	*0x69*	*38*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 8 (control output value)***	*0x69*	*39*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 9 (control output value)***	*0x69*	*40*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 10 (control output value)***	*0x69*	*41*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 1 (flow rate value)***	*0x69*	*42*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 2 (flow rate value)***	*0x69*	*43*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 3 (flow rate value)***	*0x69*	*44*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 4 (flow rate value)***	*0x69*	*45*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 5 (flow rate value)***	*0x69*	*46*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 6 (flow rate value)***	*0x69*	*47*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 7 (flow rate value)***	*0x69*	*48*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 8 (flow rate value)***	*0x69*	*49*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 9 (flow rate value)***	*0x69*	*50*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Calibration of flow rate point 10 (flow rate value)***	*0x69*	*51*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Kp***	*0x69*	*52*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Ti***	*0x69*	*53*	*Ulong*	*Get Attribute Single / Set Attribute Single*
Dosing	***Td***	*0x69*	*54*	*Ulong*	*Get Attribute Single / Set Attribute Single*

Chapter	Name	EtherNet/IP Class	EtherNet/IP Attribute (dec)	Type	Service
Dosing	***PID behavior***	*0x69*	*55*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***PID adjustment Flow Rate***	*0x69*	*56*	*float*	*Get Attribute Single / Set Attribute Single*
Dosing	***Full hopper level***	*0x69*	*57*	*Ulong*	*Get Attribute Single / Set Attribute Single*
Dosing	***Number of cycles for adjustment PID***	*0x69*	*58*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Cut-off frequency for totalization flow rate (x100)***	*0x69*	*59*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Time unit for totalization flow rate***	*0x69*	*60*	*Uint*	*Get Attribute Single / Set Attribute Single*
Dosing	***Weight unit for totalization flow rate***	*0x69*	*61*	*Ulong*	*Get Attribute Single / Set Attribute Single*

The register "Command register" uses the mechanism of ***eNod4*** functional commands defined in another chapter.
Note: "reset" and "Restore default settings" commands cannot be sent via cyclic and acyclic exchanges immediately after a restart of ***eNod4***. To be able to use these commands, it must first be processed another command ("cancel Tare" for example).
Note: The "Command register" data **must be** set to 0x0000 before each new command.

19 ETHERNET/IP ODVA COMMONLY DEFINED REGISTER MAP

Name	EtherNet/IP Class	EtherNet/IP Attribute	Type	Service
Identity Object Class 0x01 (01d) / Instance 0				**Get Attribute All**
Class Revision	0x01	1	Uint	Get Attribute Single
Max. Class Instance	0x01	2	Uint	Get Attribute Single
Class Max. Attributes	0x01	6	Uint	Get Attribute Single
Class Max. Instance Attributes	0x01	7	Uint	Get Attribute Single
Identity Object Class 0x01 (01d) / Instance 1				**Get Attribute All**
Vendor ID	0x01	1	Uint	Get Attribute Single / Reset
Device type	0x01	2	Uint	Get Attribute Single / Reset
Product Code	0x01	3	Uint	Get Attribute Single / Reset
Major Revision / Minor Revision	0x01	4	Uint	Get Attribute Single / Reset
status	0x01	5	Uint	Get Attribute Single / Reset
Serial Number	0x01	6	Ulong	Get Attribute Single / Reset
Length (bytes) / Product Name	0x01	7	string (14 bytes)	Get Attribute Single / Reset
State	0x01	8	byte	Get Attribute Single / Reset
Conf. Consist. Value	0x01	9	Uint	Get Attribute Single / Reset
Heart Interval	0x01	10	Uint	/
Assembly Object Class 0x04 (04d) / Instance 0				
Class Revision	0x04	1	Uint	Get Attribute Single
Max. Class Instance	0x04	2	Uint	Get Attribute Single
Connection Manager Object Class 0x06 (06d) / Instance 0				
Class Revision	0x06	1	Uint	Get Attribute Single
Max. Class Instance	0x06	2	Uint	Get Attribute Single
Connection Manager Object Class 0x06 (06d) / Instance 1				**Forward Close / Forward Open**
DLR (Device Level Ring) 0x47 (71d) / Instance 0				
Class Revision	0x47	1	Uint	Get Attribute Single
DLR (Device Level Ring) Object Class 0x47 (71d) / Instance 1				**Get Attribute All**
Network Topology	0x47	1	Byte	Get Attribute Single
Network Status	0x47	2	Byte	Get Attribute Single
Active Supervisor Address	0x47	10	Array of 10 bytes	Get Attribute Single
Capability Flags	0x47	12	Ulong	/

Name	EtherNet/IP Class	EtherNet/IP Attribute	Type	Service
QoS (Quality of Service) Object Class 0x48 (72d) / Instance 0				
Class Revision	0x48	1	Uint	Get Attribute Single
Max. Class Instance	0x48	2	Uint	Get Attribute Single
QoS (Quality of Service Object Class 0x48 (72d) / Instance 1				
802.1Q Tag Enable	0x48	1	Byte	Get Attribute Single
DSCP Urgent	0x48	4	Byte	Get Attribute Single
DSCP Scheduled	0x48	5	Byte	Get Attribute Single
DSCP High	0x48	6	Byte	Get Attribute Single
DSCP Low	0x48	7	Byte	Get Attribute Single
DSCP Explicit	0x48	8	Byte	Get Attribute Single
TCP/IP Interface Object Class 0xF5 (245d) / Instance 0				
Class Revision	0xF5	1	Uint	Get Attribute Single
Max. Class Instance	0xF5	2	Uint	Get Attribute Single
TCP/IP Interface Object Class 0xF5 (245d) / Instance 1				**Get Attribute All**
Status	0xF5	1	Ulong	Get Attribute Single
Configuration Capability	0xF5	2	Ulong	Get Attribute Single
Configuration Control	0xF5	3	Ulong	Get Attribute Single
Physical Link Object: Struct **Path size Uint** **Path Padded Epath**	0xF5	4	Array of n bytes	Get Attribute Single
Interface Configuration: Struct **IP address Uint** **Network mask Uint** **Gateway address Uint** **Name server Uint** **Name server Ulong** **Domain name String**	0xF5	5	Array of n bytes	Get Attribute Single
Host Name	0xF5	6	Array of n bytes	Get Attribute Single
Safety Network Number	0xF5	7	Array of n bytes	/
Time To Live value	0xF5	8	Array of n bytes	/
Multicast configuration	0xF5	9	Array of n bytes	/
Select ACD	0xF5	10	Array of n bytes	Get Attribute Single (01H)
Last Conflict Detected	0xF5	11	Array of n bytes	Get Attribute Single (01H)
Ethernet Link Object Class 0xF6 (246d) / Instance 0				
Class Revision	0xF6	1	Uint	Get Attribute Single
Max. Class Instance	0xF6	2	Uint	Get Attribute Single
Ethernet Link Object Class 0xF6 (246d) / Instance 1				

Name	EtherNet/IP Class	EtherNet/IP Attribute	Type	Service
Interface Speed	*0xF6*	*1*	*Ulong*	*Get Attribute Single*
Interface Flags	*0xF6*	*2*	*Ulong*	*Get Attribute Single*
Physical Address	*0xF6*	*3*	*Array of 6 bytes*	*Get Attribute Single*
Interface Control	*0xF6*	*6*	*Ulong*	*Get Attribute Single*
Length (byte) / Interface Label	*0xF6*	*10*	*string*	*Get Attribute Single*

Note:

- Get attribute All**: 0x01,** Get attribute Single**: 0x0E**
- Set attribute All**: 0x02,** Set Attribute Single**: 0x10**
- Reset**: 0x05**
- Forward open**: 0x54,** Forward close**: 0x4E**

20 CRC-16 CALCULATION ALGORITHM