Baumer

Passion for Sensors

# Manual

## Absolute encoder with SAE J1939

Baumer Electric AG
Hummelstrasse 17
CH-8501 Frauenfeld
Phone +41 52 728 11 22
Fax +41 52 728 11 44
sales.ch@baumer.com
www.baumer.com

08.18

# Contents

# Appendix

# 1 Version overview

This document is subject to changes. In order to have the most current version please download on www.baumer.com

| Document index | Date | Firmware version | CANopen Revision Number Obj. 1018 | Author | Changes |
|---|---|---|---|---|---|
| 0001 | 11.07.17 | From V01-03 | 0003.0000h | blk | Initial version replaces all draft documents |
| 0002 | 12.07.18 | From V01-05 | 0003.0000h | blk | - Electronic gear function added chapter 4.3.5 and 10.2<br>- Scaling function expanded up to 16Bit ST<br>- New „speed Filter time“ default value of 50ms (Object 4001h) range reduced from 4000 ms to 500 ms |

# 2 Safety and operating instructions

**Intended use**

- The encoder is a precision measuring device that is used to record positions and speeds. It provides measuring values as electronic output signals for the subsequently connected device. It must not be used for any other purpose. Unless this product is specially labeled, it may not be used for operation in potentially explosive environments.
- Make sure by appropriate safety measures, that in case of error or failure of the encoder, no danger to persons or damage to the system or operating facilities occurs.

**Personnel qualification**

- Installation and assembly of this product may be performed only by a person qualified in electronics and precision mechanics.

**Maintenance**

- The encoder is maintenance-free and must not be opened up nor mechanically or electronically modified. Opening up the encoder can lead to injury.

**Disposal**

- The encoder contains electronic components. At its disposal, local environmental guidelines must be followed.

**Mounting**

- Solid shaft: Do not connect encoder shaft and drive shaft rigidly. Connect drive and encoder shaft with a suitable coupling.
- Hollow shaft: Open clamping ring completely before mounting the encoder. Foreign objects must be kept at a sufficient distance from the stator coupling. The stator coupling is not allowed to have any contact to the encoder or the machine except at the mounting points.

**Electrical commissioning**

- Do not proceed any electrical modifications at the encoder.
- Do not proceed any wiring work while encoder is live.
- Do not remove or plug on connector whilst under power supply.
- Ensure that the entire system is installed in line with EMC/EMI requirements. Operating environment and wiring have an impact on the electromagnetic compatibility of the encoder. Install encoder and supply cables separately or far away from sources with high emitted interference (frequency converters, contactors, etc.).
- When working with consumers with high emitted interference provide separate encoder supply voltage.
- Completely shield encoder housing and connecting cables.
- Connect encoder to protective earth (PE) using shielded cables. The braided shield must be connected to the cable gland or connector. Ideally, aim at dual connection to protective earth (PE), i.e. housing by mechanical assembly and cable shield by the downstream devices.

**Supplementary information**

- The present manual is intended as a supplement to already existing documentation (e.g. catalogues, data sheets or mounting instructions).

# 3 Product Assignment

## 3.1 Absolute encoder

| Product |
| --- |
| Absolute encoder single- or multiturn encoder EAMxxx |

# 4 System Overview

## 4.1 General

The encoder is a rotary measuring system with a J1939 interface and supports scaling and presetting.
Galvanically isolated encoders are available on request.

## 4.2 Supported J1939 Services

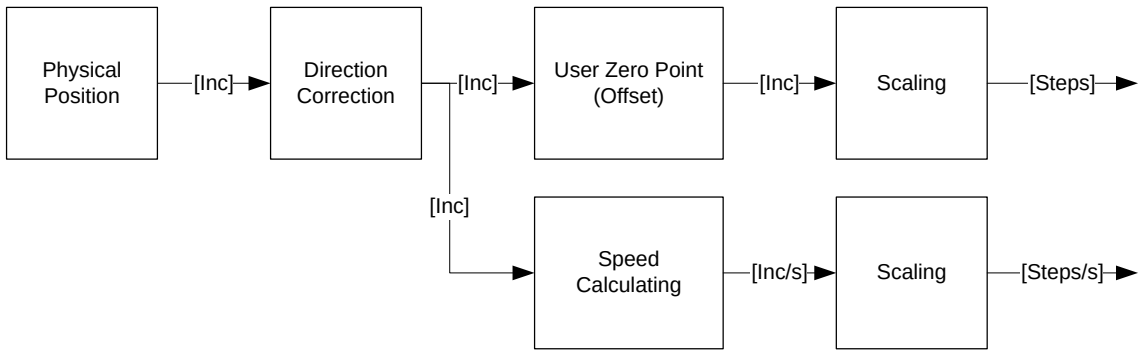
Following J1939 services are supported:

- Network Management (PGN60928)
  - Address claiming with Name-Field
- PGN65450 + Group Extension (PDU1)
  - Cyclic process data
  - Following Objects can be mapped into this PGN
    - Position (32-Bit)
    - Speed (16-Bit, [rpm])
    - Alarm (16-Bit)
    - Warnings (16-Bit)
    - Time Stamp (16-Bit)
- PGN61184 (PDU1)
  - Encoder parameters
    - Direction
    - Scaling (singleturn resolution/turn, Full resolution)
    - Preset
  - PDU2 parameters
    - Object mapping
    - Cyclic time
  - Baudrate
  - Group extension
  - Customer EEPROM (4x 32-Bit)
  - Save/Restore commands
- BAM messages
- Transport protocol

## 4.3 Function Principle

### 4.3.1 Overview

Figure 1: Function principle overview




### 4.3.2 Scaling

The scaling of speed and position objects can be adapted in the objects 6001h or 6002h.

*Relationship between object 6001h and 6002h:*

| | | | | |
|---|---|---|---|---|
| *Total measuring range* | = | *Measuring units per revolution* | *x* | *Number of distinguishable revolutions* |
| *(Value Object 6002h)* | = | *(Value Object 6001h)* | *x* | *(Value Object 6502h)* |

### 4.3.3 Position Range

The range of the position is depending on the position step setting (object 6001h-0 and Object 6002h-0). The total range can be read from object 6002h-0. The range is 0…(Value Object 6002h)-1.

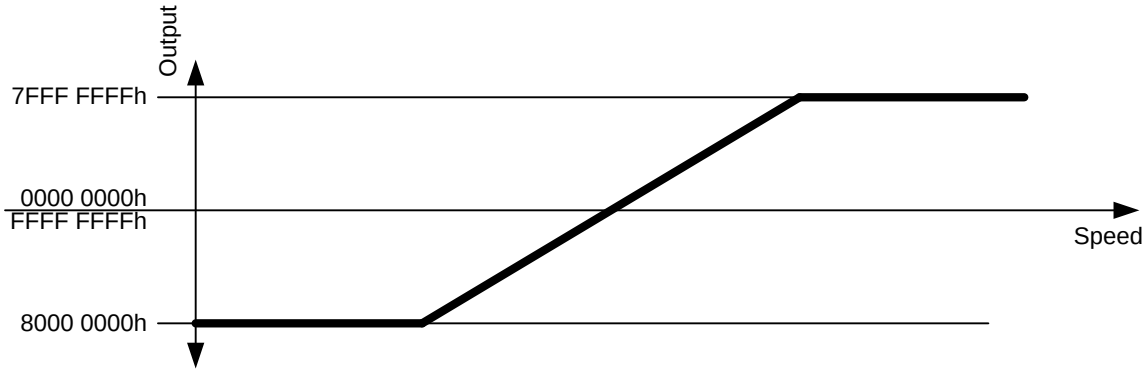
### 4.3.4 Speed range

There are two objects, which can be used for the speed information.

**0x6030-1**

This object provides a 32-Bit Speed information, which has the unit [Steps/sec].
The range for object 6030h-1 Speed encoder A is -8000'0000h…7FFF'FFFFh.
If the scaled speed value exceeds this range, the output is -8000'0000h or 7FFF'FFFFh (Saturated Logic).

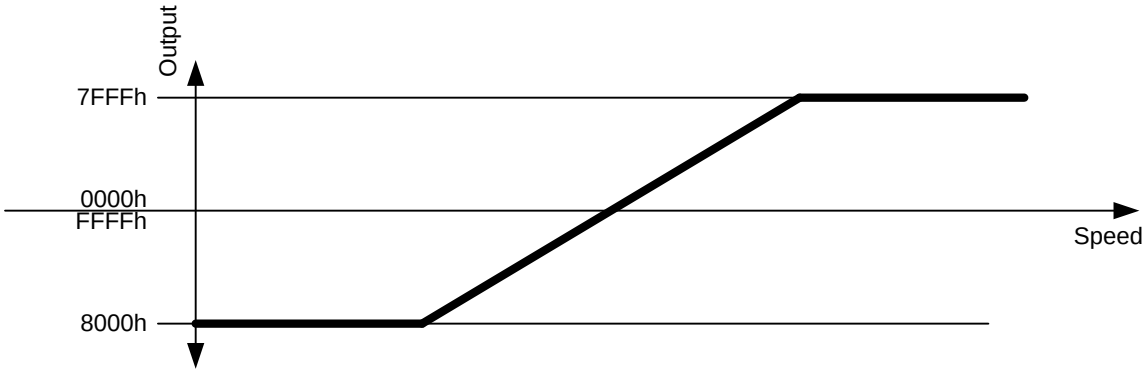
Figure 3: Speed range




**0x2120-1**

This object provides a 16-Bit Speed information, which has the unit [rpm].
The range for object 2120h-1 Speed encoder A is -8000h…7FFF'h.
If the scaled speed value exceeds this range, the output is -8000h or 7FFFh (Saturated Logic).

Figure 3: Speed range



### 4.3.5 Electronic gear function

The electronic gear function divides the position value by the gear factor. Therefore it transforms the position value into the view of the application:

$$application\ position = \frac{encoder\ position}{i}$$

The gear factor (i) is defined as followed:

$$i = \frac{GearValue1}{GearValue2}$$

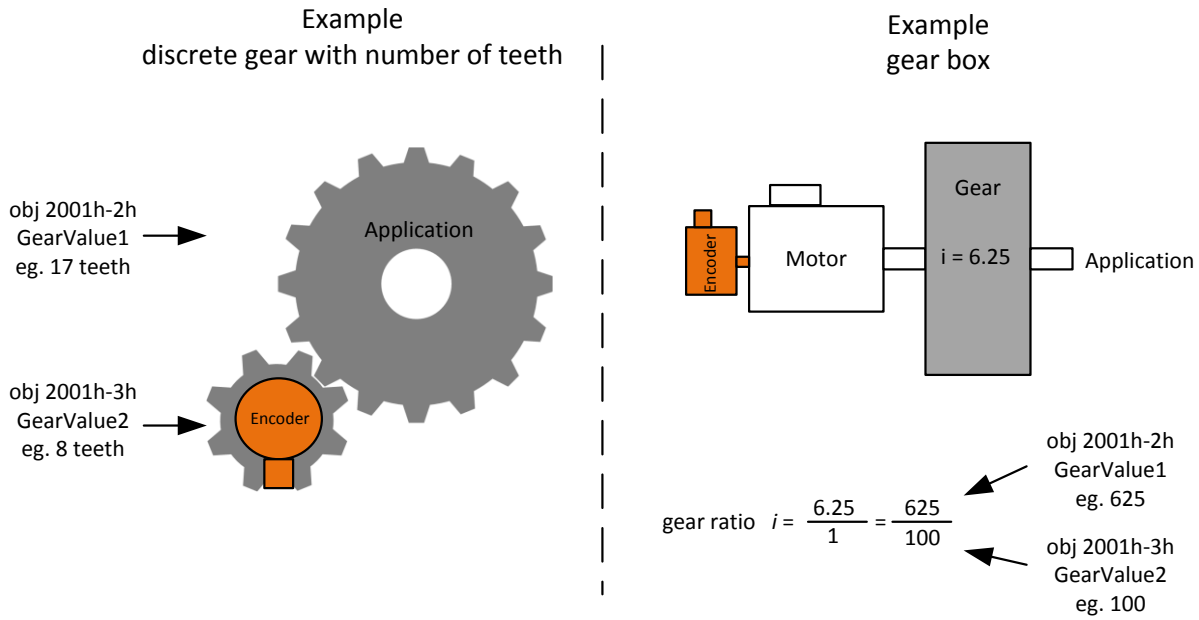
There are three objects that should be configured to use the electronic gear function.

**0x2001-1 Enable**
Set this object to the value “2” to enable the electronic gear function, while the value “1” disable it.

**0x2001-2 Gear Value 1**
This Object defines the numerator of the gear factor.
The range of this integer value is 1…32767.

**0x2001-3 Gear Value 2**
This Object defines the denominator of the gear factor.
The range of this integer value is 1…32767.

**Figure 1: Example configuration of gear values**




**Limitations**
A useful gear ratio is greater than 0.125, while a gear ratio smaller than 1 may result in higher signal noise.
The maximum encoder turns in unpowered operation must be smaller than $2^{29}$ (536’870’912) turns.
The electronic gear function is useful for multiturn encoders. In case of singleturn encoders, the position value gets lost after a power cycle.

## 4.4 Encoder as standard component with embedded software used in safety functions

If this standard encoder is used in safety functions, please request the according “Application Note MAGRES EAM” for further information.

# 5 CAN Frame

A standard CAN-Frame with a 29-Bit identifier is being used for the J1939 bus. The data in the PDU fields will be interpreted differently, depending on choosen PDU1 or PDU2 format, which is defined by the Identifier.

| | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| Bit | | | | | | | | | |

## 5.1 Identifier

The Identifier is defined as the CAN 29-Bit Identifier and can be configured by the user in two ways:

- Configuring of default ECU address (Object 0x2102)
- Configuring of Group Extension (Object 0x2103)

| | | Parameter Group Number (PGN) | | | | |
|---|---|---|---|---|---|---|
| Description | Priority | Extended Data page | Data Page | **PDU Format (PF)** | Destination Address Group extension | Source address (ECU address) |
| Bit | 28 … 26 | 25 | 24 | 23 … 16 | 15 … 8 | 7 … 0 |

Principally, there are two different message formats used in the J1939 protocol, which are defined by the data range of the PDU format field:

**PF Values 0 … 240** (called as PDU1 Format)

- Peer-to-Peer and broadcast communication
- Field Destination Address always contains the receivers address, or broadcast (25%)
- Used for Encoder parametrization

**PF Values 240 … 255** (called as PDU2 Format)

- Only broadcast communication
- Field Destination Address is used as Group Extension value
- Used for cyclic sending values of the Encoder

| | |
|---|---|
| Priority: | As lower the value, as higher is the priority on the bus.<br>This value is fixed to the value 6. |
| Extended Data page: | Only value 0 is supported |
| Data page: | Only value 0 is supported |
| PDU Format: | If Values < 0xEF -> PDU1 format is used<br>If Values > 0xFE -> PDU2 format is used |
| PDU1: Destination Address:<br>PDU2: Group Extension: | • PDU1: This is either the address of the Encoder, when requesting data or this will be the address of the ECU, which requested Data, when the Encoder responds to a message.<br>• PDU2: Group Extension, can used to create an offset to the cyclic message PGN (65450 + Group Extension) |
| Source address (ECU address) | Containing always the own address (default-value: 172). This address is claimed with the NM Service (see address claiming) |

## 5.2 Parameter Group Number (PGN)

The Parameter Group Number is an identifier for parameters.
The following PGN's are supported:

- PGN 60928 (Name-Field)
- PGN 61184 (Encoder Parameters)
- PGN 65450 (cyclic process data)

**Calculation of Parameter Group Number (PGN):**

PDU1 Format:
PGN = "PDU Format Field" * 256 + 0 -> Only every 256 PGN number exist.

PDU2 Format:
PGN = "PDU Format Field" *256 + "Group Extension value"

### 5.2.1 Identifier Examples

**Identifier for Cyclic PGN 65450 message:**

PGN 65450 => in Hex => 0xFFAA

➔ PDU-Field = 0xFF (Value > 240 -> PDU2 format with Group Extension is used)

➔ Group Extension = 0xAA

ECU-Address = 172 = 0xAC

| | | Parameter Group Number | | | | |
|---|---|---|---|---|---|---|
| Description | Priority | Extended Data page | Data Page | PDU Format (PF) | Destination Address **Group extension** | Source address (ECU address) |
| Value | 6 | 0 | 0 | 0xFF | 0xAA | 0xAC |
| Bit | 28 … 26 | 25 | 24 | 23 … 16 | 15 … 8 | 7 … 0 |
| Cob-ID [Hex] | 0x18FFAAAC | | | | | |

**Identifier for Acyclic PGN 61184 message, used for read command from another device to the encoder)**

PGN 61184 => in Hex => 0xEF00

➔ PDU-Field = 0xEF (Value < 240 -> PDU1 format with Destination Address is used)

➔ Dest. Address = 0xAC (default address of Encoder, could be higher, according to address claim)

| | | Parameter Group Number | | | | |
|---|---|---|---|---|---|---|
| Description | Priority | Extended Data page | Data Page | PDU Format (PF) | **Destination Address** Group extension | Source address (ECU address) |
| Value | 6 | 0 | 0 | 0xEF | 0xAC | 0x00 |
| Bit | 28 … 26 | 25 | 24 | 23 … 16 | 15 … 8 | 7 … 0 |
| Cob-ID [Hex] | 0x18EFAC00 | | | | | |

# 6 Network Management Service

## 6.1 NAME Field (PGN 60159)

The name field will be sent by the device on every start up as a broadcast message. The name field is used to identify the device in the network, as well as for the address claiming. The name field can't be configured by the user.

| | | | Bits of Protocol Data Unit (PDU) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | 21 Bit | 11 Bit | 3 Bit | 5 Bit | 8 Bit | 1 Bit | 7 Bit | 4 Bit | 3 Bit | 1 Bit |
| | PGN60159 | 8 | Serial-Nr | Manufact. code | ECU Instance | Function Instance | Function | Res. | Vehicle System | Vehicle Instance | Industry Group | Arbitrary Address capable |
| EAMxxx | 18EEFFACh | 8 | A7E77h | 157h | 0h | 0h | 8Eh | 0h | 0h | 0h | 5h | 1h |

## 6.2 Address Claiming

Every device on the J1939 network needs to observe name field messages on the bus in order to detect, if another device tries to use the same address as the own device.
When it is detected, that another device using the same address, the own name field is also broadcasted on the bus. The device with the lower priority (higher Identifier) needs to increment its own address by one and broadcast the name-field on the bus again. Save and restore commands

# 7 PGN65450 - cyclic message (PDU2 Format)

## 7.1 General

The device supports a cyclic PDU2 message with PGN65450+GroupExtenstion, which can be used for cyclic transmitting of the encoder values.
The data/objects which are sent with the PDU2 can be ordered customer specific. The cycle time, can be configured.

## 7.2 Frame Format

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| Value (example) | 18FFAAACh | 8 | 18h | 91h | B7h | 73h | 21h | F0h | 01h | 40h |

**Interpretation of the values according the example values:**

**Identifier:**
Priority: 6
PGN: FFAAh = 65450d
Group Extension: ACh = 172d

Data-Bytes are according to the defined mapping.
Default-mapping is:

**PDU:**

| **Bytes** | **Description** | **Hex-Value from example** | **Interpretation** |
|---|---|---|---|
| **0 … 3** | Position Value (14-Bit ST, 18Bit MT) | 73B79118h | Multiturn-Value = 118478d<br>Singleturn-Value = 4376d |
| **4 … 5** | Speed Value | F021h | - 4063d rpm |
| **6** | Warning | 01h | Frequency exceeded |
| **7** | Error | 40h | Position Error |

## 7.3 Mapping

The encoder supports a customer specific mapping for PGN65450, which can be specified, when ordering an encoder.

### 7.3.1 Mappable objects

The following objects are mappable:

| Mapping content | Mapping entry | Description |
|---|---|---|
| Position encoder | 0x60040020 | Object 6004h Subindex 00h, data length 32 Bit |
| Speed encoder [steps/sec] | 0x60300120 | Object 6030h Subindex 01h, data length 32 Bit |
| Alarms | 0x65030010 | Object 6503h Subindex 00h, data length 16 Bit |
| Warnings | 0x65050010 | Object 6505h Subindex 00h, data length 16 Bit |
| Diagnostic | 0x21170010 | Object 2117h Subindex 00h, data length 16 Bit |
| Speed [rpm] | 0x21180010 | Object 2117h Subindex 00h, data length 16 Bit |
| Time Stamp | 0x21200010 | Object 2120h Subindex 00h, data length 16 Bit |

### 7.3.2 Default mapping of absolute encoder

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|
| Position[0] | Position[1] | Position[2] | Position[3] | Speed[0] | Speed[1] | Warnings | Errors |

**Byte 0 – 3 Encoder absolute position:**

Datatype: 4 Byte, U32,

Resolution: According to scaling factors (see encoder parametrization)

**Byte 4 – 6 Encoder speed value:**

Datatype: 2 Byte, S16,

Resolution: [rpm]

**Byte 7 – 8 (Encoder diagnostic):**
See Object description in chapter 0x2117 Encoder Diagnostics

## 7.4 Timing

The minimal cycle time for PGN65450 is 1 ms, although the J1939 suggest to use minimal cycle times of 50 ms.

## 7.5 Exceptions of accurate calculation of process data

The following operations could interrupt the accurate calculation of process data such as position, speed, warnings and alarms:

- Changing the scaling parameters

# 8 PGN61184 - Encoder Parameters and values (PDU1 Format)

PDU1 Format is being used for reading and writing of the encoder parameters.

The following parameters can be accessed

- Encoder Values
  - Position (Index 6004h)
  - Speed (Index 6020h / 2020h)
  - Diagnostic (Index 2017h)
- Encoder parameters
  - Direction (Index 6000h)
  - Scaling (singleturn resolution/turn)
  - Preset
- PGN65450 cyclic message parameters
  - Transmission rate
- Baudrate
- ECU default address
- Group extension
- Customer EEPROM (4x 32-Bit)
- Save/Restore commands

## 8.1 Request frame format for PDU1

| | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Description | DLC | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| Bit | 8 | CMD | Index | | Sub-Index | Data bytes | | | |

The PF-Value (PDU Format value) is shall always be 0xEF.
The LSB of Index and Data bytes are always transferred first (see examples).

## 8.2 CMD Codes

| CMD-Codes | Description |
|---|---|
| 22h | Writing request, unspecified length |
| 23h | Writing request, 4 bytes |
| 27h | Writing request, 3 bytes |
| 2Bh | Writing request, 2 bytes |
| 2Fh | Writing request, 1 bytes |
| | |
| 40h | Read command |

## 8.3 Response Frame format for PDU1

| | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Description | DLC | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| Bit | 8 | Resp. Code | Index | | Sub-Index | Data of Object / Error-Code | | | |

Be aware, that the LSB is always transmitted before MSB. For example Index = 2104h, Byte 1 = 04h, Byte 2 = 21h.

## 8.4 Response Codes

| Response-Code | Description |
|---|---|
| 80h | Error occurred -> Check Error-Code |
| | |
| 60h | Write command successfully |
| | |
| 4Fh | Reading of 1 byte of data successfully |
| 4Bh | Reading of 2 byte of data successfully |
| 47h | Reading of 3 byte of data successfully |
| 43h | Reading of 4 byte of data successfully |

## 8.5 Error-Codes

| Error-Code | Description |
|---|---|
| 0504 0001h | Sent CMD is unknown |
| 0601 0000h | Access to object unsupported |
| 0601 0001h | Read-Access not supported to object |
| 0601 0002h | Write-Access not supported to object |
| 0602 0000h | Object doesn‘t exist |
| 0609 0011h | Sub-Index doesn’t exist |
| 0609 0030h | Value out of Range |
| 0609 0031h | Value too high |
| 0609 0032h | Value too low |

# 9 Object descriptions and examples

The frequently used objects are described in this chapter. More objects can be found in the Object directory.

## 9.1 0x1010 Saving of parameters

With the object 0x1010, the parameters can be saved to the non-volatile memory and applied at startup. The response is sent by the encoder after the parameters are saved to the non-volatile memory.

The Sub-Indexes are defined as follow:

| Index - SubIndex | Description | Parameters which are saved |
|---|---|---|
| 0x1010-1 | Saving of all parameter | Please see the object dictionary for all parameters, which are saved. |
| 0x1010-2 | Not used for J1939 | |
| 0x1010-3 | Saving of application parameters | 10.3 Standardized Device Profile Area |
| 0x1010-4 | Saving of manufacturer specific parameters | Manufacturer Specific Profile Area |

A magic number-sequence (to avoid accidental saving) is being used to trigger the save-command. The letters “save” needs to be used as Byte 4 … 7 (73h, 61h, 76h, 65h)

**Example of saving all parameters:**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Send saving of **all** parameters | 18EFAC00h | 8 | 22 | 10 | 10 | **01** | 73 | 61 | 76 | 65 |
| Successful response by encoder (60h) | 18EF00ACh | 8 | 60 | 10 | 10 | 01 | 00 | 00 | 00 | 00 |

## 9.2 0x1011 Restoring of default values

With the object 0x1011, the parameters can be restored to the factory preset values (factory values ca be defined when ordering). The values are restored after a reset of the encoder.
The response is sent by the encoder after the command for restoring the parameters is save to the non-volatile memory.

Sub-Indexes are defined as follow:

| Index-SubIndex | Description | Parameters which are restored |
|---|---|---|
| 0x1011-1 | Restore of all parameter | Please see the object dictionary for all parameters, which are restored. |
| 0x1011-2 | Not used for J1939 | |
| 0x1011-3 | Restore of application parameters | 10.3 Standardized Device Profile Area |
| 0x1011-4 | Restore of manufacturer specific parameters | Manufacturer Specific Profile Area |

A magic number-sequence (to avoid accidental saving) is being used to trigger the save-command. The letters “load” needs to be used as Byte 4 … 7 (6C, 6Fh, 61h, 64h)

**Example of restoring the application parameters:**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Send restoring of **all** parameters | 18EFAC00h | 8 | 22 | 11 | 10 | **01** | 6C | 6F | 61 | 64 |
| Successful response by encoder (60h) | 18EF00ACh | 8 | 60 | 11 | 10 | 01 | 00 | 00 | 00 | 00 |

## 9.3 0x1018-2 Reading of the product code

| Product | Product-Code (Byte 6 and Byte 7) | Device Name |
|---|---|---|
| Absolute encoder multiturn | 0x0070 | EAMxxx MT |
| Absolute encoder singleturn | 0x0071 | EAMxxx ST |

Byte 5: Major Product Code
Byte 4: Minor Product Code

**Example of reading the product code**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Reading of the product code | 18EFAC00h | 8 | 40 | 18 | 10 | 02 | 00 | 00 | 00 | 00 |
| Response by encoder:<br>0071h = Singleturn Encoder | 18EF00ACh | 8 | 60 | 02 | 21 | 00 | 71 | 00 | 00 | 00 |

## 9.4 0x6507 Reading of the software version

Byte 7: Major FW-Version
Byte 6: Minor FW-Version

**Example of reading the software version**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Reading of the software version | 18EFAC00h | 8 | 40 | 07 | 65 | 00 | 00 | 00 | 00 | 00 |
| Response by encoder: 0102h => V01.02 | 18EF00ACh | 8 | 43 | 07 | 65 | 00 | xx | xx | 02 | 01 |

## 9.5 0x2100 Baudrate

| Value | Baudrate |
|---|---|
| 0 | 10 kBit/s |
| 1 | 20 kBit/s |
| 2 | 50 kBit/s |
| 3 | 100 kBit/s |
| 4 | 125 kBit/s |
| 5 | 250 kBit/s |
| 6 | 500 kBit/s |
| 7 | 800 kBit/s |
| 8 | 1000 kBit/s |
| 9 | Autobaud |

The baudrate is activated after a reset or power-on (if parameter is saved to non-volatile memory)

**Example of changing baudrate to 500 kBit/s**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Changing baudrate to 500 kBit/s (value 6) | 18EFAC00h | 8 | 22 | 00 | 21 | 00 | 06 | 00 | 00 | 00 |
| Successful encoder response (60h) | 18EFAAACh | 8 | 60 | 00 | 21 | 00 | 00 | 00 | 00 | 00 |
| Send saving of all parameters | 18EFAC00 | 8 | 22 | 10 | 10 | 01 | 73 | 61 | 76 | 65 |
| Successful encoder response (60h) | 18EFAAACh | 8 | 60 | 10 | 10 | 01 | 00 | 00 | 00 | 00 |
| Reboot of Encoder and changing of CAN-Adapter baudrate | | | | | | | | | | |

## 9.6 0x6001 Measuring units per revolution

The measuring units per revolution (Singleturn resolution) of the encoder can be set WITH this object (32-Bit value)

**Example of setting the ST Resolution to 3600**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Write Measuring Units to Encoder | 18EFAC00h | 8 | 22 | 01 | 60 | 00 | **00** | **00** | **0E** | **10** |
| Successful response by encoder (60h) | 18EF00ACh | 8 | 60 | 03 | 60 | 00 | 00 | 00 | 00 | 00 |
| *Optional send saving of all parameters* | *18EFAC00h* | *8* | *22* | *10* | *10* | *01* | *73* | *61* | *76* | *65* |
| *Optional successful response by encoder (60h)* | *18EF00ACh* | *8* | *60* | *10* | *10* | *01* | *00* | *00* | *00* | *00* |

## 9.7 0x6003 Preset value encoder

The position value of the encoder can be preset by writing the desired position to this object. The preset position value is a 32-Bit value.

**Example of setting the position to 0x52660204**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Write preset Position to Encoder | 18EFAC00h | 8 | 22 | 03 | 60 | 00 | **04** | **02** | **66** | **52** |
| Successful response by encoder (60h) | 18EF00ACh | 8 | 60 | 03 | 60 | 00 | 00 | 00 | 00 | 00 |
| *Optional send saving of all parameters* | *18EFAC00h* | *8* | *22* | *10* | *10* | *01* | *73* | *61* | *76* | *65* |
| *Optional successful response by encoder (60h)* | *18EF00ACh* | *8* | *60* | *10* | *10* | *01* | *00* | *00* | *00* | *00* |
| Cyclic PGN65450 with new Position Value | 18FFDDACh | 8 | **04** | **02** | **66** | **52** | 00 | 00 | 00 | 00 |

## 9.8 0x2103 Group Extension

The group extension is a 8-Bit value, which is transmitted in Byte 4.

**Example of changing the Group Extension to 55h**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Cyclic PGN65450 with Group Ext. 00h | 18**FFAA**ACh | 8 | 18 | 91 | B7 | 73 | 00 | 00 | 00 | 00 |
| Changing Group Extension to 33h | 18EFAC00h | 8 | 22 | 03 | 21 | 00 | **33** | 00 | 00 | 00 |
| Successful response by encoder (60h) | 18EF00ACh | 8 | 60 | 03 | 21 | 00 | 00 | 00 | 00 | 00 |
| *Optional Send saving of all parameters* | *18EFAC00h* | *8* | *22* | *10* | *10* | *01* | *73* | *61* | *76* | *65* |
| *Optional Successful response by encoder (60h)* | *18EF00ACh* | *8* | *60* | *10* | *10* | *01* | *00* | *00* | *00* | *00* |
| Cyclic PGN65450 with Group Ext. **33h** | 18**FFDD**ACh | 8 | 18 | 91 | B7 | 73 | 00 | 00 | 00 | 00 |

**Example of Reading Group Extension**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Read Group Extension of encoder | 18EFAC00h | 8 | 40 | 03 | 21 | 00 | 00 | 00 | 00 | 00 |
| Response by encoder with **00h** | 18EF00ACh | 8 | 4F | 03 | 21 | 00 | **00** | 00 | 00 | 00 |

## 9.9 0x2104 Transmission rate for PGN65450

The transmission time is a 16-Bit value, which is in Byte 4 and Byte 5 and has the unit [ms].

**Example of changing the transmission Rate to 500 ms (1F4h)**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Changing of transmission time to 500ms | 18EFAC00h | 8 | 22 | 04 | 21 | 00 | **F4** | **01** | 00 | 00 |
| Successful response by encoder (60h) | 18EF00ACh | 8 | 60 | 04 | 21 | 00 | 00 | 00 | 00 | 00 |
| *Optional Send saving of all parameters* | *18EFAC00h* | *8* | *22* | *10* | *10* | *01* | *73* | *61* | *76* | *65* |
| *Optional Successful response by encoder (60h)* | *18EF00ACh* | *8* | *60* | *10* | *10* | *01* | *00* | *00* | *00* | *00* |
| Cyclic PGN65450 is sent every 500ms | | | | | | | | | | |

**Example of Reading tPGN65450 transmission time**

| | | | Protocol Data Unit (PDU) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Description | Identifier | DLC | Byte 0 … 7 [hex] | | | | | | | |
| Read transmission time of encoder | 18EFAC00h | 8 | 40 | 04 | 21 | 00 | 00 | 00 | 00 | 00 |
| Response by encoder with **1F4h** | 18EF00ACh | 8 | 4B | 03 | 21 | 00 | **F4** | **01** | 00 | 00 |

## 9.10 0x2117 Encoder Diagnostics

**Byte 0: Warnings**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| MTSuper | STSys | MTSys | MountErr | BattLow | OpTimeLim | WDT | FreqExc |

| Byte 0 | Correspondent bit & register | Meaning |
|---|---|---|
| Bit 7 | Bit 15<br>0x6505 Warning | **MTSuper**: Multiturn supervision<br>Set, when Multi- and Singleturn system are out of sync. |
| Bit 6 | Bit 14<br>0x6505 Warning | **STSys**: Singleturn system<br>Set, when Singleturn system error detected |
| Bit 5 | Bit 13<br>0x6505 Warning | **MTSys**: Multiturn system (corresponds to Bit 13 of Warning Register 0x6505)<br>Set, when Multiturn system error detected |
| Bit 4 | Bit 12<br>0x6505 Warning | **MountErr:** Mounting error<br>Set, when Mounting error has been detected (i.e. magnetic field disturbed) |
| Bit 3 | Bit 4<br>0x6505 Warning | **BattLow**: Batterie low charge<br>Set, when the battery of the Multiturn system gets low. Encoder should be replaced to guarantee absolute position. |
| Bit 2 | Bit 3<br>0x6505 Warning | **OpTimeLim**: Operating time limit<br>Set, when a possible problem with the non-volatile memory is detected. Encoder should be replaced to guarantee absolute position. |
| Bit 1 | Bit 2<br>0x6505 Warning | **WDT**: Watch dog timer triggered<br>Set, when a SW-Reset due to a watchdog timer occurred. Encoder should be replaced to guarantee absolute position. |
| Bit 0 | Bit 0<br>0x6505 Warning | **FreqExc:** Frequency exceeded<br>Set, when the maximum guaranteed rotation speed is exceed. |

**Byte 1: Errors**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| BattEmpt | PosErr | TempErr | CommErr | Reserved | Reserved | Reserved | Reserved |

| Byte 1 | Correspondent bit & register | Meaning |
|---|---|---|
| Bit 7 | Bit 15<br>0x6503 Alarm | **BattEmpt**: Battery empty<br>Set, when the battery of the Multiturn system is empty. Absolute position is not guaranteed. |
| Bit 6 | Bit 12<br>0x6503 Alarm | **PosErr**: Position Error<br>Set, when Position error occurred (empty battery of magnetic field distortion) |
| Bit 5 | Bit 3<br>0x1001 Error | **TempErr**: Temperature Error<br>Set, when the maximum operating temperature is exceeded |
| Bit 4 | Bit 4<br>0x1001 Error | **CommErr**: Communication Error<br>Set, when a communication error has been detected (Bus off, Bus Warning, RX overflow) |
| Bit 0-3 | | **Reserved fields** |

# 10 Object directory

The following tables provide a summary of all objects supported by the encoder.

**Object** Object number in Hex
**Name** Object name
**Format** U/I = Unsigned/Integer, No. = no of bits, ARR = Array, REC = Record, STR = String
**Access** ro = read only, wo = write only, rw = read write, m = mappable
**Default** Default value on first init
**Save** X = can be stored in the EEPROM
**Description** Additional information

## 10.1 Communication Profile Area

| Object | Sub-Index | Name | Format | Access | Default | Save | Description |
|---|---|---|---|---|---|---|---|
| 1000h | | Device Type | U32 | ro | | | Singleturn Encoders: 0001'0196h<br>Multiturn Encoders: 0002'0196h |
| 1001h | | Error Register | U8 | ro | 0h | | Bit0 = Generic error<br>Bit3 = Temperature error<br>Bit4 = Communication error |
| 1008h | | DeviceName | STR | ro | | | Devicename =<br>"EAMxxx MT"<br>"EAMxxx ST" |
| 1009h | | Hardware Version | STR | ro | | | Hardware version in ASCII |
| 100Ah | | Software Version | STR | ro | | | Software version in ASCII |
| 1010h | | Store parameters | ARR | | | | |
| | 00h | Largest Subindex | U8 | ro | 4h | | No. of save possibilities 4 |
| | 01h | Save all parameters | U32 | rw | | | ="evas " (0x65766173) to save |
| | 03h | Application parameters (Scaling, Direction) | U32 | rw | | | ="evas " (0x65766173) to save |
| | 04h | Manuf. Specific parameters (Default ECU-Address, Baudrate, Group Extension) | U32 | rw | | | ="evas " (0x65766173) to save |
| 1011h | | Restore default parameters | ARR | | | | |
| | 00h | Largest Subindex | U8 | ro | 4h | | No. of reset possibilities = 4 |
| | 01h | All parameters | U32 | rw | | | ="daol" (0x64616F6C) to load |
| | 03h | Application parameters | U32 | rw | | | ="daol" (0x64616F6C) to load |
| | 04h | Manufacturer specific parameters | U32 | rw | | | ="daol" (0x64616F6C) to load |
| 1018h | | Identity object | REC | ro | | | |
| | 00h | Largest subindex | U8 | ro | 4h | | |
| | 01h | Vendor ID | U32 | ro | 5Fh | | Vendor ID |
| | 02h | Product code | U32 | ro | | | Product code:<br>70h = EAMxxx Multiturn Encoder<br>71h = EAMxxx Singleturn Encoder |
| | 03h | Revision number | U32 | ro | | | Product revision No. |
| | 04h | Serial number | U32 | ro | | | Serial No. |

## 10.2 Manufacturer Specific Profile Area

| Object | Sub-Index | Name | Format | Access | Default | Save | Description |
|---|---|---|---|---|---|---|---|
| 2100h | | Baudrate | U8 | rw | 2h | X | 0=10 kBit/s<br>1=20 kBit/s<br>2=50 kBit/s<br>3=100 kBit/s<br>4=125 kBit/s<br>5=250 kBit/s<br>6=500 kBit/s<br>7=800 kBit/s<br>8=1000 kBit/s<br>9=Autobaud<br>The baudrate is activated after a reset or power-on (if parameter is saved to non volatile memory) |
| 2001h | | Electronic Gear | ARR | | | | Configuration of electronic gear function |
| | 00h | Largest Subindex | U8 | ro | 3 | | |
| | 01h | Enable | U16 | rw | 1 | X | 1 = electronic gear function disabled<br>2 = electronic gear function enabled |
| | 02h | Gear Value1 | U16 | rw | 1 | X | Numerator of the gear factor, Range 1…32767 |
| | 03h | Gear Value2 | U16 | rw | 1 | X | Denominator of the gear factor, Range 1…32767 |
| 2102h | | ECU default address | U8 | rw | ACh | X | ECU Address. Address-Range: 0 … 253 |
| 2103h | | Group Extension PDU2 | U8 | rw | 0h | X | Group Extension, used for PDU2 (proprietary B) |
| 2104h | | PGN65450 Transmission Rate | U16 | rw | 32h | X | Transmission Rate for PGN65450 [ms] |
| 2110h | | Feature control | U16 | rw | 0008h | X | Bit 3: CAN Bus Off behavior<br>1 = Automatic CAN restarting<br>0 = Encoder behaves according obj. 1029h |
| 2114h | | Manufacturer Reserved | U32 | rw | 0 | | |
| 2117h | | Encoder diagnostic | U16 | ro, m | | | Encoder diagnostic bits |
| 2118h | | Speed [rpm] | S16 | ro, m | | | Speed value of Encoder in [rpm] |
| 2120h | | Time stamp [us] | U16 | ro, m | | | Time stamp in [us] of current position acquisition |
| 2300h | | Customer EEPROM | ARR | | | | Customer EEPROM to save any data |
| | 00h | Largest Subindex | U8 | ro | 4 | | |
| | 01h | CustomerEEPROM[0] | U32 | rw | 0 | X | |
| | 02h | CustomerEEPROM[1] | U32 | rw | 0 | X | |
| | 03h | CustomerEEPROM[2] | U32 | rw | 0 | X | |
| | 04h | CustomerEEPROM[3] | U32 | rw | 0 | X | |
| 4001h | | Speed sampling interval in ms | U16 | rw | 50 | X | The speed sampling interval sets up the sampling interval of the speed calculation<br>Range 1…500ms |

## 10.3 Standardized Device Profile Area

| Object | Sub-Index | Name | Format | Access | Default | Save | Beschreibung |
|---|---|---|---|---|---|---|---|
| 6000h | | Operating parameter | U16 | rw | 4h | X | Bit0 = 0 Position CW<br>1 Position CCW<br>Bit 2 = 0 Scaling function disabled<br>1 Scaling function enabled |
| 6001h | | Measuring units per revolution [Step/rev] | U32 | rw | 4000h | X | Measuring units per revolution. Allowed range 4…65'536<br>Writing this object will adjust object 6502.<br>6502h = 6002h / 6001h |
| 6002h | | Total measuring range [Step] | U32 | rw | | | Total measuring range. This value results from multiplication of objects 6001h and 6502h.<br>Writing to this object will adjust object 6001h.<br>6002h/6502H = 6001h<br><br>(Value 0h means 1'0000'0000h Steps) |
| 6003h | | Preset value encoder [Step] | U32 | rw | 0h | | Preset in steps for encoder A → Offset<br>(Internally linked to object 6020h-1) |
| 6004h | | Position encoder [Step] | U32 | ro m | | | Position in steps for encoder A (Internally linked to object 6020h-1) |
| 6010h | | Preset Value | Array | | | | |
| | 00h | Largest Subindex | U08 | ro | 1 | | |
| | 01h | Preset for encoder A [Step] | U32 | rw | 0 | | Preset in steps for encoder A |
| 6030h | | Speed Values | Array | | | | |
| | 00h | Largest Subindex | U08 | ro | 1 | | |
| | 01h | Speed encoder A [Step/s] | U32 | ro, m | | | Speed in steps/second for encoder A |
| 6500h | | Operating Status | U16 | ro | 4h | | Bit0 = 0 Position CW<br>1 Position CCW<br>Bit2 = 0 Scaling function disabled<br>1 Scaling function enabled |
| 6501h | | Used single turn resolution [step/rev] | U32 | ro | 4000h | | |
| 6502h | | Number of distinguishable revolutions | U16 | ro | FFFFh | | |
| 6503h | | Alarms | U16 | ro, m | 0h | | The following alarms are evaluated:<br>Bit0 = Position error<br>Bit12 = Data valid encoder<br>Bit15 = Battery Empty |
| 6504h | | Supported alarms | U16 | ro | Standard: 9001h | | The following alarms are supported:<br>Bit0 = Position error<br>Bit12 = Data valid encoder<br>Bit15 = Battery Empty |
| 6505h | | Warnings | U16 | ro, m | 0h | | The following warnings are evaluated:<br>Bit0 = Frequency exceeded (SpeedMonitoring)<br>Bit2 = CPU watchdog status reset generated<br>Bit3 = Operating time limit<br>Bit4 = Battery charge too low<br>Bit12= Mounting error<br>Bit13 = Multiturn System<br>Bit14 = Singleturn System<br>Bit15 = Multiturn Supervision |
| 6506h | | Supported warnings | U16 | ro | F00Dh | | The following warnings are supported:<br>Bit0 = Frequency exceeded (SpeedMonitoring)<br>Bit2 = CPU watchdog status reset generated<br>Bit3 = Operating time limit<br>Bit4 = Battery charge too low<br>Bit12= Mounting error<br>Bit13 = Multiturn System<br>Bit14 = Singleturn System<br>Bit15 = Multiturn Supervision |
| 6507h | | Profile & software version | U32 | ro | | | Byte 0..1: Profile-Version =4.14 = 040Eh<br>Byte 2: Software minor version<br>Byte 3: Software major version |
| 6508h | | Operating time | U32 | ro | 0h | | Always FFFF'FFFFh |
| 6509h | | Offset encoder [step] | I32 | ro | 0h | | Offset encoder (Internally linked to object 650Ch-1) |
| 650Ah | | Module identification | Array | | | | |
| | 00h | Largest Subindex | U08 | ro | 1 | | |
| | 01h | Manufacturer offset | I32 | ro | 0 | | |
| 650Bh | | Serial number | U32 | ro | | | Internally linked to object 1018h-4h |
| 650Ch | | Offset values | Array | | | | |
| | 00h | Largest subindex | U08 | ro | 1 | | |
| | 01h | Offset encoder [step] | I32 | ro | 0h | X | Offset encoder A |

# Appendix

## A. Pin Assignments

Default Pin Assignments for galvanically isolated encoders are listed below (non-isolated on request).

Description

| signal | description |
|---|---|
| +Vs | Supply voltage |
| 0 V | Supply voltage |
| CAN_GND | CAN bus ground |
| CAN_H | CAN bus Signal (dominant HIGH) |
| CAN_L | CAN bus signal (dominant LOW) |

### A.1 Assignment cable (connection – L) and connector 1 x M12 (connection – N)

| Cable color | M12 | Cable |
|---|---|---|
| grey | 1 | 0 V |
| brown | 2 | +Vs |
| white | 3 | 0 V |
| green | 4 | CAN_H |
| yellow | 5 | CAN_L |
| pink | - | - |
| blue | - | - |
| red | - | - |