OCTOGON

# MANUAL

# EMFA

## CAN INLINE ADAPTER FOR LOAD CELLS

Operating manual with description of the CANopen interface


| Document | Version | Date | Applies to |
|---|---|---|---|
| Operating manual CANopen | V00x | 09/2026 | EMFA, CANopen protocol, firmware V1.1 |

The EMFA sits between an analogue strain gauge load cell and the CAN bus: it excites the load cell, converts its mV/V signal at 24 bit and outputs it over CANopen. Sections 3 and 4 describe the device and its connection, sections 5 to 7 the CANopen interface, the object dictionary and all commands with example frames.

octogon GmbH · Roseggerstrasse 16 · 8700 Leoben · Austria · office@octogon.org

# Contents

# 1 General

## 1.1 About this manual

– This manual describes the connection of the EMFA and its operation over the CANopen protocol: network management, process and service data objects, the object dictionary and every command with an example frame.
– It describes the features of firmware V1.1. The user parameters (section 7.7.5) and the option to disable RPDO1 are available from this version on.
– The EMFA can be switched from CANopen to the J1939 protocol (section 7.3.3). J1939 is not described in this manual but in a separate document.
– The technical data are given in the EMFA datasheet; section 3 summarises them.
– This manual is part of the product. Keep it accessible for all users and read it before connection and start-up.

## 1.2 Liability

The information in this manual has been compiled with care; it is subject to change without notice. No liability is accepted for damage caused by accident, misuse, use contrary to the intended purpose, unauthorised modifications or repairs, or failure to follow this manual.

## 1.3 Notation and data types

– Hexadecimal values carry the suffix “h” (e.g. **`9Ch`**); numbers without a suffix are decimal.
– All data are transmitted in little-endian format, i.e. least significant byte first. The index **`3004h`** therefore appears in the frame as **`04h 30h`**.
– Unless stated otherwise, all parameters and results are transmitted as signed 32-bit integers.

| Abbreviation | Data type | Range |
|---|---|---|
| UI8 | unsigned 8-bit integer | 0 … 255 |
| UI16 | unsigned 16-bit integer | 0 … 65535 |
| UI32 | unsigned 32-bit integer | 0 … 4294967295 |
| I32 | signed 32-bit integer | –2147483648 … 2147483647 |
| R32 | 32-bit floating point number as per IEEE 754 | ≈ ± 3.4 × 10^38 |
| RO · RW · WO · Const | read only · read and write · write only · constant | |

## 1.4 Terms and abbreviations

| Term | Meaning |
|---|---|
| bps, kbps | bits per second, kilobits per second (data rate) |
| CAN | Controller Area Network; communication bus for vehicles and automation |
| CANopen | higher-layer communication protocol of the CAN in Automation (CiA) organisation |
| Client-server | communication started by the client with a request; the server always responds |
| COB, COB-ID | communication object (one or more CAN messages) and its identifier |
| Confirmed service | message whose receipt is acknowledged by the receiver (as opposed to an unconfirmed service) |
| DLC | Data Length Code; number of data bytes in a CAN frame |
| ECU | Electronic Control Unit; here the EMFA |
| FDC | PC application for analysis and configuration of the device |
| Heartbeat | periodic CANopen message showing that a node is still present on the network |
| Index, sub-index | 16-bit address in the object dictionary, extended by an 8-bit sub-index for arrays and records |
| J1939 | higher-layer communication protocol of the Society of Automotive Engineers (SAE) |
| LSB, MSB | least and most significant byte of a number |
| LSS | Layer Setting Services; setting of node ID and baud rate |
| Node ID | unique identifier of a node on the bus |
| NMT | Network Management; state control of the nodes |
| NVM | non-volatile memory; retains data without power |
| Object dictionary | list of all communication and application parameters of a CANopen device |
| PDO, TPDO, RPDO | process data object (unconfirmed, high priority); transmitted or received PDO |
| RTR | Remote Transmission Request; requests the data of a node |
| SDO | service data object; confirmed client-server access to the object dictionary |

# 2 Safety instructions

**Caution!** Read and follow this manual before operating or servicing the device. Only trained personnel may operate, clean, inspect, maintain or service the device. Always disconnect the device from the power supply before cleaning or maintenance.

**Warning!** Only qualified personnel may service the device. Take care with checks and adjustments made with power on – failure to do so can result in bodily harm.

**Warning!** Before connecting or disconnecting any wiring between electronic equipment, remove power and wait at least 30 seconds. Otherwise equipment can be damaged or destroyed, and there is a risk of bodily harm.

**Caution!** The device contains electrostatic sensitive components; observe the precautions for handling them.

For parts, information and service, please contact octogon.

# 3 Product description

## 3.1 Function

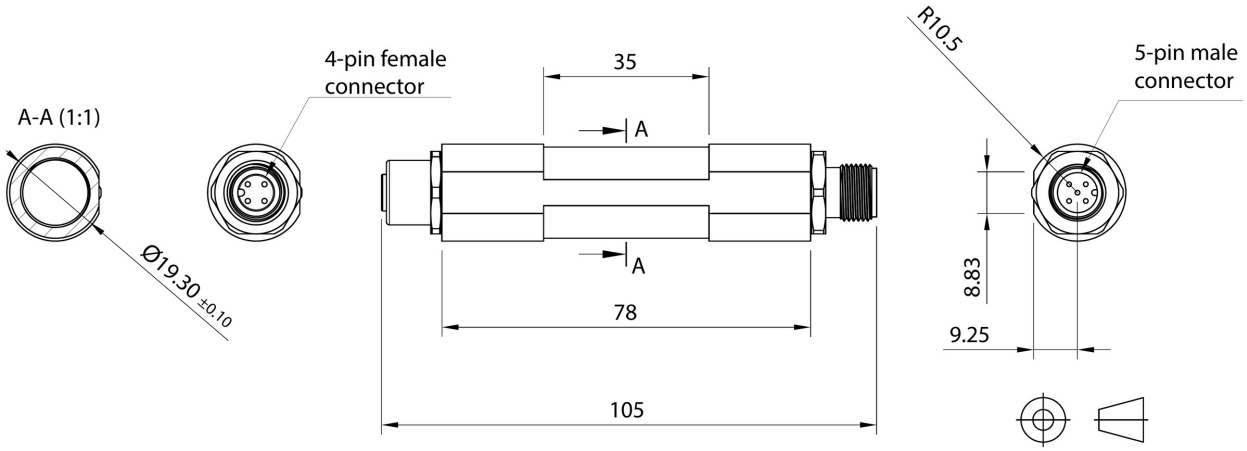
The EMFA is a digitiser for strain gauge load cells with a 24-bit delta-sigma A/D converter, for weighing and force measurements mainly in vehicles and mobile machines. It excites the load cell with 5 V, measures its mV/V signal and transmits it over CAN, either with CANopen (10 kbps to 1 Mbps) or J1939. The 120 Ω termination resistor is built in and is switched on or off over the bus.

Components designed to meet Regulation 10, ISO 13766:2018 and ISO 14982:1998; the unit itself is not certified for these standards – suitability is to be confirmed by the user.

## 3.2 Key data

| Feature | Value |
|---|---|
| Input | 1 strain gauge load cell, bridge output mV/V, 4- or 6-wire |
| Measuring range | ± 3 mV/V |
| Load cell excitation | 5 V DC |
| Load cell impedance | 87.5 … 1,100 Ω (e.g. 1 × 350, 4 × 350, 4 × 1,100 Ω) |
| A/D converter | 24 bit, delta-sigma |
| Filter | selectable: moving average or IIR |
| Update rate CANopen | 5 … 2,500 samples/s |
| Protocol | CANopen or J1939, switchable |
| Baud rate CANopen | 10, 20, 50, 125, 250, 500, 800, 1000 kbit/s |
| Termination resistor | 120 Ω, selectable over CAN |
| Supply voltage | 9 … 36 V DC, via the M12 CAN plug |
| Operating temperature | –20 … +70 °C |
| Storage temperature | –40 … +125 °C |
| Connectors | M12 socket 4-pole (load cell) · M12 plug 5-pole (CAN and supply) |
| Housing | stainless steel 17-4PH, IP67 according to EN 60529 |
| Dimensions, weight | Ø 21 × 105 mm, approx. 120 g |

## 3.3 Dimensions




Dimensions in mm

**Information!** Secure the EMFA, e.g. with P-clips or a bracket – do not support it by the cables only.

# 4 Connection and start-up

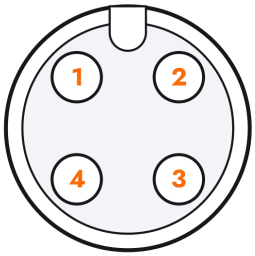
## 4.1 Equipment required

- a USB to CAN converter for CANopen,
- a load cell or scale with test weights, or a load cell simulator,
- a 12 ... 24 V DC power supply capable of delivering approximately 100 mA,
- suitable software, e.g. the FDC application for analysis and configuration.

**Information!** The FDC application was developed with the PEAK-System USB-to-CAN adapter. When the FDC is used with CANopen or J1939, this adapter should be used.

## 4.2 Pin assignment

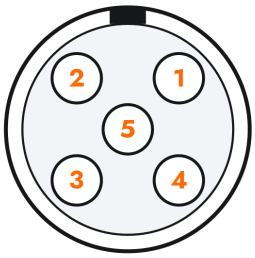
### Load cell input · M12 socket, 4-pole




M12 socket, 4-pole · notch at the top

| | | |
|---|---|---|
| 1 | **+EXC** | + excitation |
| 2 | **+SIG** | + signal |
| 3 | **–EXC** | – excitation |
| 4 | **–SIG** | – signal |

### CAN output · M12 plug, 5-pole




M12 plug, 5-pole · notch at the top

| | | |
|---|---|---|
| 1 | **—** | not used |
| 2 | **+VDC** | supply 9 … 36 V DC |
| 3 | **GND** | ground |
| 4 | **CAN_H** | CAN High |
| 5 | **CAN_L** | CAN Low |

Load cell at the 4-pole M12 socket, CAN bus and supply at the 5-pole M12 plug. The product label of the specific device always applies.

## 4.3 Connecting the load cell

A 4-wire load cell is connected directly to the socket. With a 6-wire load cell, connect Sense+ to Excitation+ (EXC+) and Sense– to Excitation– (EXC–).

| Pin | Signal | 4-wire load cell | 6-wire load cell |
|---|---|---|---|
| 1 | +EXC | excitation + | excitation + and sense+ |
| 2 | +SIG | signal + | signal + |
| 3 | –EXC | excitation – | excitation – and sense– |
| 4 | –SIG | signal – | signal – |

## 4.4 Power supply

The EMFA is supplied via the 5-pole M12 plug, pin 2 (+VDC, 9 … 36 V DC) and pin 3 (GND). The power supply ground is the same potential as the communications: only use the appropriate pins for supply and return, and reference all communication lines to the supply ground.

## 4.5 CAN bus and termination

For the CAN network to operate properly, a 120 Ω termination resistor between CAN High and CAN Low is required both at the host end and at the furthest point of the network. The EMFA has a built-in 120 Ω resistor, enabled at the factory. It is enabled or disabled with the termination resistor parameter (section 7.3.4).

## 4.6 First start-up

– Factory settings: node ID 1, baud rate 500 kbps, heartbeat off, no self-start, mV/V values as integers (× 10000), filter 16-sample moving average at 50 samples/s, termination resistor enabled.
– After power-on the EMFA sends its boot-up message (**`700h`** + ID, data **`00h`**) and stays in the Pre-operational state.
– With the NMT command Operational (**`000h`**: **`01h 00h`**) it starts to transmit TPDO1 with the mV/V signal. To have it do so after every power-on, enable self-start (section 7.3.6).

– Write changed settings to non-volatile memory with "Save setup parameters" (section 7.7.3) – otherwise they are lost at the next power cycle.

# 5 CANopen interface

## 5.1 General information

– The CANopen interface conforms to CAN 2.0B with 11-bit identifiers; the factory baud rate is 500 kbps.

– The EMFA conforms to the requirements for an NMT slave as defined in CiA 302 DSP Part 2: Network Management (February 2009).

– On first connection to the network it sends its boot-up message (a heartbeat with node state 0), followed – if enabled – by periodic heartbeat messages.

– If self-start is configured, it passes through Pre-operational into the Operational state. Otherwise it stays in Pre-operational, waiting for NMT messages or SDO requests.

– The EMFA implements static PDO mapping and a single, statically defined SDO server.

## 5.2 Node ID

Each node on the CANopen network must have a unique 7-bit node ID in the range 1 … 127 (**7Fh**); node ID 0 is reserved for the NMT master. It is set via SDO (section 7.3.2) or via LSS (section 7.4.3).

## 5.3 Network management (NMT)

The NMT protocol is used to tell the nodes to change state and to indicate device boot-up and error conditions. The EMFA supports the following for this purpose:

| Name | CAN-ID | DLC | Data |
|---|---|---|---|
| Boot-up | 700h + ID | 1 | 00h (initialising) |
| Heartbeat | 700h + ID | 1 | 04h Stopped · 05h Operational · 7Fh Pre-operational |
| NMT | 000h | 2 | byte 0: 01h Operational · 02h Stopped · 80h Pre-operational · 81h Reset Node · 82h Reset Communication; byte 1: 00h all nodes or 01h … 7Fh node ID |

The current state dictates how the node communicates on the network:

| State | reached by | SDO | PDO |
|---|---|---|---|
| Initialising | power-on; then Reset Node and Reset Communication, then automatically Pre-operational | – | – |
| Pre-operational | automatically after initialising; NMT 80h | yes | no |
| Operational | NMT 01h; automatically after initialising if self-start is enabled | yes | yes |
| Stopped | NMT 02h | no | no |

Heartbeat messages – if enabled – are transmitted in Pre-operational, Operational and Stopped. Reset Node resets the object dictionary range **2000h** … **FFFFh**, Reset Communication the range **1000h** … **1FFFh**. NMT commands lead from each operating state to each other; power-off or reset ends every state.

## 5.4 Process data objects (PDO)

PDOs transmit real-time measurement and status information. They are unconfirmed and have a higher message priority than SDOs, so they take precedence on the bus.

### 5.4.1 Transmit PDOs (TPDO)

In the Operational state the EMFA transmits the TPDOs on specific events; they can also be requested with a Remote Transmission Request (RTR). A TPDO is disabled by writing a 1 to bit 31 of its COB-ID (**1800h**:1 for TPDO1, **1801h**:1 for TPDO2).

| | TPDO1 | TPDO2 |
|---|---|---|
| Content | net mV/V signal and ECU status | tare signal and ECU status |
| COB-ID | 180h + ID | 280h + ID |
| Trigger | on event: A/D sample available | on event: Set mV/V tare or Reset mV/V tare command executed |
| Data 0 … 3 | R32 or I32 (as per data output options): mV/V value, SDO 3004h:02h | R32 or I32 (as per data output options): tare signal, SDO 3004h:04h |
| Data 4 | UI8: ECU status, SDO 3004h:03h | UI8: ECU status, SDO 3004h:03h |
| DLC | 5 | 5 |

**Information!** Depending on the baud rate, the higher sample rates (up to 2500 samples/s) can exceed what can be transmitted on the CAN bus. Multiple nodes on the network or lower baud rates can also reduce the output rate. In these cases, set a lower sample rate (section 7.2.1).

### 5.4.2 Receive PDOs (RPDO)

In the Operational state the EMFA accepts RPDO1 to set or reset the tare. RPDO1 is disabled by writing a 1 to bit 31 of its COB-ID (**1400h**:1).

| | RPDO1 |
|---|---|
| Content | Set and reset mV/V tare commands |
| COB-ID | 200h + ID |
| Data 0 (DLC = 1) | UI8: set / reset mV/V tare (section 7.6), SDO 3005h:01h |

## 5.5 Service data objects (SDO)

SDOs give access to all entries in the object dictionary: reading device attributes, reading and writing configuration parameters and executing commands. The SDO protocol is a confirmed client-server service; one SDO is made up of two CAN frames with different IDs, the client request and the server response. The EMFA is the SDO server and supports only expedited (single-frame) transfers. SDOs operate only in the Pre-operational and Operational states.

### 5.5.1 The SDO message frame

SDO requests and responses always contain 8 data bytes (DLC = 8), regardless of the data they contain.

| Frame | CAN-ID | D0 | D1 D2 | D3 | D4 … D7 |
|---|---|---|---|---|---|
| Request (client to server) | 600h + ID | command | index | sub-index | data |
| Response (server to client) | 580h + ID | command | index | sub-index | data |

The command byte D0 is split up further (only the services supported by the EMFA):

| Bit | Symbol | Meaning |
|---|---|---|
| 7 … 5 | SP | command specifier. Client: 1 = initiate write, 2 = initiate read, 4 = abort transfer. Server: 2 = initiate read response, 3 = initiate write response, 4 = abort transfer |
| 4 | T | toggle bit; for segmented reads only – not used by the EMFA |
| 3 … 2 | N | unused data bytes (valid only if E = S = 1): 0 = four, 1 = three, 2 = two, 3 = one data byte in the transfer |
| 1 | E | 0 = segmented transfer (not used by the EMFA), 1 = expedited transfer (single frame) |
| 0 | S | 0 = data size not indicated, 1 = data size indicated by bits 3 … 2 |

This gives the command bytes used in the examples: **40h** read; **23h**, **2Bh**, **2Fh** write of 4, 2 or 1 byte; **43h**, **4Bh**, **4Fh** response with 4, 2 or 1 byte; **60h** write confirmed; **80h** abort.

### 5.5.2 SDO abort codes

As the SDO protocol is a confirmed service, the EMFA responds to every request. If it cannot execute a request, it sends an abort message with command specifier **80h** and a 32-bit abort code in the data bytes:

| Code | Description |
|---|---|
| 05040001h | invalid bit combination in the command specifier byte |
| 06010001h | attempt to read a write-only object |
| 06010002h | attempt to write a read-only object |
| 06020000h | object does not exist in the object dictionary (invalid index) |
| 06070010h | incorrect data length for an SDO write |
| 06090011h | sub-index does not exist for the given object |
| 06090030h | value for an SDO write is out of range |
| 08000000h | general error (e.g. the request does not have 8 bytes or there is non-zero data in a read request) |
| 08000022h | conditions not correct to execute the request at this time (e.g. the EMFA is not in administrator mode) |

## 5.6 Emergency messages (EMCY)

On detection of certain errors in the EMFA or on the bus itself, the EMFA sends an unsolicited message containing details of the error:

| | EMCY |
|---|---|
| COB-ID | 80h + ID |
| Trigger | on event: a new error type has been detected |
| Data 0 … 1 | UI16: error code (table below) |
| Data 2 | UI8: error register, SDO 1001h:00h |
| Data 3 … 7 | vendor-specific data |
| DLC | 8 |

| Error code | Description |
|---|---|
| 0000h | error reset – transmitted when all errors are cleared |
| 5001h | an error has been detected in the A/D converter |
| 5002h | an open circuit has been detected in one of the load cell connections |
| 5003h | the load cell excitation voltage is too low |
| 8110h | CAN overrun – a message has been lost |
| 8120h | the EMFA CAN controller has entered the Error Passive state |
| 8140h | the EMFA CAN controller has recovered from a Bus Off condition |
| 8150h | COB-ID collision: another device on the bus has the same node ID |
| 8210h | an RPDO1 message has been received with an incorrect data length |
| FF01h | non-recoverable fault in the factory settings memory – return the device |
| FF02h | error in the configuration data memory – recoverable by reconfiguring all user parameters |

# 6 Object dictionary

The following tables show the object dictionary entries needed for access to the EMFA data. For data types and access types see section 1.3.

## 6.1 Communication segment

| Index | Sub | Name | Type | Access | Default | Information |
|---|---|---|---|---|---|---|
| 1000h | 0 | Device type | UI32 | RO | 0 | non-standard device profile |
| 1001h | 0 | Error register | UI8 | RO | 0 | bit 0 generic error, bit 4 communication error, bit 7 manufacturer-specific error |
| 1003h | | Pre-defined error field | | | | |
| | 0 | Number of errors | UI8 | RW | 0 | read the number; write 0 to erase the history |
| | 1 … 8 | Standard error field | UI32 | RO | – | most recent errors (EMCY sent) |
| 1010h | | Store parameters | | | | |
| | 0 | Number of entries | UI8 | RO | 4 | |
| | 1 | Save all parameters | UI32 | RW | 1 | save setup parameters (7.7.3) |
| | 2 | Dummy | UI32 | RO | 0 | not supported |
| | 3 | Dummy | UI32 | RO | 0 | entries 6000h … 9FFFh not supported |
| | 4 | Reserved | UI32 | RW | – | do not write to this entry |
| 1011h | | Restore default parameters | | | | |
| | 0 | Number of entries | UI8 | RO | 1 | |
| | 1 | Restore all defaults | UI32 | RW | 1 | restore factory defaults (7.7.4) |
| 1014h | 0 | Emergency COB-ID | UI32 | RO | 80h + ID | CAN ID of the EMCY message |
| 1017h | 0 | Producer heartbeat time | UI16 | RW | 0 | cycle time in ms (7.3.5) |
| 1018h | | Identity object | | | | |
| | 0 | Number of entries | UI8 | RO | 4 | |
| | 1 | Vendor ID | UI32 | RO | 044Ah | |
| | 2 | Product code | UI32 | RO | device-specific | application firmware part number (7.1.1) |
| | 3 | Revision number | UI32 | RO | version-specific | application firmware version (7.1.2) |
| | 4 | Serial number | UI32 | RO | – | device-specific serial number (7.1.3) |
| 1026h | | OS prompt | | | | |
| | 0 | Number of entries | UI8 | RO | 2 | |
| | 1 | StdIn | UI8 | WO | – | reserved for the manufacturer |
| | 2 | StdOut | UI8 | RO | – | reserved for the manufacturer |
| 1400h | | RPDO1 communication parameter | | | | |
| | 0 | Number of entries | UI8 | RO | 2 | |
| | 1 | COB-ID of RPDO1 | UI32 | RW | 200h + ID | set bit 31 to disable RPDO1 |
| | 2 | Transmission type | UI8 | RO | FFh | asynchronous |
| 1600h | | RPDO1 mapping | | | | |
| | 0 | Number of entries | UI8 | Const | 1 | |
| | 1 | Mapping entry 1 | UI32 | Const | 30050108h | mV/V tare commands (7.6) |
| 1800h | | TPDO1 communication parameter | | | | |
| | 0 | Number of entries | UI8 | RO | 2 | |
| | 1 | COB-ID of TPDO1 | UI32 | RW | 180h + ID | set bit 31 to disable TPDO1 |
| | 2 | Transmission type | UI8 | RO | FEh | manufacturer-specific |
| 1801h | | TPDO2 communication parameter | | | | |
| | 0 | Number of entries | UI8 | RO | 2 | |
| | 1 | COB-ID of TPDO2 | UI32 | RW | 280h + ID | set bit 31 to disable TPDO2 |
| | 2 | Transmission type | UI8 | RO | FEh | manufacturer-specific |

| Index | Sub | Name | Type | Access | Default | Information |
|---|---|---|---|---|---|---|
| 1A00h | | TPDO1 mapping | | | | |
| | 0 | Number of entries | UI8 | Const | 2 | |
| | 1 | Mapping entry 1 | UI32 | Const | 30040220h | filtered mV/V signal (7.5.2) |
| | 2 | Mapping entry 2 | UI32 | Const | 30040308h | ECU status (7.5.3) |
| 1A01h | | TPDO2 mapping | | | | |
| | 0 | Number of entries | UI8 | Const | 2 | |
| | 1 | Mapping entry 1 | UI32 | Const | 30040420h | tare signal (7.5.4) |
| | 2 | Mapping entry 2 | UI32 | Const | 30040308h | ECU status (7.5.3) |
| 1F80h | 0 | NMT start-up | UI32 | RW | 4 (no self-start) | self-start (7.3.6) |

## 6.2 Manufacturer segment

| Index | Sub | Name | Type | Access | Default | Information |
|---|---|---|---|---|---|---|
| 3000h | | Device identification | | | | |
| | 0 | Number of entries | UI8 | RO | 4 | |
| | 1 | Bootloader part number | UI32 | RO | – | 7.1.4 |
| | 2 | Bootloader f/w version | UI32 | RO | – | 7.1.5 |
| 3002h | | General parameters | | | | |
| | 0 | Number of entries | UI8 | RO | 3 | |
| | 1 | ADC sample rate | I32 | RW | 50 (Hz) | 7.2.1 |
| | 2 | Filter type | I32 | RW | 2 | digital filter (7.2.2) |
| | 3 | Warm-up time | I32 | RW | 0 | time for system stabilisation (7.2.3) |
| 3003h | | Communication parameters | | | | |
| | 0 | Number of entries | UI8 | RO | 4 | |
| | 1 | Baud rate | I32 | RW | 500000 (bps) | 7.3.1 |
| | 2 | Node ID | I32 | RW | 1 | 7.3.2 |
| | 3 | Bus protocol | I32 | RW | – | CANopen or J1939 (7.3.3) |
| | 4 | R120 status | I32 | RW | 1 (enabled) | termination resistor (7.3.4) |
| 3004h | | Data access | | | | |
| | 0 | Number of entries | UI8 | RO | 5 | |
| | 1 | Data output options | UI8 | RW | 0 (I32) | integer or floating point (7.5.1) |
| | 2 | Filtered mV/V signal | R32 / I32 | RO | – | latest filtered value (7.5.2) |
| | 3 | ECU status | UI8 | RO | – | status flags (7.5.3) |
| | 4 | mV/V tare signal | R32 / I32 | RO | – | tare value (7.5.4) |
| | 5 | Filtered ADC sample | I32 | RO | – | latest filtered ADC output (7.5.5) |
| 3005h | | mV/V tare control | | | | |
| | 0 | Number of entries | UI8 | RO | 1 | |
| | 1 | Set and reset mV/V tare | UI8 | WO | – | 7.6 |
| 3007h | | Miscellaneous | | | | |
| | 0 | Number of entries | UI8 | RO | 3 | |
| | 1 | System reset | I32 | WO | – | warm start (7.7.1) |
| | 2 | Write passcode register | I32 | WO | – | administrator mode (7.7.2) |
| | 3 | Reserved | I32 | WO | – | reserved for the manufacturer |
| 3008h | | User parameters | | | | |
| | 0 | Number of entries | UI8 | RO | 4 | |
| | 1 … 4 | User parameter 1 … 4 | I32 | RW | 0 | 7.7.5 |

# 7 Commands

This section describes all commands the EMFA accepts via SDO, each with an example of request and response. "ID" stands for the node ID. For the frame layout and the command bytes see section 5.5.

## 7.1 Device identification

### 7.1.1 Read firmware part number (1018h:02h)

Returns the application firmware part number; it is also the product code for LSS. The value is device-specific — 112328 in the example.

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 18h 10h | 02h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 18h 10h | 02h | C8h B6h 01h 00h |

### 7.1.2 Read application firmware version (1018h:03h)

Data bytes 4 and 5: minor version (LSB, MSB), data bytes 6 and 7: major version (LSB, MSB). The example returns version 1.0.

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 18h 10h | 03h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 18h 10h | 03h | 00h 00h 01h 00h |

### 7.1.3 Read serial number (1018h:04h)

Returns the device-specific serial number as an unsigned 32-bit integer; 2052999 in the example.

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 18h 10h | 04h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 18h 10h | 04h | 87h 53h 1Fh 00h |

### 7.1.4 Read bootloader part number (3000h:01h)

Part number 109960 in the example.

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 00h 30h | 01h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 00h 30h | 01h | 88h ADh 01h 00h |

### 7.1.5 Read bootloader firmware version (3000h:02h)

Data byte 4: minor version, data byte 5: major version, data bytes 6 and 7: compatibility number (LSB, MSB). The example returns version 2.1, compatibility number 8177.

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 00h 30h | 02h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 00h 30h | 02h | 01h 02h F1h 1Fh |

## 7.2 General parameters

### 7.2.1 ADC sample rate (3002h:01h)

Reads and sets the A/D converter sample rate: 5 … 2500 samples/s, factory default 50 samples/s. The IIR filters have a fixed sample rate; if such a filter is selected, writing a new value has no effect (section 7.2.2).

Read — the response returns 125 samples/s:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 02h 30h | 01h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 02h 30h | 01h | 7Dh 00h 00h 00h |

Write – set 250 samples/s:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 02h 30h | 01h | FAh 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 02h 30h | 01h | 00h 00h 00h 00h |

### 7.2.2 Filter type (3002h:02h)

Reads and sets the digital filter. Factory default: 2 (16-sample moving average). The IIR filters each have a fixed sample rate; the moving average filters use the configured sample rate (section 7.2.1).

| Value | Filter type |
|---|---|
| 00h | no filtering |
| 01h · 02h · 03h · 04h | 8 · 16 · 32 · 64-sample moving average |
| 20h … 25h | IIR, 40 SPS, cut-off 0.2 · 0.5 · 1 · 2 · 5 · 10 Hz |
| 26h … 2Dh | IIR, 600 SPS, cut-off 0.25 · 0.5 · 1 · 2 · 3 · 4 · 8 · 18 Hz |

The values within a row belong together in order, e.g. **22h** = IIR, 40 SPS, 1 Hz and **28h** = IIR, 600 SPS, 1 Hz.

Read – the response shows the 32-sample moving average filter:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 02h 30h | 02h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 02h 30h | 02h | 03h 00h 00h 00h |

Write – set the IIR filter 600 SPS, 1 Hz:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 02h 30h | 02h | 28h 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 02h 30h | 02h | 00h 00h 00h 00h |

### 7.2.3 Warm-up time (3002h:03h)

The warm-up time is the time from power-on or system reset during which the mV/V signal is reported as the under-range value –109, to avoid false readings while the system is stabilising. Range 0 … 3600 s, factory default 0 s.

Read – the response returns 15 s:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 02h 30h | 03h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 02h 30h | 03h | 0Fh 00h 00h 00h |

Write – set 30 s:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 02h 30h | 03h | 1Eh 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 02h 30h | 03h | 00h 00h 00h 00h |

## 7.3 Communications

### 7.3.1 Baud rate (3003h:01h)

Reads and sets the CAN bus speed in bit/s: 10000, 20000, 50000, 125000, 250000, 500000, 800000 or 1000000, factory default 500000. Once saved with "Save setup parameters", the new baud rate is enabled after the next power cycle or system reset. It can also be set via LSS (section 7.4).

Read – the response returns 500000 bps:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 03h 30h | 01h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 03h 30h | 01h | 20h A1h 07h 00h |

Write – set 1 Mbps:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 03h 30h | 01h | 40h 42h 0Fh 00h |
| EMFA responds | 580h + ID | 60h | 03h 30h | 01h | 00h 00h 00h 00h |

### 7.3.2 Node ID (3003h:02h)

Reads and sets the node ID: **01h** … **7Fh**, factory default **01h**. Once saved, it becomes active after the next power cycle or system reset. It can also be set via LSS (section 7.4).

Read – the response returns **33h**:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 03h 30h | 02h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 03h 30h | 02h | 33h 00h 00h 00h |

Write – set node ID **0Fh**:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 03h 30h | 02h | 0Fh 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 03h 30h | 02h | 00h 00h 00h 00h |

### 7.3.3 Bus protocol (3003h:03h)

Switches between CANopen (**12Dh**) and J1939 (**793h**). Once saved, the new protocol is enabled after the next power cycle or system reset. The EMFA must be in administrator mode to write this parameter (section 7.7.2). "Restore factory defaults" does not change this parameter.

Read – the response shows: CANopen after the next power cycle:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 03h 30h | 03h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 03h 30h | 03h | 2Dh 01h 00h 00h |

Write – J1939 after the next power cycle:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 03h 30h | 03h | 93h 07h 00h 00h |
| EMFA responds | 580h + ID | 60h | 03h 30h | 03h | 00h 00h 00h 00h |

### 7.3.4 CAN termination resistor (3003h:04h)

Reads and sets the built-in 120 Ω termination resistor: 0 = disabled, 1 = enabled, factory default 1 (section 4.5).

Read – the response shows: disabled:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 03h 30h | 04h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 03h 30h | 04h | 00h 00h 00h 00h |

Write – enable:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 03h 30h | 04h | 01h 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 03h 30h | 04h | 00h 00h 00h 00h |

### 7.3.5 Producer heartbeat time (1017h:00h)

The EMFA implements the heartbeat protocol to monitor the network; the message – if enabled – is transmitted in Pre-operational, Operational and Stopped. Node guarding is not supported. The parameter is the period between two heartbeat messages in ms (UI16): 0 (disabled) … 65535, factory default 0.

Read – the response returns 250 ms:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 17h 10h | 00h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 4Bh | 17h 10h | 00h | FAh 00h 00h 00h |

Write – disable the heartbeat:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 2Bh | 17h 10h | 00h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 17h 10h | 00h | 00h 00h 00h 00h |

### 7.3.6 Self-start / NMT start-up (1F80h:00h)

The EMFA supports only the NMT Master Start flag of this object: whether it switches to Operational on its own at start-up and then transmits TPDO1 without an NMT message. **00h** = self-start, **04h** = stays in Pre-operational; factory default 4.

Read – the response shows: no self-start:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 80h 1Fh | 00h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 80h 1Fh | 00h | 04h 00h 00h 00h |

Write – enable self-start:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 80h 1Fh | 00h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 80h 1Fh | 00h | 00h 00h 00h 00h |

## 7.4 Layer Setting Services (LSS)

Node ID and baud rate can also be set via LSS. The current node ID is not used in the protocol, so LSS can be used if the node ID is not known or two nodes share the same ID. The EMFA supports:

| Code | Service |
|---|---|
| 04h | switch state global |
| 11h · 13h · 15h · 17h | configure node ID · configure bit timing · activate bit timing · store configuration |
| 40h · 41h · 42h · 43h | switch state selective: select vendor ID · product code · revision number · serial number |
| 5Ah · 5Bh · 5Ch · 5Dh · 5Eh | inquire vendor ID · product code · revision number · serial number · node ID |

Before configuring, the device must be selected for configuration – globally or selectively.

### 7.4.1 Switch state global

Selects all devices on the bus with a single command. Mode **00h** = waiting (no configuration via LSS), **01h** = configuration. The service is unconfirmed – no node responds.

| Direction | CAN-ID | DLC | Data | Meaning |
|---|---|---|---|---|
| Controller sends | 7E5h | 8 | 04h 01h 00h 00h 00h 00h 00h 00h | all devices into configuration state |

### 7.4.2 Switch state selective

Selects a specific node. Four messages are sent, each identifying a different attribute; only when all four match does the node enter the configuration state and respond.

| Attribute | EMFA value |
|---|---|
| Vendor ID | 044Ah |
| Product code | application firmware part number (1018h:02h, section 7.1.1) |
| Revision number | application firmware version (1018h:03h) |
| Serial number | device serial number (1018h:04h) |

Example: select the device with product code 112328, version 1.0 and serial number 202101999:

| Direction | CAN-ID | DLC | Data | Meaning |
|---|---|---|---|---|
| Controller sends | 7E5h | 8 | 40h 4Ah 04h 00h 00h 00h 00h 00h | vendor ID |
| Controller sends | 7E5h | 8 | 41h C8h B6h 01h 00h 00h 00h 00h | product code |
| Controller sends | 7E5h | 8 | 42h 00h 00h 01h 00h 00h 00h 00h | revision number |
| Controller sends | 7E5h | 8 | 43h EFh D4h 0Bh 0Ch 00h 00h 00h | serial number |
| EMFA responds | 7E4h | 8 | 44h 00h 00h 00h 00h 00h 00h 00h | all four attributes matched |

### 7.4.3 Configure node ID

On the selected device, the node ID is set with "Configure node ID", the NMT Reset Communication message and "Store configuration" (**01h** … **7Fh**). Example for node ID **3Bh**:

| Direction | CAN-ID | DLC | Data | Meaning |
|---|---|---|---|---|
| Controller sends | 7E5h | 8 | 11h 3Bh 00h 00h 00h 00h 00h 00h | configure node ID |
| EMFA responds | 7E4h | 8 | 11h 00h 00h 00h 00h 00h 00h 00h | error code 0, spec error 0 |
| Controller sends | 000h | 2 | 82h 00h | reset communications, all nodes |
| EMFA responds | 73Bh | 1 | 00h | boot-up with 700h + new node ID |
| Controller sends | 7E5h | 8 | 17h 00h 00h 00h 00h 00h 00h 00h | store configuration |
| EMFA responds | 7E4h | 8 | 17h 00h 00h 00h 00h 00h 00h 00h | error code 0, spec error 0 |

### 7.4.4 Configure bit timing

On the selected device, the baud rate is set with "Configure bit timing", "Activate bit timing" and "Store configuration". The EMFA uses a manufacturer-specific bit timing table (table selector **80h**):

| Table index | CAN bus speed |
|---|---|
| 0 · 1 · 2 · 3 · 4 | 1 Mbps · 800 kbps · 500 kbps · 250 kbps · 125 kbps |
| 5 | reserved, do not use |
| 6 · 7 · 8 | 50 kbps · 20 kbps · 10 kbps |

Example for 125 kbps:

| Direction | CAN-ID | DLC | Data | Meaning |
|---|---|---|---|---|
| Controller sends | 7E5h | 8 | 13h 80h 04h 00h 00h 00h 00h 00h | table 80h, index 4 |
| EMFA responds | 7E4h | 8 | 13h 00h 00h 00h 00h 00h 00h 00h | error code 0, spec error 0 |
| Controller sends | 7E5h | 8 | 15h 32h 00h 00h 00h 00h 00h 00h | activate, switch delay 50 ms (UI16) |
| Controller sends | 7E5h | 8 | 17h 00h 00h 00h 00h 00h 00h 00h | store configuration |
| EMFA responds | 7E4h | 8 | 17h 00h 00h 00h 00h 00h 00h 00h | error code 0, spec error 0 |

The switch delay is the time in ms the nodes wait before accepting the new bit rate; after a second delay of the same length they transmit again. "Activate bit timing" is unconfirmed — no response is sent.

## 7.5 Data access

### 7.5.1 Data output options (3004h:01h)

Determines the format of all mV/V data (UI8). Bit 0 = IEEE 754 flag: 0 = signed 32-bit integer, multiplied by 10000; 1 = 32-bit floating point number as per IEEE 754. Bits 7 … 1 are always 0. Factory default 0 (integer).

Read — the response shows: integer format:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 04h 30h | 01h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 4Fh | 04h 30h | 01h | 00h 00h 00h 00h |

Write — output in IEEE 754 format:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 2Fh | 04h 30h | 01h | 01h 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 04h 30h | 01h | 00h 00h 00h 00h |

### 7.5.2 Read filtered mV/V signal (3004h:02h)

Returns the latest value of the load cell signal in mV/V after the selected filter – as net value, with any tare already subtracted. Format as per the data output options. Above +3.3 mV/V the over-range value 109 is returned. The under-range value –109 is returned while the warm-up time is running, the signal is below –3.3 mV/V or a fault flag in the ECU status is set.

Read – the response returns 11084 (integer), i.e. 1.1084 mV/V:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 04h 30h | 02h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 04h 30h | 02h | 4Ch 2Bh 00h 00h |

### 7.5.3 Read ECU status (3004h:03h)

Returns the status of the EMFA as a set of bit-mapped flags (UI8):

| Bit | Value | Condition |
|---|---|---|
| 7 | 80h | critical fault, non-recoverable – return the device |
| 6 | 40h | load cell fault (open circuit or low excitation voltage) |
| 5 | 20h | configuration fault – recoverable by configuring all parameters |
| 4 | 10h | copy of the IEEE 754 flag in the data output options |
| 3 | 08h | mV/V signal above the maximum limit of +3.3 mV/V |
| 2 | 04h | mV/V signal below the minimum limit of –3.3 mV/V |
| 1 | 02h | mV/V net/tare function active |
| 0 | 01h | warm-up active (warm-up time not yet expired) |

Read – the response **12h** shows: output in IEEE 754 format and tare active:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 04h 30h | 03h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 4Fh | 04h 30h | 03h | 12h 00h 00h 00h |

### 7.5.4 Read mV/V tare signal (3004h:04h)

Returns the current tare signal level in mV/V, format as per the data output options; zero if the net/tare function is not active.

Read – the response returns 0.2003 mV/V (IEEE 754):

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 04h 30h | 04h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 04h 30h | 04h | 71h 1Bh 4Dh 3Eh |

### 7.5.5 Read filtered ADC sample (3004h:05h)

Returns the last calculated ADC value after the selected filter; 10255261 counts in the example.

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 04h 30h | 05h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 04h 30h | 05h | 9Dh 7Bh 9Ch 00h |

## 7.6 Set and reset mV/V tare (3005h:01h)

Each bit of the parameter (UI8) represents a command executed only if that bit is set: bit 0 (**01h**) = set mV/V tare, bit 1 (**02h**) = reset mV/V tare, bits 7 ... 2 always 0. The commands run from bit 0 – with **03h**, set is executed first, immediately followed by reset. The object is mapped to RPDO1 and can be used as an unconfirmed service in the Operational state (section 5.4.2).

### 7.6.1 Set mV/V tare

**01h** sets the tare to the current mV/V signal level and activates the net/tare function; all subsequent readings are offset from this value. The command is rejected if the signal is outside ± 3.3 mV/V, the warm-up time has not yet expired or a fault flag in the ECU status is set. When executed, the tare flag in the ECU status is set and – in Operational – TPDO2 is broadcast.

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 2Fh | 05h 30h | 01h | 01h 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 05h 30h | 01h | 00h 00h 00h 00h |

### 7.6.2 Reset mV/V tare

**02h** resets the tare to zero and deactivates the net/tare function; all subsequent readings are the true load cell signal. The tare flag in the ECU status is cleared and – in Operational – TPDO2 is broadcast.

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 2Fh | 05h 30h | 01h | 02h 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 05h 30h | 01h | 00h 00h 00h 00h |

## 7.7 Miscellaneous commands

### 7.7.1 System reset (3007h:01h)

Writing any value instructs the EMFA to perform a full firmware reset (warm start). The response is returned before the reset; initialisation afterwards can take up to 250 ms, during which messages may not be received.

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 07h 30h | 01h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 60h | 07h 30h | 01h | 00h 00h 00h 00h |

### 7.7.2 Write passcode register (3007h:02h)

Writes a 32-bit integer into the passcode register. If it matches the system passcode (default **9A52Fh**), the EMFA enters administrator or OEM mode, allowing changes to restricted parameters and access to restricted functions. An incorrect code outside these modes returns an error, and the command becomes unavailable for around 5 s. The mode ends when the register is written again while in the mode (a success response is returned), when no administrator command is executed for around 10 minutes, or on a power cycle or system reset.

Write – enter administrator mode:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 07h 30h | 02h | 2Fh A5h 09h 00h |
| EMFA responds | 580h + ID | 60h | 07h 30h | 02h | 00h 00h 00h 00h |

### 7.7.3 Save setup parameters (1010h:01h)

Writing the value **65766173h** saves changed setup parameters to non-volatile memory; unsaved changes are lost on a power cycle or system reset. Saved are: RPDO1, TPDO1 and TPDO2 disable bits, ADC sample rate, filter type, warm-up time, baud rate, node ID, bus protocol, termination resistor status, producer heartbeat time, NMT start-up, data output options and user parameters 1 ... 4.

**Information!** Saving can take up to 75 ms. The response only means that the request was valid and saving has begun. The CPU is halted during this time; messages sent within 75 ms may be lost.

| Direction | CAN-ID | Command | Index | Sub | Data (D4 ... D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 10h 10h | 01h | 73h 61h 76h 65h |
| EMFA responds | 580h + ID | 60h | 10h 10h | 01h | 00h 00h 00h 00h |

### 7.7.4 Restore factory defaults (1011h:01h)

Writing the value **64616F6Ch** overwrites all setup parameters except the bus protocol with the factory defaults and saves them at once — "Save setup parameters" is not needed. The 75 ms note of section 7.7.3 applies. The EMFA must be in administrator mode.

| Parameter | Index:Sub | Default value |
|---|---|---|
| Producer heartbeat time | 1017h:00h | 0 (heartbeat disabled) |
| RPDO1 · TPDO1 · TPDO2 COB-ID | 1400h · 1800h · 1801h:01h | 200h · 180h · 280h + ID (enabled) |
| NMT start-up | 1F80h:00h | 04h (self-start disabled) |
| ADC sample rate | 3002h:01h | 50 samples/s |
| Filter type | 3002h:02h | 2 (16-sample moving average) |
| Warm-up time | 3002h:03h | 0 s |
| Baud rate | 3003h:01h | 500000 bps |
| Node ID | 3003h:02h | 1 |
| CAN termination resistor | 3003h:04h | 1 (enabled) |
| Data output options | 3004h:01h | 0 (mV/V data in integer format) |
| User parameters 1 … 4 | 3008h:01h … 04h | 0 |

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 11h 10h | 01h | 6Ch 6Fh 61h 64h |
| EMFA responds | 580h + ID | 60h | 11h 10h | 01h | 00h 00h 00h 00h |

### 7.7.5 User parameters 1 … 4 (3008h:01h … 04h)

Four locations for arbitrary user data, not used by the firmware. Reading returns the value, writing overwrites it. Store new values afterwards with "Save setup parameters". The EMFA must be in administrator mode to use this command.

Write — user parameter 3 = –16180:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 23h | 08h 30h | 03h | CCh C0h FFh FFh |
| EMFA responds | 580h + ID | 60h | 08h 30h | 03h | 00h 00h 00h 00h |

Read — user parameter 3, response –16180:

| Direction | CAN-ID | Command | Index | Sub | Data (D4 … D7) |
|---|---|---|---|---|---|
| Controller sends | 600h + ID | 40h | 08h 30h | 03h | 00h 00h 00h 00h |
| EMFA responds | 580h + ID | 43h | 08h 30h | 03h | CCh C0h FFh FFh |

# 8 Messages and error values

Overview of how an error shows up and where it is described:

| Symptom | Cause | see |
|---|---|---|
| mV/V value 109 | signal above +3.3 mV/V | 7.5.2 |
| mV/V value –109 | warm-up running, signal below –3.3 mV/V or fault flag in the ECU status | 7.5.2, 7.5.3 |
| ECU status bit 6, EMCY 5002h / 5003h | open circuit in a load cell line or excitation too low — check the connection | 4.3, 5.6 |
| EMCY 8150h | two nodes with the same node ID — change via LSS | 7.4 |
| EMCY 8110h | message lost — lower the sample rate | 5.4.1, 7.2.1 |
| SDO abort 08000022h | command requires administrator mode | 7.7.2 |
| EMCY FF02h, ECU status bit 5 | configuration memory faulty — set all parameters again and save them | 7.7.3 |
| EMCY FF01h, ECU status bit 7 | non-recoverable — return the device | 5.6 |
| setting lost after power-on | not saved | 7.7.3 |