# MicroControl

## Systemhaus für Automatisierung

# µCAN.4.ao-BOX

Manual analogue output module
Version 1.00

**Document conventions**

For better handling of this manual the following icons and head-lines are used:

This symbol marks a paragraph containing useful information about the device operation or giving hints on configuration.

This symbol marks a paragraph which explains possible danger. This danger might cause a damage to the system or damage to personnel. Read these sections carefully!

**Keywords**      Important keywords appear in the border column to help the reader when browsing through this document.

# Contents

# Contents

# 1. Safety Regulations

**Please read the following chapter in any case, because it contains important information about the secure handling of electrical devices.**

## 1.1 General Safety Regulations

This paragraph gives important information about the conditions of use. It was written for personnel which is qualified and trained on electrical devices.

Qualified and trained personnel are persons who fulfil at least one of the following conditions:

● You know the safety regulations for automated machines and you are familiar with the machine.

● You are the operator for the machine and you have been trained on operation modes. You are familiar with the operation of devices described in this manual.

● You are responsible for setting into operation or service and you are trained on repairing automated machines. In addition you are trained in setting electrical devices into operation, to connect the earthing conductor and to label these devices.

The devices described in this manual may only be used for the mentioned applications. Other devices used in conjunction have to meet the safety regulations and EMI requirements.

To ensure a trouble free and safe operation of the device please take care of proper transport, appropriate storage, proper assembly as well as careful operation and maintenance.

Please take care to observe the actual local safety regulations.

If devices are used in a fixed machine without a mains switch for all phases or fuses, this equipment has to be installed. The fixed machine must be connected to safety earth.

**1**

If devices are supplied by mains please take care that the selected input voltage fits to the local mains.

## 1.2 Safety Notice

If devices are supplied by 24V DC, this voltage has to be isolated from other voltages.

The cables for power supply, signal lines and sensor lines must be installed in a way that the device function is not influenced by EMI.

Devices or machines for industrial automation must be constructed in a manner that an unintentional operation is impossible.

By means of hardware and software safety precautions have to be taken in order to avoid undefined operation of an automated machine in case of a cable fraction.

If automated machines can cause damage of material or personnel in case of a malfunction the system designer has to take care for safety precautions. Possible safety precautions might be a limit switch or locking.

## 2. Operation of µCAN.4.ao-BOX

### 2.1 Overview

The µCAN.4.ao-BOX is the right solution for the digital to analogue output of standard signals via CAN. Just sending the digital values over the CAN will generate a high precision analogue signal ( +/-10V or 0..20mA ) on the outputs.



*Fig. 1: Four channel analogue output module µCAN.4.ao-BOX*

Use of a fieldbus for signal acquisition and signal generating has the advantage of reduced costs because expensive I/O cards for a PLC or PC can be omitted. In addition, the design of an application is more flexible and modifications are more easily to achieve.

**2**

The development in automation towards decentralized „intelligent" systems makes the communication between these components quite important.

Modern automated systems require the possibility to integrate components from different manufacturers. The solution for this problem is a common bus system.

All these requirements are fulfilled by the µCAN.4.ao-BOX module. The µCAN.4.ao-BOX runs on the standard fieldbus CAN.

Typical applications for the µCAN.4.ao-BOX are industrial automation, transportation, food industry and environmental technology.

The µCAN.4.ao-BOX operates with the CAN protocol



according to DS-301 (version 4.02). Other protocol stacks are available on request.

space saving and compact

The µCAN.4.ao-BOX is designed for heavy duty applications. The aluminium cast ensures protection class IP66. The compact, space saving case gives the freedom to mount the module in many places.

cost-effective and service friendly

The quick and easy integration of the µCAN.4.ao-BOX in your application reduces the development effort. Costs for material and personnel are reduced. The easy installation makes maintenance and replacement quite simple.

# 3. Project Planning

The chapter Project Planning contains information which are important for the system engineer when using the µCAN.4.ao-BOX. These information include case dimensions and conditions of use.

## 3.1 Module Layout

The following figure shows the top view of the µCAN.4.ao-BOX PCB. Use the figure to identify the terminal blocks, LED's and DIP-switches.
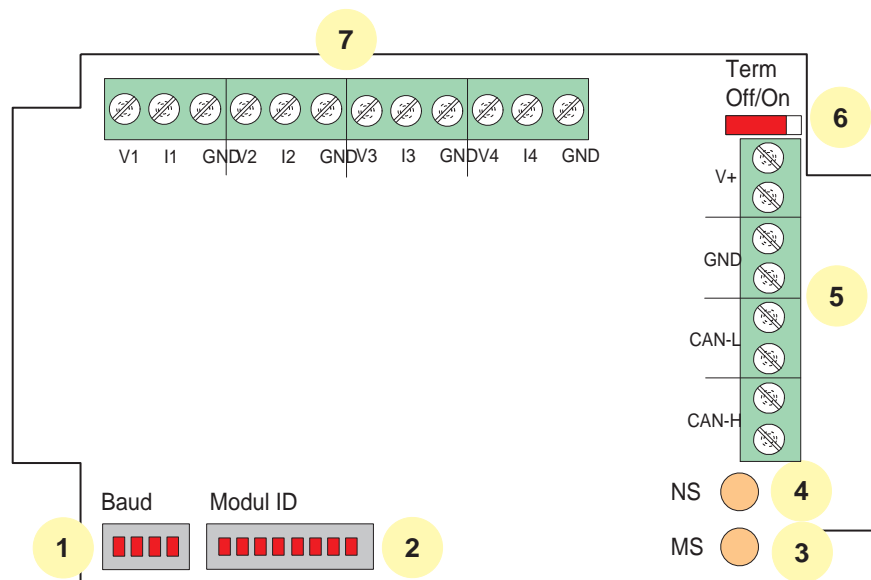
1: Baudrate switch
2: Module address / ID
3: Bi-color LED for module status
4: Bi-color LED for network status
5: Terminal block for Power / CAN
6: Switch for CANbus termination
7: Bi-color LED for signal status
8: Terminal block for digital signals

*Fig. 2: Top view of the µCAN.4.ao-BOX PCB*

## 3.2 Operation Area

The µCAN.4.ao-BOX is a robust field module for the output of analogue standard signals such as +/-10V or 0(4)..20mA. The output configuration is done via software communication over the CANbus. The signals are output on the according terminal block with a high precision of 16bit resolution. The module has a power supply range of 12V - 60V DC.

The µCAN.4.ao-BOX needs a four core cable for connection of power supply and CAN bus, in order to reduce the amount of cabling. Special CAN bus cables are available as accessories.

## 3.3 Maximum System Configuration

For an operational system at least one network manager must be connected to the bus. This network manager might be a PLC or PC equipped with a CAN card. Every µCAN.4.ao-BOX module is an active node.

A CANopen network manager can access **logically** up to 127 CANopen slaves (refer to Fig. 3, "Maximum system configuration"). Every module gets a unique address, which is set up via a DIP switch. The CANbus bus is connected through the µCAN modules. The last module in the network must be terminated by a termination switch (refer to "Termination" on page 25).

**3**
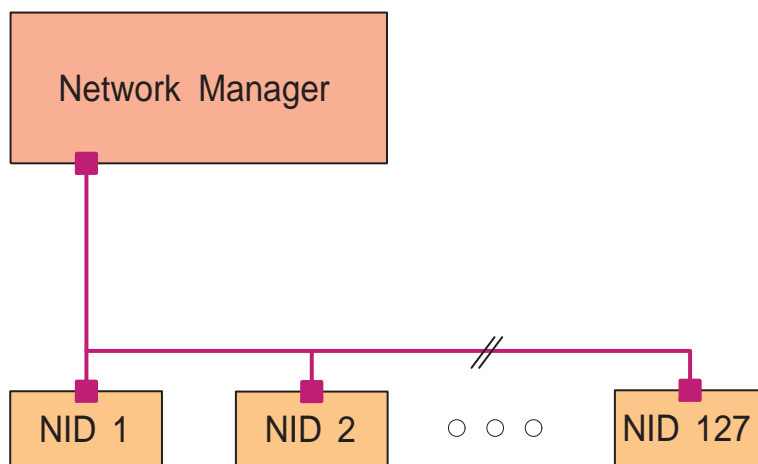


*Fig. 3: Maximum system configuration*

The maximum cable length depends on the selected baudrate. The following table shows the maximum cable length recommended by the CAN in Automation (http://www.can-cia.org). These distances can be realized with the µCAN.4.ao-BOX.

| Baudrate | Cable length |
|----------|--------------|
| 1000 kBit/s | 25 m |
| 800 kBit/s | 50 m |
| 500 kBit/s | 100 m |
| 250 kBit/s | 250 m |
| 125 kBit/s | 500 m |
| 100 kBit/s | 650 m |
| 50 kBit/s | 1000 m |
| 20 kBit/s | 2500 m |

*Tabelle 1: Dependence of baudrate from cable length*

It is recommended by the CAN in Automation **not to use** the baudrate 100 kBit/s in new CANopen systems.

## 3.4 Case Dimensions

The case dimensions of the module are given in the following drawing. The high protection class IP66 of the module allows an assembly at places with a harsh environment. It is possible to mount the module inside a switching cabinet as well as direct on a machine. Please check the technical data section for detailled information about maximum environment conditions.
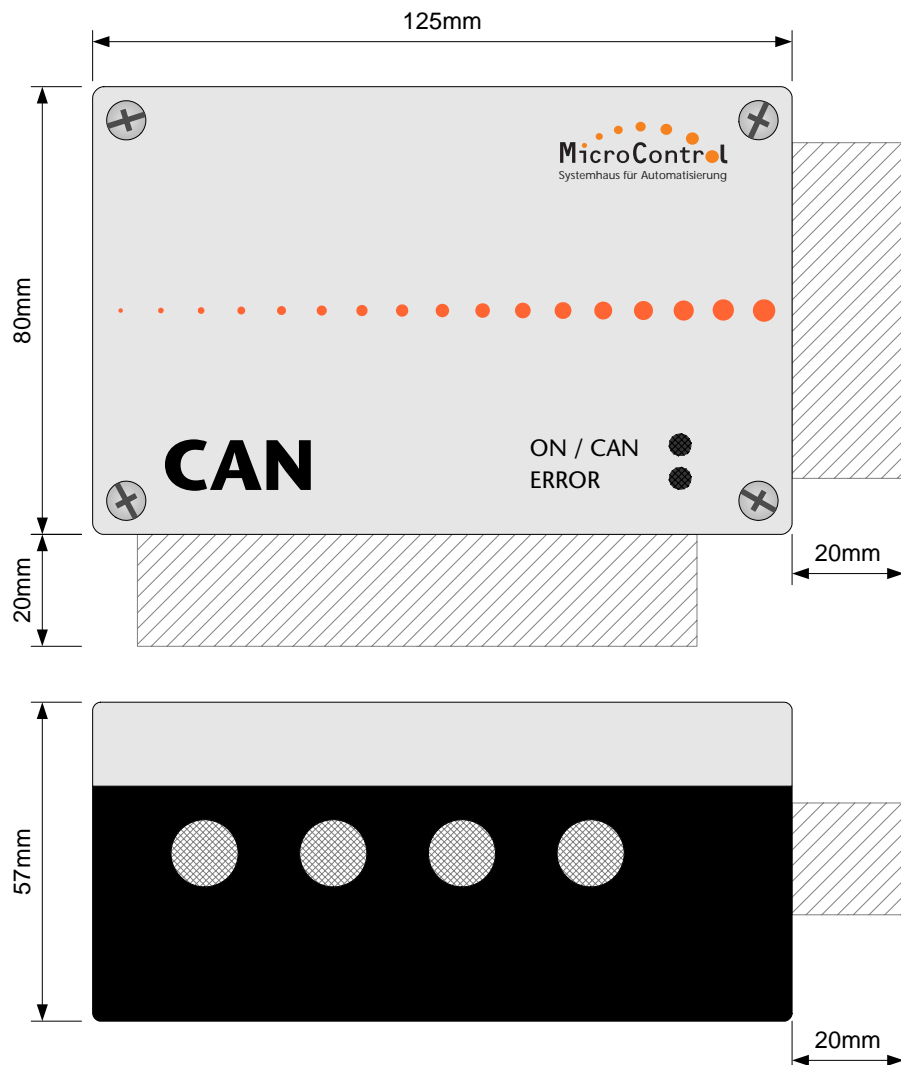


*Fig. 4: Case dimensions*

3

## 4. Assembly and Disassembly


## 4.1 Safety Regulations

This paragraph gives important information about the conditions of use. It was written for personnel which is qualified and trained on electrical devices.

Qualified and trained personnel are persons who fulfill at least one of the following conditions:

● You know the safety regulations for automated machines and you are familiar with the machine.

● You are the operator for the machine and you have been trained on operation modes. You are familiar with the operation of devices described in this manual.

● You are responsible for setting into operation or service and you are trained on repairing automated machines. In addition you are trained in setting electrcal devices into operation, to connect the earthing conductor and to label these devices.

Terms of Use

The devices described in this manual can only be used for the mentioned applications. Other devices used in conjuction have to meet the safety regulations and EMI requirements.

To ensure a trouble free and safe operation of the device please take care of proper transport, appropriate storage, proper assembly as well as careful operation and maintenance.

## 4.2 General Information

Assembly

The µCAN module should be assembled on an at least 2 mm thick mounting plate or direct in the plant. The module is fixed with 2 screws of type M4, which are plugged into the bottom part of the case. You find an assembly template in the appendix of this manual.

Power Supply

The µCAN module requires a two core cable for power supply. The cable is inserted from the right side into the case, where the terminals for power supply are located. However it makes sense to use a four core cable in order to run the CAN bus over the same cable.

The non-fused earthed conductor is connected at the terminal outside the case (refer to Fig. 5, "Connection of earthed conductor"). The non-fused earthed conductor may not lead inside the case because of EMI.

**STOP**    The non-fused earthed conductor may not lead inside the µCAN case and may not be connected to a terminal inside the case.
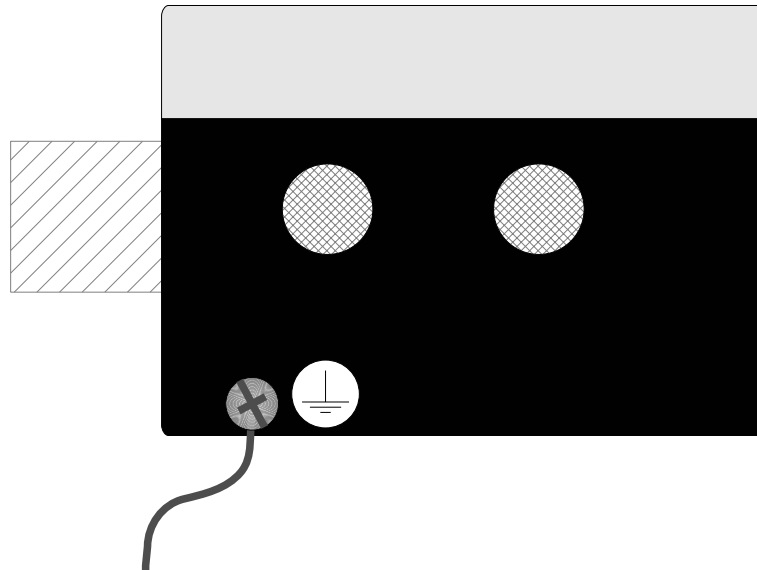


*Fig. 5: Connection of earthed conductor*

**STOP**    Operation of the µCAN module is only permitted with closed case.

## 4.3 Assembly

Assembly is performed with help of the template attached to this manual. With the template all necessary bore-holes for screws of type M4 can easily be drilled. If the module is directly fixed to the machine make sure to take the proper drill size for tapping.
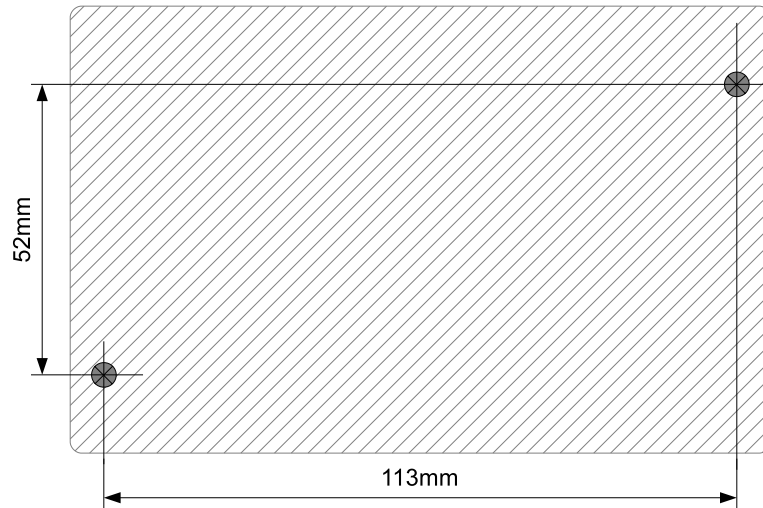


*Fig. 6: Assembly pattern for the module*

When assembling several modules at the same location please make sure to leave some area for the PG screws.

For a quick identification of the modules during operation you may use an adhesive label / tag on top of the module. Please write down the node ID that is set for the module.

Please make sure that the first node and the last node in the CAN network are terminated with a resistor (refer to "Termination" on page 25).

## 4.4 Disassembly

Please make sure to disconnect the power supply from the device first!

Open the cover of the module and remove all signal lines first. Next remove the cables for CAN bus and power supply from the terminals.

For a safe transport remove the PG screws and close the cover again.

**4**

## 4.4 Disassembly

## 5. Installation

## 5.1 Potential Basics

The potential environment of a system that is realized with a µCAN.4.ao-BOX module is characterized by following features:

- The CAN bus potential is isolated from the power supply.

- The electronic of the µCAN.4.ao-BOX module is not isolated from the power supply.

- All /O signal lines are not isolated among each other.

- All I/O signals are optically isolated from the CAN bus potential.

**5**

## 5.2 EMC Considerations

EMC (Electromagnetic Compatibility) is the ability of a device to work in a given electromagnetic environment without influencing this environment in a not admissible way.

All µCAN modules fit these requirements and are tested for electromagnetic compatibility in a EMC laboratory. However a EMC plan should be done for the system in order to exclude potential noise sources.

Noise signals can couple in different ways. Depending on that way (guided wave propagation or non-guided wave propagation) and the distance to the noise source the kinds of coupling are differentiated.

### DC Coupling

If two electronic circuits use the same conductor we speak of a DC coupling. Noise sources are in that case: starting motors, frequency converters (switching devices in general) and different potentials of cases or of the common power supply.

### Inductance Coupling

An inductance coupling is given between two current-carrying conductors. The current in a conductor will cause a magnetic field which induces a voltage in the second conductor (transformer principle). Typical noise sources are transformer, power cables and RF signal cables.

### Capacitive Coupling

A capacitive coupling is given between two conductors which have a different potential (principle of a capacitor). Noise sources are in that case: parallel running conductors, static discharge and contactors.

### RF Coupling

A RF coupling is given when electromagnetic fields hit a conductor. This conductor works like an antenna for the electromagnetic field and couples the noise into the system. Typical noise sources are spark plugs and electric motors. Also a radio set might be a noise source.

To reduce the impact of noise sources please take care to follow the basic EMC rules.

### 5.2.1 Grounding

All inactive metal plates must be grounded with low impedance. This method ensures that all elements of the system will have the same potential.

Please take care that the ground potential never carries a dangerous voltage. The grounding must be connected to the safety earth.

**STOP** The µCAN modules are grounded by the contact which is located under one of the PG screws (see fig. 5, "Connection of earthed conductor"). Additional contacts can be mounted under the PG screws for shielding purposes on demand. The ground potential may not be connected to a terminal inside the case.

**5**

## 5.2.2 Shielding of cables

If noise is coupled to a cable shield it is grounded to safety earth via the metal cover. The cable shields have to be connected to the safety earth with low impedance.

### *Cable type*

For installation of the µCAN module you should only use cable with a shield that covers at least 80% of the core. Do not use cable with a shield made from metallized foil because it can be damaged very easy and has not a good shielding.

### *Cable connection*

In general the cable shield should be grounded on both ends. The cable shield should only be grounded on one end if an attenuation is necessary in the low frequency range. The cable shield can not be grounded on both ends for temperature sensors. The grounding on one end of the cable is necessary if

- there is no contact to the safety earth possible,
- analogue signals with only a few mV or mA are transmitted (e.g. temperature sensors).

The shield of the CAN bus cable may not lead inside the housing of the µCAN.4.ao-BOX. Never connect the shield to the terminals inside the device.

### 5.2.3 CAN Cable

The CAN cable must meet the requirements of ISO11898. The cable must meet the following specifications:

| Parameter | Value |
|---|---|
| Impedance | 108 - 132 Ohm (nom. 120 Ohm) |
| Specific Resistance | 70 mOhm/Meter |
| Specific Signal Delay | 5 ns/Meter |

*Tabelle 2: Specifications of CAN bus cable*

The CAN bus cable is connected to the µCAN.4.ao-BOX module via terminals inside the case. For the pinning of the terminals refer to "CAN Bus" on page 21 of this manual.

**STOP** Do not confuse the signal lines of the CAN bus, otherwise communication between the modules is impossible.

**5**

## 5.3 Power Supply

The µCAN.4.ao-BOX module is designed for industrial applications. By means of a DC/DC converter the CAN bus of the module is isolated from the supply voltage. Also the analogoue outputs are isolated against the power supply. The supply voltage must be within the range from 12 V DC to 60 V DC. The input is protected against confusing the poles.

Please make sure not to confuse the poles when connecting the power supply. The positive supply is connected to the terminal **V+**. There are 2 terminals for the possitive supply, which are internally connected.

The negative supply is connected to the terminal **GND**. There are 2 terminals for the negative supply, which are internally connected.
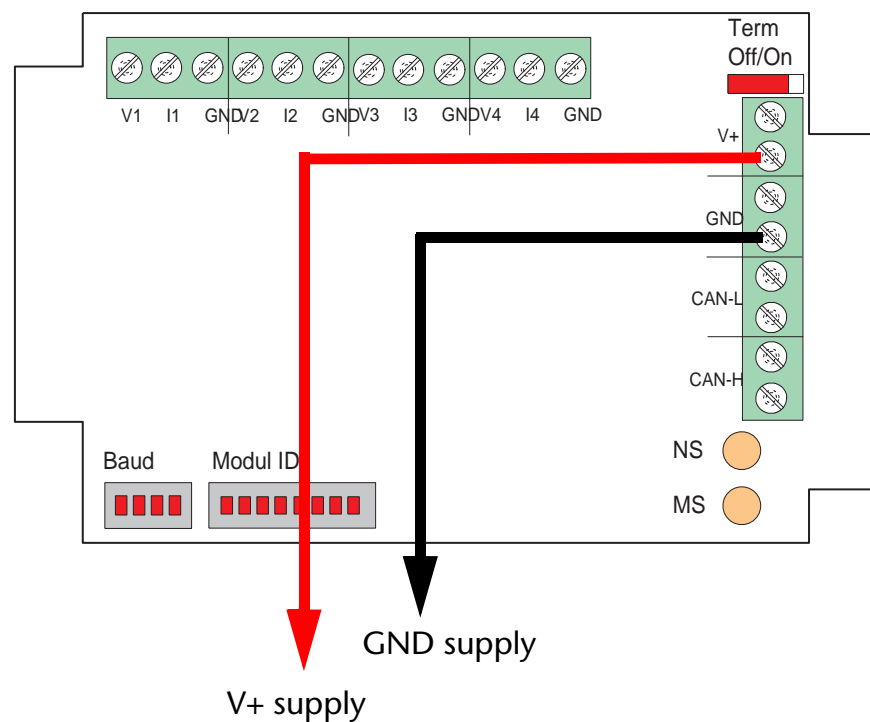


*Abb. 7: Connection of power supply*

The maximum supply voltage for the electronic is **60V** DC. Higher voltages will destroy the electronic.

## 5.4 CAN Bus

The two wires of the CAN bus are connected to the corresponding terminals.

To reduce the influence of EMI please take care that the CAN bus cable does not cross the wires of the signal lines.

The CAN bus line with positive potential must be connected to the terminal **CAN_H**. The CAN bus line with negative potential must be connected to the terminal **CAN_L**.
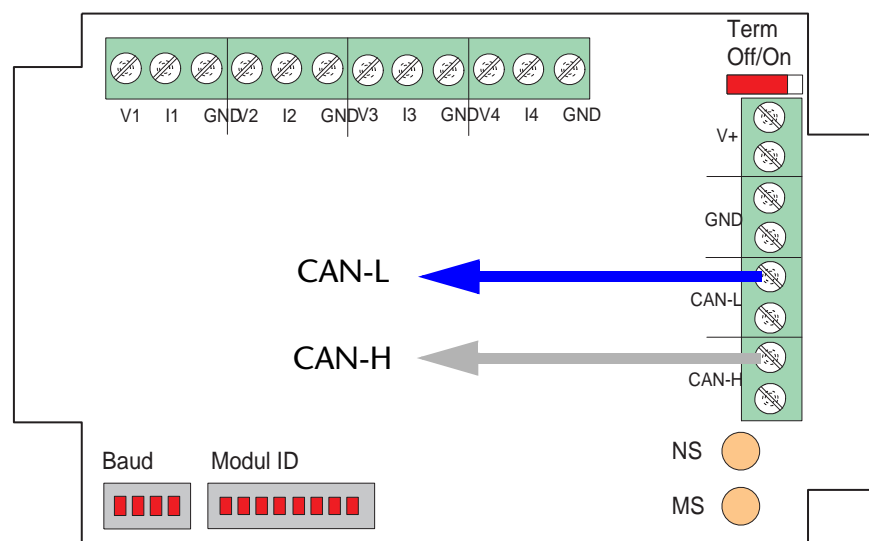
*Abb. 8: Connection of CAN bus*

Confusing the poles of the CAN bus lines will lead to a communication error on the complete network. The shield of the CAN bus cable may not lead into the housing and may not be connected to a terminal inside the housing. Cable shields have to be connected to the terminals outside the housing.

If you use a Sub-D connector with 9 pins (according to CiA standard), the conductor **CAN_H** is connected to pin 7 and the conductor **CAN_L** is connected to pin 2.

## 5.5 Address Selection

Address selection of the µCAN.4.ao-BOX module is done via an 8-pin DIP-switch, marked "Modul-ID" which is located at the lower left corner of the PCB. Selection of the address may be done with a small screw driver.

Modul ID
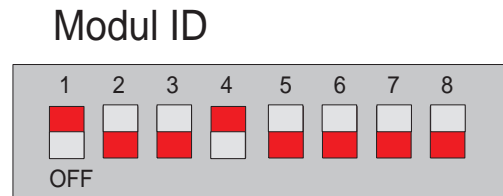


*Abb. 9: Setup of module address (here address 9 is shown)*

The 8-pin DIP-switch sets the binary code for the module address. The first pin of the switch (marked with '1') represents bit 0 of a byte. The last pin of the switch (marked with '8') represents bit 7 of a byte.

Valid module addresses are within the range from 1..127, resp. 01h..7Fh. Each node within a CANopen network must have a unique module address (node ID). Two nodes with the same node ID are not allowed.

The selected address is read during initialization of the module, after Power-on or Reset. The module runs with the selected node ID until a new node ID is selected and a Reset is performed (via the CAN bus) or the power supply is switched off

Switch 8 must always be in OFF position. Do not put all switches in the OFF position. In these configurations the module will not start to communicate on the bus.

## 5.6 Baudrate

Baudrate selection of the µCAN.4.ao-BOX module is done via a 4-pin DIP-switch, marked "Baud" which is located at the lower left corner of the PCB. Selection of the baudrate may be done with a small screw driver.
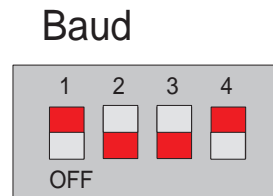
Baud

*Abb. 10: Setup of baudrate (drawing shows 1 MBit/s)*

The 4-pin DIP-switch sets the binary code for the module baudrate. The first pin of the switch (marked with '1') represents bit 0 of a byte. The last pin of the switch (marked with '4') represents bit 3 of a byte.

The supported baudrates of the µCAN.4.ao-BOX module are given in the following table. The values are recommended by the CiA.

| Baudrate | DIP-switch position | | | |
|---|---|---|---|---|
| | **1** | **2** | **3** | **4** |
| Autobaud / LSS[a] | 0 | 0 | 0 | 0 |
| Autobaud | 1 | 0 | 0 | 0 |
| 20 kBit/s | 0 | 1 | 0 | 0 |
| 50 kBit/s | 1 | 1 | 0 | 0 |
| 100 kBit/s | 0 | 0 | 1 | 0 |
| 125 kBit/s | 1 | 0 | 1 | 0 |
| 250 Kbit/s | 0 | 1 | 1 | 0 |
| 500 kBit/s | 1 | 1 | 1 | 0 |
| 800 kBit/s | 0 | 0 | 0 | 1 |
| 1 MBit/s | 1 | 0 | 0 | 1 |

*Tabelle 3: Einstellung der Baudrate*

a.LSS will be used only if all Adress-Switches are in the "OFF"-Position and the Baudrate-Switches are in the "OFF"-Position

The baudrate 10 kBit/s is not supported with the μCAN.4.ao-BOX module. In the position **Autobaud** an automatic detection of the baudrate on the CAN bus is started.

**5**

## 5.7 Termination

The modules at both ends in the CAN network have to be terminated with a resistor of 120 ohms. That means the modules at the end of the bus line are not reflecting back power and the communication can not be disturbed.

For termination of the µCAN.4.ao-BOX the "**Term**" switch must be turned from position "Term Off" to position "Term On".

Please make sure that only the devices at both ends of a CAN bus are terminated. In un-powered condition the correct termination value is 60 Ohm between the lines CAN-H and CAN-L.
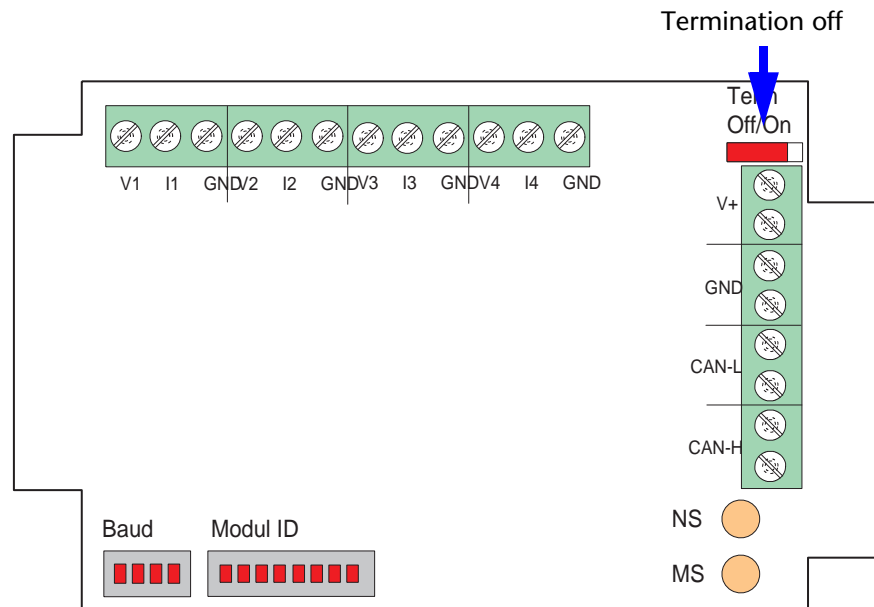


*Abb. 11: Termination of CAN bus*

Figure 10 shows the termination of the module in "off" position which is the factory default.

**5**

# 6. Analogue Output Signals

The µCAN.4.ao-BOX has four analogue output signals. The terminals are labeled with "**V1 I1 GND**" to "**V4 I4 GND**" from left to right on the PCB.

Please keep the basics of EMI rules in mind when planning the wiring. Only proper wiring and EMI precautions make sure that the module runs without trouble.

## 6.1 Function principle

Configuration of each terminal (voltage or current output) is done via configuration over the CANbus.
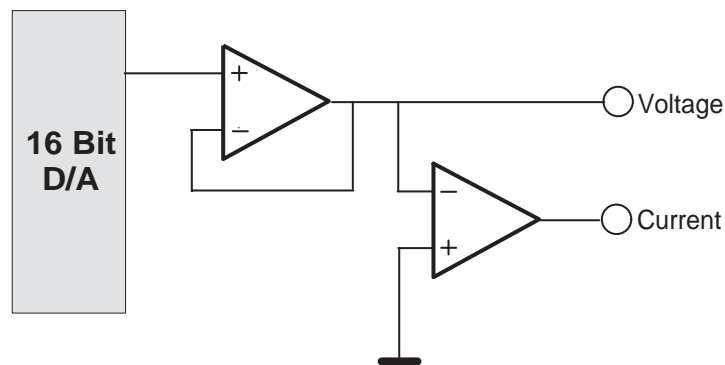
*Abb. 12: Schematic of the analogue output terminal*

The internal driver architecture consist of an 16-bit digital to analogue converter. By means of additional OPVs the signals are then transformed into the voltage or current output signals.

| Parameter | Value | Driver Impedance |
|-----------|-------|------------------|
| $V_{out}$ | -10 .. +10 V | min. 500 Ohm |
| $I_{out}$ | 0 .. 20 mA | max. 500 Ohm |

*Tabelle 4: Electrical Parameters*

## 6.2 Pin Assignment

The terminal block of the µCAN.4.ao-BOX is designed to connect 4 analogue output signal lines. The terminal blocks marked with "**V1**" to "**V4**" will source an output voltage ( -10V .. +10V ) and the terminal blocks marked with "**I1**" to "**I4**" will source an output current ( 0 .. 20mA ). Each output has a common ground signal which can be connected to the "**GND**" terminal block.
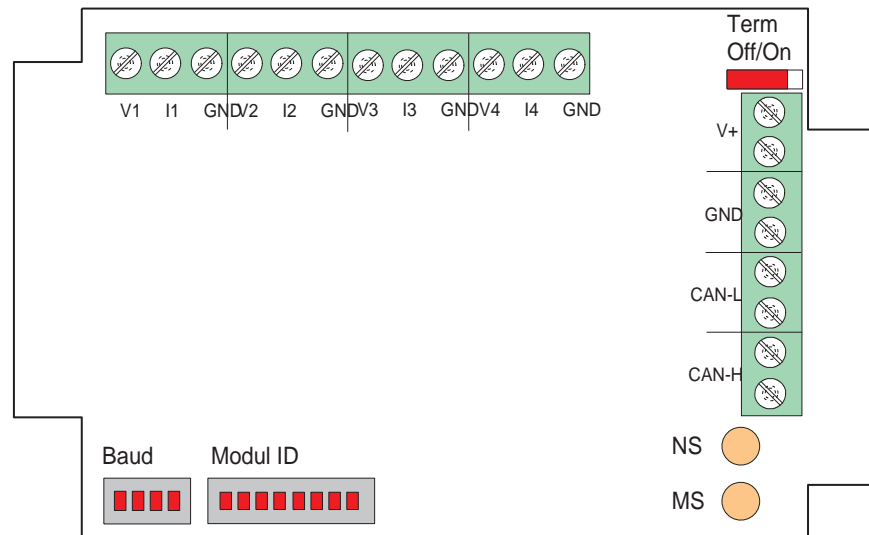


*Abb. 13: Terminals for signal lines*

| Pin | Function |
|-----|----------|
| V1 | Output 1: voltage |
| I1 | Output 1: current |
| GND | Common GND of output 1 |
| .... | .... |
| V4 | Output 4: voltage |
| I4 | Output 4: current |
| GND | Common GND of output 4 |

*Tabelle 5: Pin assignment*

Make sure to connect signal lines only in powered off state in order to prevent a damage of the electronic.

# 7. Diagnosis

All modules of the µCAN family have LEDs to display the operating state and to signalize an error state.

The µCAN.4.ao-BOX has two bi-color LEDs (green/red) labeled with "NS" (Network Status) and "MS" (Module Status) on the PCB.

On the case cover the LEDs are marked as **ON/CAN** for the network status and **ERROR** for the module status.
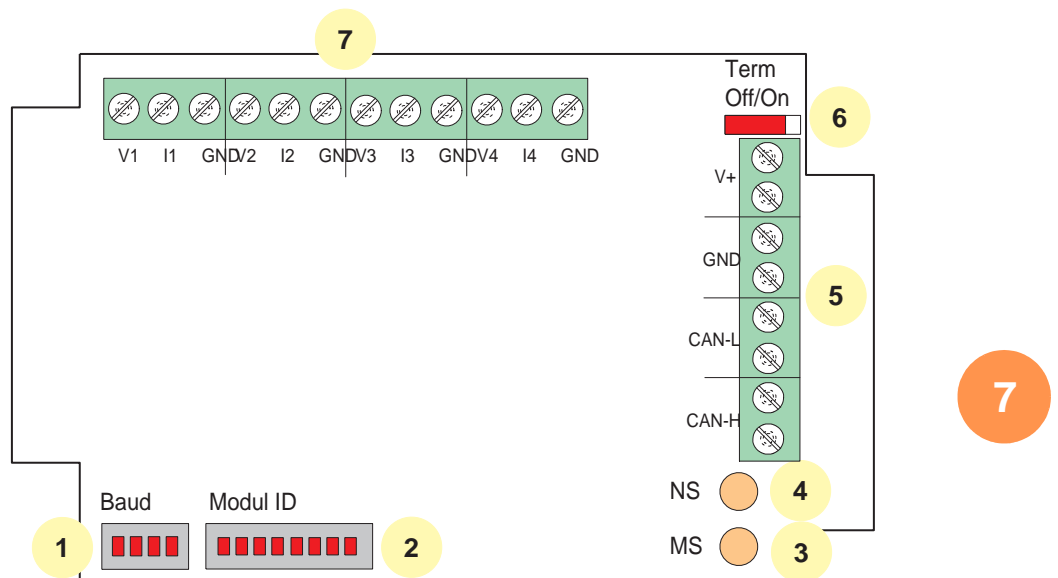


*Abb. 14: Position of LEDs on the module*

In normal operation all LEDs should have a green or orange color. A red steady light or a red blinking of a LED indicates an error condition.

## 7.1 Network Status

The bi-color LED labeled with "NS" (on the case cover marked as ON/CAN) shows the status of the CANopen state machine as well as the error state of the CAN controller.

## 7.1.1 Representation of NMT state machine

The green light of the NS-LED represents the status of the CANopen network management state machine.

Initialisation (Autobaud detection)

NMT state: device is "stopped"
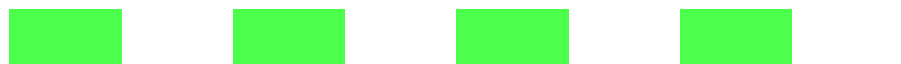
NMT state: device is "pre-operational"

NMT statue: device is "operational"

### 7.1.2 Representation of CAN controller state

The red light of the NS-LED represents the error state of the CAN controller. The red light is off during error-free condition.

CAN status: controller in "warning" state

CAN status: controller in "error passive" state

CAN status: controller in "bus-off" state
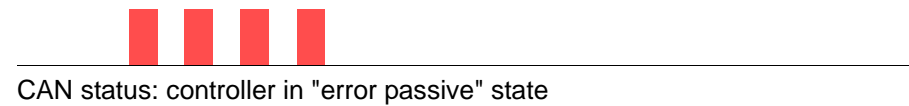
### 7.1.3 Combined representation

The combination of the green light and the red light of the NS-LED allows the representation of both - the NMT status and the CAN controller status. The following pictures give an example of combined representation.

Device in "pre-operational" state, CAN controller in "warning" state

Device in "operational" state, controller in "error passive" state

## 7.2 Module Status

The bi-color LED labeled with "MS" (on the case cover marked as ERROR) shows the status of the device hardware.

Module status: function/power OK

Module status: wrong baudrate setup

Module status: wrong address setup

## 8. CANopen Protocol

This chapter provides detailed information on how to connect the modules of the µCAN-series to a CANopen manager. A CANopen manager might be a PLC, a PC with a CAN interface or any other CAN device with NMT functionality.

For more information about CANopen manager please refer to the supplied manuals of your CANopen master device.

This documentation provides the actual implemented functions and services of the µCAN.4.ao-BOX.

8

## 8.1 Introduction

The identifiers of the µCAN.4.ao-BOX are set up according to the **Pre-defined Connection Set**, which is described in detail in the CANopen communication profile DS-301. The following table gives an overview of the supported services.

| Object | COB-ID (dec.) | COB-ID (hex) |
|---|---|---|
| Network Management | 0 | 0x000 |
| SYNC | 128 | 0x080 |
| EMERGENCY | 129 - 255 | 0x081 - 0x0FF |
| PDO 1 (Receive) | 513 - 639 | 0x201 - 0x27F |
| PDO 2 (Receive) | 769 - 895 | 0x301 - 0x37F |
| SDO (Transmit) | 1409 - 1535 | 0x581 - 0x5FF |
| SDO (Receive) | 1537 - 1663 | 0x601 - 0x67F |
| Heartbeat / Boot-up | 1793 - 1919 | 0x701 - 0x77F |

*Table 6: Identifier values according to the Pre-defined Connection Set*

The direction (Transmit / Receive) has to be seen from the devices point of view.

8

## 8.2 Network Management

By means of the Network Management (**NMT**) messages the state of a CANopen node can be changed (Stopped / Pre-Operational / Operational).

Start Node
*Start Node*

| ID | DLC | B0 | B1 |
|----|-----|-----|------|
| 0 | 2 | 01h | Node |

Node = module address, 0 = all modules

By transmitting the "Start Node" command the CAN-node will be set into Operational mode. This means that the node can handle PDO-communication.

Stop Node
*Stop Node*

| ID | DLC | B0 | B1 |
|----|-----|-----|------|
| 0 | 2 | 02h | Node |

Node = module address, 0 = all modules

By transmitting the "Stop Node" command the CAN-node will be set into Stopped mode. This means that the node can not handle any services except NMT commands.

Pre-Operational
*Enter Pre-Operational*

| ID | DLC | B0 | B1 |
|----|-----|-----|------|
| 0 | 2 | 80h | Node |

Node = module address, 0 = all modules

By transmitting the „Enter Pre-Operational" command the CAN-node will be set into Pre-Operational mode. In this state the node can not handle PDO messages.

**8**

**8**

Reset Node    *Reset Node*

| ID | DLC | B0 | B1 |
|----|-----|-----|------|
| 0 | 2 | 81h | Node |

Node = module address, 0 = all modules

By transmitting the „Reset Node" command the CAN-node will issue a reset operation. After reset the node will send a "Boot-up message" (refer to "Heartbeat protocol" on page 56) and enter the Pre-operational state automatically.

**8**

## 8.3 SDO Communication

All parameters of the devices (organized in an object dictionary) are accessed via the SDO service (Service Data Object). A SDO message has the following contents:

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|    | 8 | CMD | Index | | Sub-In-dex | Data | | | |

For calculation of the SDO message identifier please refer to "Introduction" on page 34.

The "Command Byte" (**CMD**) is defined according to the following table.

| SDO client (CANopen master) | SDO server (CANopen slave) | Function |
|-----------------------------|----------------------------|----------|
| $22_h$ | $60_h$ | write, size not specified |
| $23_h$ | $60_h$ | write, size = 4 bytes |
| $27_h$ | $60_h$ | write, size = 3 bytes |
| $2B_h$ | $60_h$ | write, size = 2 bytes |
| $2F_h$ | $60_h$ | write, size = 1 byte |
| $40_h$ | $42_h$ | read, size not specified |
| $40_h$ | $43_h$ | read, size = 4 bytes |
| $40_h$ | $47_h$ | read, size = 3 bytes |
| $40_h$ | $4B_h$ | read, size = 2 bytes |
| $40_h$ | $4F_h$ | read, size = 1 byte |

*Table 7: Command byte for  für SDO Expedited Botschaft*

The byte order for the fields "**Index**" and "**Data**" is least significant byte first (Intel format).

The minimum time delay between two succeeding SDO messages must be greater than 20ms. Faster communication might lead to an unpredictible device status.

## 8.3.1 SDO Abort Protocol

The SDO abort protocol is used to signalize a fault when accessing an object. This SDO abort protocol has the following format:

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|----|-----|-----|----|----|----|----|----|----|----|
|    | 8   | 80h | Index |  | Sub-In-dex | Abort Code |  |  |  |

The identifier as well as the index and sub-index correspond to the SDO request.

The abort code may have the following values:

| Abort code | Description |
|------------|-------------|
| 0504 0001h | Client / Server command specifier not valid / unknown |
| 0601 0000h | Unsupported access to an object |
| 0601 0001h | Attempt to read a "write-only" object |
| 0601 0002h | Attempt to write a "read-only" object |
| 0602 0000h | Object does not exist in the object dictionary |
| 0609 0011h | Sub-index does not exist |

*Table 8: SDO abort codes*

**8**

## 8.4 Object Dictionary

This chapter describes the implemented objects for the module µCAN.4.ao-BOX. For additional information please refer to the CANopen communication profile DS-301 and the device profile DS-404.

EDS

The implemented objects of the module µCAN.4.ao-BOX are listed in an "Electronic Data Sheet" (EDS). The EDS file can be downloaded from the MicroControl homepage.

**8**

## 8.4.1 Communication Profile

The module µCAN.4.ao-BOX supports the following objects from the communication profile DS-301:

| Index | Name |
|-------|------|
| 1000h | Device Profile |
| 1001h | Error Register |
| 1002h | Manufacturer Status |
| 1003h | Predefined Error-Register |
| 1005h | COB-ID SYNC-Message |
| 1008h | Manufacturer Device Name |
| 1009h | Manufacturer Hardware Version |
| 100Ah | Manufacturer Software Version |
| 100Ch | Guard Time |
| 100Dh | Life Time Factor |
| 1010h | Store Parameters |
| 1011h | Restore Default Parameters |
| 1014h | COB-ID Emergency-Message |
| 1016h | Heartbeat Consumer Time |
| 1017h | Heartbeat Producer Time |
| 1018h | Identity Object |
| 1029h | Error Behaviour |
| 1400h | 1$^{st}$ Receive PDO Parameters |
| 1600h | 1$^{st}$ Receive PDO Mapping Parameters |
| 1F80h | NMT Startup |

*Table 9: Supported objects of the CANopen communication profile*

**8**

### Device Profile

Index 1000h    The object at index 1000h describes the type of device and its functionality.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|---|---|---|---|---|
| 0 | Unsigned32 | ro | Device Profile | 0008 0194h |

The object is read-only. Only sub-index 0 is supported. An access to other sub-indices will lead to an error message.

*Example:* read parameter, module ID = 2, index = 1000h

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|---|---|---|---|---|---|---|---|---|---|
| 602h | 8 | 40h | 00h | 10h | 00h | 00h | 00h | 00h | 00h |

As response the µCAN.4.ao-BOX will send:

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|---|---|---|---|---|---|---|---|---|---|
| 582h | 8 | 42 | 00 | 10h | 00 | 91h | 04h | 08h | 00 |

Byte 5 + Byte 6 = 0194h = $404_d$ (Device Profile Number)
Byte 7 + Byte 8 = 0008h = 8 (Additional Information)

### Error Register

Index 1001h    The object at index 1001h is an error register for the device.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|---|---|---|---|---|
| 0 | Unsigned8 | ro | Error Register | 00h |

The object is read-only. Only sub-index 0 is supported. An access to other sub-indices will lead to an error message.

*Example:* read parameter, module ID = 2, Index = 1001h

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|---|---|---|---|---|---|---|---|---|---|
| 602h | 8 | 40h | 01h | 10h | 00 | 00 | 00 | 00 | 00 |

**8**

As response the module will return its error register value. The following error types are supported:

Generic Error

Bit 0 is set to '1'. The generic error is set due to hardware faults.

Communication Error

Bit 4 is set to '1'. The communication error is set due to faults on the CAN bus.

The object is read-only. Only sub-index 0 is supported. An access to other sub-indices will lead to an error message.

*Pre-defined Error Field*

Index 1003

The object at index 1003h holds the errors that have occured on the device. The object stores a maximum of 10 error conditions.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned8 | rw | Number of errors | 00h |
| 1 .. 10 | Unsigned32 | ro | Standard error field | 0000 0000h |

The object supports the sub-indices 0 to 10. An access to other sub-indices will lead to an error message. Writing to sub-index 0 will clear the error history.

*Example:* read parameter, module ID = 2, Index = 1003h

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 602h | 8 | 40h | 03h | 10h | 05h | 00h | 00h | 00h | 00h |

As response the module will return the error value at position 5 in the history.

**8**

*Manufacturer Device Name*

Index 1008    The object at index 1008h contains the manufacturer device name.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|---|---|---|---|---|
| 0 | Visible String | ro | Device name | mCAN.4.ao-BOX |

The object is read-only. Only sub-index 0 is supported. An access to other sub-indices will lead to an error message.

*Manufacturer Hardware Version*

Index 1009h    The object at index 1009h contains the manufacturer hardware version.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|---|---|---|---|---|
| 0 | Visible String | ro | Hardware version | - |

The object is read-only. Only sub-index 0 is supported. An access to other sub-indices will lead to an error message.

*Manufaturer Software Version*

Index 100Ah    The object at index 100Ah contains the manufacturer software version.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|---|---|---|---|---|
| 0 | Visible String | ro | Software version | - |

The object is read-only. Only sub-index 0 is supported. An access to other sub-indices will lead to an error message.

*Store Parameters*

Index 1010h          The object at index 1010h supports the saving of parameters in
                     a non volatile memory.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned8 | ro | Number of objects | 04h |
| 1 | Unsigned32 | rw | Save all parameters | 0000 0001h |
| 2 | Unsigned32 | rw | Save communication | 0000 0001h |
| 3 | Unsigned32 | rw | Save application | 0000 0001h |
| 4 | Unsigned32 | rw | Save manufacturer | 0000 0001h |

In order to avoid storage of parameters by mistake, storage is
only executed when a specific signature is written to the appro-
priate sub-index. The signature is "save".

*Example:* save all parameters, module ID = 2, index = 1010h

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 602h | 8 | 23h | 10h | 10h | 01h | 73h | 61h | 76h | 65h |

As response the µCAN.4.ao-BOX will send:

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 582h | 8 | 60h | 10h | 10h | 01h | 00h | 00h | 00h | 00h |

Parameters are stored in a non-volatile memory after reception of
the store request message.

**8**

### *Restore Default Parameters*

Index 1011h    The object at index 1011h supports the restore operation of default parameters.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned8 | ro | Number of objects | 04h |
| 1 | Unsigned32 | rw | Restore all param. | 0000 0001h |
| 2 | Unsigned32 | rw | Restore commun. | 0000 0001h |
| 3 | Unsigned32 | rw | Restore application | 0000 0001h |
| 4 | Unsigned32 | rw | Restore manufacturer | 0000 0001h |

In order to avoid the restoring of default parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-index. The signature is "load".

*Example:* restore all parameters, module ID = 2, Index = 1011h

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|----|-----|----|----|----|----|----|----|----|----|
| 602h | 8 | 23h | 11h | 10h | 01h | 6Ch | 6Fh | 61h | 64h |

As response the µCAN.4.ao-BOX will send:

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|----|-----|----|----|----|----|----|----|----|----|
| 582h | 8 | 60h | 11h | 10h | 01h | 00h | 00h | 00h | 00h |

**8**

### *COB-ID for emergency message*

Index 1014h    The object at index 1014h defines the identifier value for the emergency message.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned32 | rw | COB-ID EMCY | 80h + Node-ID |

Only sub-index 0 is supported. An access to other sub-indices will lead to an error message. The default value for the identifier is 80h + selected node ID.

*Identity Object*

Index 1018h     The object at index 1018h holds the identity object (LSS address) of device.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|---|---|---|---|---|
| 0 | Unsigned8 | ro | Largest Sub-Index | 4 |
| 1 | Unsigned32 | ro | Vendor ID | 0000 000Eh |
| 2 | Unsigned32 | ro | Product Code | 0013 A745h |
| 3 | Unsigned32 | ro | Revision Number | 0298 00xxh |
| 4 | Unsigned32 | ro | Serial Number | - |

The object is read-only. Only sub-indices 0 to 4 are supported. An access to other sub-indices will lead to an error message.

Vendor ID     The "Vendor ID" contains a unique value allocated to each manufacturer. The numbers are managed by the CAN in Automation

Product Code     The "Product Code" identifies a specific product, i.e. it is unique to the order code of devices from MicroControl.

Revision Number     The "Revision Number" consists of a major revision number (upper word) and a minor revision number (lower word). The major revision number identifies a specific CANopen behaviour. The minor revision number identifies different versions with the same CANopen behaviour.

Serial Number     The "Serial Number" contains the serial number of a device.

**8**

*Error behaviour*

Index 1029h   If a serious CANopen device failure is detected in NMT state Operational, the CANopen device will enter by default autonomously the NMT state Pre-operational. The object 1029h allows the device to enter alternatively the NMT state Stopped or remain in the current NMT state.

| Sub-Index | Data Type | Acc. | Name | Defaul Value |
|-----------|-----------|------|------|--------------|
| 0 | Unsigned8 | ro | number of entries | 01h |
| 1 | Unsigned8 | rw | Communication error | 00h |

The following codes are possible:

| Value | Description |
|-------|-------------|
| 00h | Change to NMT state Pre-operational |
| 01h | No change of the NMT state |
| 02h | Change to NMT state Stopped |

*Table 10: Codes for error behaviour setup*

The device detects  the following communication errors:
- Bus-off conditions of the CAN interface
- Life guarding event with the state "occurred" and the reason "time out"
- Heartbeat event with state "occurred" and the reason "time out"

**8**

### 8.4.2 Device Profile

This section describes all device profile specific objects (DS-404) of the µCAN.4.ao-BOX:

| Index | Name |
|-------|------|
| 6301h | AO Physical Unit |
| 6302h | AO Decimal Digits |
| 6310h | AO Output Type |
| 6340h | AO Fault Mode |
| 7300h | Output Process Value |
| 7330h | Output Field Value |
| 7341h | Output Fault Value |

*Table 11: Supported objects of the device profile DS-404*

**8**

*AO Physical Unit*

Index 6301h       The object at index 6301h defines the physical unit of the process value.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|---|---|---|---|---|
| 0 | Unsigned8 | ro | Largest Sub-Index | 04h |
| 1 | Unsigned16 | ro | Physical Unit Out 1 | 000Bh |
| 2 | Unsigned16 | ro | Physical Unit Out 2 | 000Bh |
| 3 | Unsigned16 | ro | Physical Unit Out 3 | 000Bh |
| 4 | Unsigned16 | ro | Physical Unit Out 4 | 000Bh |

The object allows read-only access. Only sub-indices 0 to 4 are supported. An access to other sub-indices will lead to an error message.

The following signal types are supported. Each channel can be configured individually.

| Value | Function |
|---|---|
| 0Bh | Volt |
| 15h | mA |

*Table 12: Output signal type definitions*

*AO Decimal Digits*

Index 6302h       The object at index 6302h defines the number of digits after the comma (fixed-point representation).

| Sub-Index | Data Type | Acc. | Name | Default Value |
|---|---|---|---|---|
| 0 | Unsigned8 | ro | Largest Sub-Index | 04h |
| 1 | Unsigned8 | ro | Decimal Digits Out 1 | 03h |
| 2 | Unsigned8 | ro | Decimal Digits Out 2 | 03h |
| 3 | Unsigned8 | ro | Decimal Digits Out 3 | 03h |
| 4 | Unsigned8 | ro | Decimal Digits Out 3 | 03h |

The object allows read-only access. Only sub-indices 0 to 4 are supported. An access to other sub-indices will lead to an error message. The number of decimal digits after the comma is always 3, i.e. the process value has a resolution of 0,001 V or mA.

*Analogue Output Signal Type*

Index 6310h          The object at index 6310h defines the analogue output signal type.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned8 | ro | Largest Sub-Index | 04h |
| 1 | Unsigned16 | rw | Signal Type Output 1 | 000Bh |
| 2 | Unsigned16 | rw | Signal Type Output 2 | 000Bh |
| 3 | Unsigned16 | rw | Signal Type Output 3 | 000Bh |
| 4 | Unsigned16 | rw | Signal Type Output 4 | 000Bh |

The object allows read write access. Only sub-indices 0 to 4 are supported. An access to other sub-indices will lead to an error message.

The following signal types are supported. Each channel can be configured individually.

| Value | Function |
|-------|----------|
| 0Bh | Voltage Output ( +/- 10 V ) |
| 15h | Current Output ( 0..20 mA ) |

*Table 13: Output signal type definitions*

*Example:* Set output type to current, module address 1, channel 1

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 601h | 8 | 22h | 10h | 63h | 01h | 15h | 00h | 00h | 00h |

As response the µCAN.4.ao-BOX will send:

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 581h | 8 | 60h | 10h | 63h | 01h | 00h | 00h | 00h | 00h |

*Analogue Output Fault Mode*

Index 6340h    The object at index 6340h defines the analogue output signal in case of an error. The signal can be defined in the AO Output Faulr Value Object ( 7341h ). An error can be a Heart-Beat Timeout, Node-Guard Event or Bus-Off situation.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned8 | ro | Largest Sub-Index | 04h |
| 1 | Unsigned8 | rw | Signal Type Output 1 | 00 |
| 2 | Unsigned8 | rw | Signal Type Output 2 | 00 |
| 3 | Unsigned8 | rw | Signal Type Output 3 | 00 |
| 4 | Unsigned8 | rw | Signal Type Output 4 | 00 |

The object allows read write access. Only sub-indices 0 to 4 are supported. An access to other sub-indices will lead to an error message.

The following fault types are supported. Each channel can be configured individually.

| Value | Function |
|-------|----------|
| 00h | Actual output remains unchanged |
| 01h | Output will be changed to value according to settings in Object 7341h / Output Fault Value |

*Table 14: Output signal type definitions*

**8**

*AO Process Value*

Index 7300h

The object at index 7300h holds the 16-bit values to set up the according output individually.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned8 | ro | Largest Sub-Index | 04h |
| 1 | Unsigned16 | rw | Output 1 | 0000 |
| 2 | Unsigned16 | rw | Output 2 | 0000 |
| 3 | Unsigned16 | rw | Output 3 | 0000 |
| 4 | Unsigned16 | rw | Output 4 | 0000 |

The object allows read write access. Only sub-indices 0 to 4 are supported. An access to other sub-indices will lead to an error message.

In the following tables there are shown some output values for the object 7300h and the according sinals which can be seen on the terminal block output.

Example values for the AO signal type "**Voltage**"

| Object 7300h, Sub x | Output Signal |
|---------------------|---------------|
| 00 00h | 0,000 V |
| 03 E8h | 1,000 V |
| EC 78h | -5,000 V |
| 27 10h | 10,000V |

*Table 15: Output signal examples / Voltage*

Example values for the AO signal type "**Current**"

| Object 7300h, Sub x | Output Signal |
|---------------------|---------------|
| 00 00h | 0,000 mA |
| 03 E8h | 1,000 mA |
| 27 10h | 10,000 mA |
| 4E 20h | 20,000 mA |

*Table 16: Output signal examples / Voltage*

**8**

*AO Field Value*

Index 7330h          The object at index 7330h holds the raw digital values of the DACs for each input.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned8 | ro | Largest Sub-Index | 04h |
| 1 | Unsigned32 | ro | DAC Value 1 | - |
| 2 | Unsigned32 | ro | DAC Value 2 | - |
| 3 | Unsigned32 | ro | DAC Value 3 | - |
| 4 | Unsigned32 | ro | DAC Value 4 | - |

The object allows read-only access. Only sub-indices 0 to 4 are supported. An access to other sub-indices will lead to an error message.
Here the digital values for the DAC output can be read for each channel. The digital values might not be always linear to the output signal due to some internal scaling / calibration parameters.

*AO Output Fault Value*

Index 7341h          The object at index 7341h holds the 16-bit values to be written to the analogue output in case of an error condition. Each output channel can be set up individually.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned8 | ro | Largest Sub-Index | 04h |
| 1 | Unsigned16 | rw | Output 1 | 0000 |
| 2 | Unsigned16 | rw | Output 2 | 0000 |
| 3 | Unsigned16 | rw | Output 3 | 0000 |
| 4 | Unsigned16 | rw | Output 4 | 0000 |

The object allows read write access. Only sub-indices 0 to 4 are supported. An access to other sub-indices will lead to an error message. The values can be written in the ranges according to object 7300.

The output value will only be set in case of an error if the object 6340h ( AO Fault Mode ) is set up properly.

## 8.5 Device monitoring

For device monitoring CANopen provides two mechanisms (protocols):
- heartbeat
- node guarding

It is recommended by the CAN in Automation **not to use** node guarding for device monitoring (CiA AN802 V1.0: CANopen statement on the use of RTR messages).

### 8.5.1 Heartbeat protocol

The heartbeat protocol is used in order to survey other CANopen nodes in the network and retrieve their network state.

heartbeat ID

The identifier for the heartbeat protocol is set to 700h + module address. The identifier can not be changed. The message repetition time (called "heartbeat producer time") is configured with object 1017h.

The heartbeat protocol transmits one byte of data, which represents the network state.

| Network State | Code (dec.) | Code (hex) |
|---|---|---|
| Bootup | 0 | 00h |
| Stopped | 4 | 04h |
| Operational | 5 | 05h |
| Pre-Operational | 127 | 7Fh |

*Table 17: Status Information for Heartbeat*

After Power-on / Reset the module will send the "Boot-up message" to signal that it finished the initialization sequence.

*Example:* Power-on of module with address 2

| ID | DLC | B0 |
|---|---|---|
| 702h | 1 | 00h |

**8**

*Consumer heartbeat time*

Index 1016h

The object at index 1016h defines the consumer heartbeat time.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|---|---|---|---|---|
| 0 | Unsigned8 | ro | Number of objects | 2 |
| 1 | Unsigned32 | rw | Heartbeat Cons. 1 | 0000 0000h |
| 2 | Unsigned32 | rw | Heartbeat Cons. 2 | 0000 0000h |

The µCAN.4.ao-BOX can monitor the presence of two other devices (heartbeat producer) in the network. If a heartbeat producer message is not received within an adjustable period, an emergency message with value 8130h (life guard error or heartbeat error) is transmitted. The 32-bit value of the object defines heartbeat time and the producers node address.

| Bit 31 ... 24 | Bit 23 ... 16 | Bit 15 ... 0 |
|---|---|---|
| reserved (00h) | producer node address | heartbeat producer time |

If the heartbeat time is 0 or the node-ID is 0 or greater than 127 the corresponding object entry is not used. The heartbeat time is given in multiples of 1 millisecond. Monitoring starts after reception of the first heartbeat.

**8**

*Producer heartbeat time*

Index 1017h   The object at index 1017h defines the cycle time of the heart-beat. The producer heartbeat time is 0 if it is not used. The time is a multiple of 1ms.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned16 | rw | Producer Time | 0000h |

The object allows read-write access. Only sub-index 0 is supported. An access to other sub-indices will lead to an error message.

*Example:* Producer time 1000 ms, module address 1

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 601h | 8 | 22h | 17h | 10h | E8h | 03h | 00h | 00h | 00h |

The answer you will receive from the module is:

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 581h | 8 | 60h | 17h | 10h | 00h | 00h | 00h | 00h | 00h |

The heartbeat producer time is not saved inside the non-volatile memory autonomously. It is necessary to store this parameter via object 1010h (refer to "Store Parameters" on page 45).

**8**

### 8.5.2 Node guarding

The NMT master polls each NMT slave at regular time intervals. This time-interval is called the guard time. The response of the NMT slave contains the NMT state of that NMT slave. The node lifetime is given by the guard time multiplied by the lifetime factor. If the NMT slave has not been polled during its lifetime, a remote node error is indicated through the NMT service life guarding event.

Upon life guard error the μCAN.4.ao-BOX will transmit an emergency message with emergency code 8130h.

#### *Guard time*

Index 100Ch
The object at index 100Ch defines the guard time. The life time factor multiplied with the guard time gives the life time for the life guarding protocol.

| Sub-Index | Data Type | Acc | Name | Default Value |
|-----------|-----------|-----|------------|---------------|
| 0 | Unsigned16 | rw | Guard time | 0000h |

The value is given in multiple of 1 millisecond. The value of 0000h disables the life guarding.

#### *Life time factor*

Index 100Dh
The object at index 100Dh defines the life time factor. The life time factor multiplied with the guard time gives the life time for the life guarding protocol.

| Sub-Index | Datentyp | Zugriff | Bedeutung | Defaultwert |
|-----------|----------|---------|------------------|-------------|
| 0 | Unsigned8 | rw | Life time factor | 00h |

The value 00h disables the life guarding.

**8**

## 8.6 PDO Communication

The real-time data transfer is performed by means of "Process Data Objects" (PDO). The transfer of PDOs is performed with no protocol overhead.

PDO communication is only possible when the device is in the network state "Operational".

### 8.6.1 Transmission Modes

*Event Driven*

Message transmission is triggered by the occurrence of an object specific event. For synchronous PDOs this is the expiration of the specified transmission period, synchronised by the reception of the SYNC object. For acyclically transmitted synchronous PDOs and asynchronous PDOs the triggering of a message transmission is a device-specific event specified in the device profile.

*Timer Driven*

Message transmission is either triggered by the occurrence of a device-specific event or if a specified time has elapsed without occurrence of an event.

The µCAN.4.ao-BOX does not support the RTR-based PDO transmission types FCh ($252_d$) and FDh ($253_d$). Bit 30 inside the COB-ID field of the PDO communication parameter record is always set to 1.

**8**

## 8.6.2 Receive-PDO

Index 1400h

The object at index 1400h defines communication parameters for the Receive-PDO.

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned8 | ro | Largest Sub-Index | 2 |
| 1 | Unsigned32 | rw | COB-ID for PDO | 200h + Node |
| 2 | Unsigned8 | rw | Transmission Type | FFh |

Only sub-indices 0 to 2 are supported. An access to other sub-indices will lead to an error message.

COB-ID for PDO

Sub-Index 1 defined the identifier for the Receive PDO. The 32-bit value has the following structure.

| Bit 31 | Bit 30 | Bit 29 | Bit 28 - 0 |
|--------|--------|--------|------------|
| PDO valid, 0 = valid 1 = not valid | RTR allowed, 0 = yes 1 = no RTR | Frame type, 0 = 11 Bit 1 = 29 Bit | Identifier, |

*Tabelle 18: Definition of COB-ID for PDO*

In order to enable the PDO the most significant bit (Bit 31) must be set to 0. In order to disable the PDO the most significant bit must be set to 1. In the default setting the PDO is active (Bit 31 = 0).

Transmission Type

The transmission type defines the transmission character of the PDO.

| Transmission Type | Description |
|-------------------|-------------|
| 00h | acyclic synchronous, |
| 01h - F0h (1 - 240 dez) | cyclic synchronous, |

*Tabelle 19: Einstellung des Transmission Type*

The Receive-PDO processes a message with 8 byte process data. The contents is copied into object 7300h (refer to "Index 7300h" on page 53) and modifies the digital outputs.

The Receive PDO has to be send with 8 Byte of data. The contents of that 8 Byte data field is copied to the object 7300, with 2 Byte each for the output channels. So by receiving the Rx-PDO the output values will change on receiving new / different data.

Example: Receiving an Rx-PDO ( Default PDO-ID / Modul address = 1 ).

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|------|-----|------|------|------|------|------|------|------|------|
| 201h | 8 | 00h | 00h | 00h | 00h | 00h | 00h | 00h | 00h |

Index 7300h

| Sub -Index | B0 | B1 |
|------|------|------|
| 01h Output 1 | 00 | 00 |

| Sub -Index | B2 | B3 |
|------|------|------|
| 02h Output 2 | 00 | 00 |

| Sub -Index | B4 | B5 |
|------|------|------|
| 03h Output 3 | 00 | 00 |

| Sub -Index | B6 | B7 |
|------|------|------|
| 04h Output 4 | 00 | 00 |
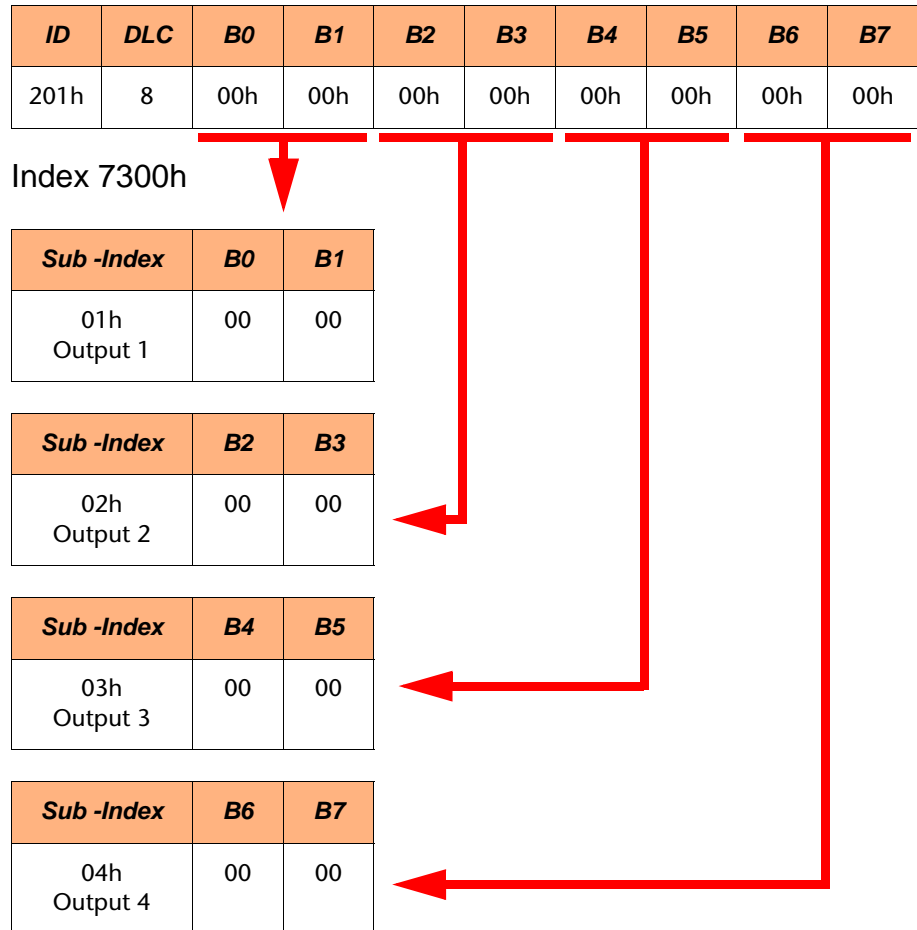
*Example:* Set up +5V on channel 2.(Modul ID: 1)

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|------|-----|------|------|------|------|------|------|------|------|
| 201h | 8 | 00h | 00h | 4Ch | 1Dh | 00h | 00h | 00h | 00h |

A communication with PDOs is only possible in Operational Mode.

### 8.6.3 Receive-PDO Mapping

Index 1600h

The object at index 1600h defines the mapping parameters for the Receive-PDO

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned8 | ro | Largest Sub-Index | 4 |
| 1 | Unsigned32 | ro | Process Value Out 1 | 73000110 |
| 2 | Unsigned32 | ro | Process Value Out 2 | 73000210 |
| 3 | Unsigned32 | ro | Process Value Out 3 | 73000310 |
| 4 | Unsigned32 | ro | Process Value Out 4 | 73000410 |

The object allows read-only access. Only sub-indices 0 to 4 are supported. An access to other sub-indices will lead to an error message.

8

### 8.6.4 Synchronisation Message

Index 1005h

The object at index 1005h defines the identifier for the SYNC-message. On reception of a message with this identifier the receipt / processing of PDOs is triggered (refer to "Receive-PDO" on page 62)..

| Sub-Index | Data Type | Acc. | Name | Default Value |
|-----------|-----------|------|------|---------------|
| 0 | Unsigned32 | rw | COB-ID SYNC | 80h |

Only sub-index 0 is supported. An access to other sub-indices will lead to an error message.

*Example:* Set SYNC-ID to 10, module address 1

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 601h | 8 | 22h | 05h | 10h | 0Ah | 00h | 00h | 00h | 00h |

As answer you will get the following message:

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 581h | 8 | 60h | 05h | 10h | 00h | 00h | 00h | 00h | 00h |

The default identifier is 80h in order to ensure a high priority of the SYNC-message.

The SYNC-identifier is not saved inside the non-volatile memory autonomously. It is necessary to store this parameter via object 1010h (refer to "Store Parameters" on page 45)

**8**

## 8.7 Emergency Message

Emergency objects are triggered by the occurrence of a device internal error situation and are transmitted from an emergency producer on the device.

An emergency is different from a SDO error message. The last one only holds the access error to the object dictionary, whereas an emergency indicates a severe hardware/software failure.

The emergency identifier has the default value $128_d$ + module-address. The emergency message has the following structure:

| ID | DLC | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|----|-----|----|----|----|----|----|----|----|----|
|    | 8   | Error Code | | ER | Manufacturer Specific Error Field | | | | |

The following emergency error codes are supported:

| Error Code | Description |
|------------|-------------|
| 0000h | Error reset or no error |
| 1000h | generic error |
| 5000h | module hardware |
| 6000h | module software |
| 8100h | CAN controller entered "warning" state |
| 8110h | CAN controller overrun |
| 8120h | CAN controller entered "error passive" state |
| 8130h | heartbeat event / node guarding event |
| 8140h | device recovered bus-off |
| 8150h | identifier collision (Tx-ID reception) |

*Table 20: Emergency error codes*

# 9. Technical Data

| Power Supply | |
|---|---|
| Supply Voltage, $U_{PWR}$ | 12 .. 60 V DC, reverse current protected |
| Power Consumption | 1,5 W (60 mA @ 24 V DC) without load |
| Isolation | Fieldbus/Supply: 500 Veff |
| Isolation | Analogue Output Block/Supply: 500 Veff |
| Physical Interface | Terminal Block (2,5 mm$^2$ ) |

| CAN-Bus | |
|---|---|
| Baudrates | 20 kBit/s .. 1 MBit/s |
| Status on the bus | active node |
| Protocol | CANopen, DS-404 |
| Physical Interface | Terminal Block (2,5 mm$^2$ ) |

| EMC | |
|---|---|
| Electromagnetic immunity | according to EN 50082-2 |
| Electrostatic discharge | 8 kV air discharge, 4 kV contact discharge, according to EN 61000-4-2 |
| Electromagnetic fields | 10 V/m, according to ENV 50204 |
| Burst | 5 kHz, 2 kV according to EN 6100-4-4 |
| Conducted RF-Disturbance | 10 V, according to EN 61000-4-6 |
| Electromagnetic emission | according to EN 50081-2 |

9

| Mechanic | |
|---|---|
| Case | Aluminium |
| Dimensions | 125 * 80 * 57 mm (L * B * H) |
| Weight | 540 g |
| Protection class | IP66 |

| Analogue Outputs / Voltage | |
|---|---|
| Impedance | min. 500 Ohm |
| Signal Span | -10V .. +10V DC |
| Protection | Short Circuit protected |
| Resolution | 16 bit |
| Accuracy (@ 23°C) | 0,02% of Span End |

| Analogue Outputs / Current | |
|---|---|
| Impedance | max. 500 Ohm |
| Signal Span | 0 .. 20 mA DC |
| Protection | Short Circuit protected |
| Resolution | 16 bit |
| Accuracy (@ 23°C) | 0,01% of Span End |

**9**

# Index

## A

Address
　range **22**
　selection **22**
Autobaud **24**

## B

Baudrate
　automatic detection **24**
　bus length **8**
　setup **23**
Bootup message **56**

## C

Cable length **8**
CAN
　cable **19**
　connection **21**
CANopen
　DS-301 **34**
　DS-404 **49**
Case dimensions **9**
Communication Profile **40**

## D

Device Profile **42**

## E

EMC **16**
EMCY
　see Emergency message
Emergency message **66**

## H

Heartbeat Protocol **56**
　Consumer **57**
　Producer **58**

## I

Identity object **47**

## M

Manufacturer Device Name **44**
Module Status
　LED **29**

## N

Network Management **35**
　Enter Pre-Operational **35**
　Reset Node **37**
　Start Node **35**
　Stop Node **35**
Network manager **7**
Network Status
　LED **29**
NMT
　see Network Management
Node Guarding **59**
Non-fused earthed conductor **12**

## O

Object
　1000h **42**
　1001h **42**
　1003h **43**
　1005h **65**
　1008h **44**
　1009h **44**
　100Ah **44**
　1010h **45**
　1011h **46**
　1014h **46**
　1017h **58**
　1018h **47**
　1029h **48**
　1400h **62**, **64**

## P

PDO
　see Process Data Object
Power supply **20**
Pre-defined Connection Set **34**
Pre-defined Error Field **43**
Process Data Object **60**
　Transmission modes **61**

## S

SYNC
　see Synchronisation Message
Synchronisation Message **65**

## T

Terminal
　CAN bus **21**
　GND **20**
　power supply **20**
Termination **25**

**Index**

MicroControl reserves the right to modify this manual and/or product described herein without further notice. Nothing in this manual, nor in any of the data sheets and other supporting documentation, shall be interpreted as conveying an express or implied warranty, representation, or guarantee regarding the suitability of the products for any particular purpose. MicroControl does not assume any liability or obligation for damages, actual or otherwise of any kind arising out of the application, use of the products or manuals.

The products described in this manual are not designed, intended, or authorized for use as components in systems intended to support or sustain life, or any other application in which failure of the product could create a situation where personal injury or death may occur.

No part of this documentation may be copied, transmitted or stored in a retrieval system or reproduced in any way including, but not limited to, photography, magnetic, optic or other recording means, without prior written permission from MicroControl GmbH & Co. KG.