MicroControl

Systemhaus für Automatisierung

µCAN.6.ai-SNAP

Manual sixfold analogue input acquisition for DIN-rail mounting
Version 1.02

Document conventions

For better handling of this manual the following icons and headlines are used:

This symbol marks paragraphs containing useful information on device operation or giving hints on configuration.

This symbol marks paragraphs which explain possible sources of-danger which might cause damage to the system or operating personnel. Read these sections carefully!

Keywords

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

MicroControl GmbH & Co. KG
Lindlaustraße 2c
D-53842 Troisdorf
Fon: +49 / 2241 / 25 65 9 - 0
Fax: +49 / 2241 / 25 65 9 - 11
http://www.microcontrol.net

1. Safety Regulations 7
1.1 General Safety Regulations 7
1.2 Safety Notice 8

2. Operation of µCAN.6.ai-SNAP 9
2.1 Overview 9

3. Project Planning 11
3.1 Functional Units of a µCAN Module 11
3.2 General Description 12
3.3 Maximum System Configuration 13
3.4 Case Dimensions 15

4. Assembly and Disassembly 17
4.1 General Information 17
4.2 Assembly 18
4.3 Disassembly 18

5. Installation 19
5.1 Potential Basics 19
5.2 EMC Considerations 20
5.2.1 Earthing of inactive Metal Parts 21
5.2.2 Shielding of Cables 21
5.3 General Information on Wiring 23
5.3.1 Groups of Wires 23
5.4 CAN Cable 25
5.5 Power Supply 26
5.6 CAN Bus 27
5.7 Module Address 28
5.8 Baud Rates 29
5.9 Termination 31

6. Analogue Inputs 33
6.1 Connection of Signal Lines 34
6.2 Monitoring of Limit Values 35

7. Diagnosis 37
7.1 Status of Network 38

7.1.1 Representation of CANopen NMT states 38
7.1.2 Representation of CAN state 39
7.1.3 Combined Representation 39

8. CANopen Protokoll 41
8.1 General Information 42
8.2 Network Management 43
8.3 SDO Communication 45
8.3.1 SDO Abort Protocol 46
8.4 Object Dictionary 47
8.4.1 Communication Profile 48
8.4.2 Manufacturer-Specific Objects 59
8.4.3 Device Profile CiA 404 63
8.5 Scaling 83
8.6 CANopen Device Monitoring 84
8.6.1 Heartbeat Protocol 85
8.6.2 Node Guarding 88
8.7 PDO Communication 89
8.7.1 Transmission Modes 89
8.7.2 Transmit PDO 1 Parameter 90
8.7.3 Transmit PDO 2 Parameter 91
8.7.4 Transmit PDO 3 Parameter 92
8.7.5 Transmit PDO 4 Parameter 93
8.7.6 Transmit PDO 1 Mapping Parameter 94
8.7.7 Transmit PDO 2 Mapping Parameter 95
8.7.8 Transmit PDO 3 Mapping Parameter 96
8.7.9 Transmit PDO 4 Mapping Parameter 97
8.7.10 Example of Transmit PDO 98
8.8 Synchronization Message 100
8.9 Emergency Message 101
8.9.1 Overview of Error Codes 102

9. Technical Data 105

Appendix 107
9.1 Drilling Template 107

9.2 Index. 107

1. Safety Regulations

Please read the following chapter in any case, as it contains important information about safe handling of electrical devices.

1.1 General Safety Regulations

This paragraph provides important information about the conditions of use. It has been 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 of the machine and you have previously been trained on operation modes. You are familiar with the operation of the devices described in this manual.
- You are responsible for setting devices into operation or servicing and you are trained on repairing automated machines. In addition, you are trained on setting electrical devices into operation, to connect the grounding conductor and to label these devices accordingly.

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

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

Please see to it that the local safety regulations are observed.

If devices are to be integrated into stationary machines without a mains switch for all phases or fuses, this equipment has to be in-

1

stalled first. The stationary machine must be connected to a grounding conductor.

If devices are supplied by mains, please see to it that the selected input voltage fits 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 such a way that the functionability of the device is not influenced by EMI.

Devices or machines for industrial automation must be constructed in such a manner to prevent any unintentional operation.

Safety precautions have to be taken by means of hardware and software in order to avoid undefined operational states of automated machines in case of a cable break.

Where automated machines can cause damage to material or personnel in case of a malfunction the system designer has to ensure that the safety precautions are met. Possible safety precautions might be the integration of a limit switch or mechanical locking device.

2. Operation of µCAN.6.ai-SNAP

2.1 Overview

The µCAN.6.ai-SNAP is the ideal solution for acquisition and linearisation of analogue standard signals ±10V, 0..20mA or 4..20mA. The signals will be transmitted via CAN bus as voltage or current values. If desired, the measured values may be scaled and its measurement units can be changed if necessary.

Abb. 1: µCAN.6.ai-SNAP module for acquisition of analogue signals

During evaluation of signals the module monitors limit values and will send a CAN message if values fall below or exceed preset limits.

Independent from the main system the µCAN modules are designed to acquire the signals on the spot which results in reduced costs for wiring.

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 to this problem is a common bus system

Our µCAN.6.ai-SNAP fully meets all these requirements. The µCAN.6.ai-SNAP runs on the standard fieldbus CAN.

Typical applications for the µCAN.6.ai-SNAP are industrial automation, transportation, food industry and environmental technology.

The µCAN.6.ai-SNAP operates with the CAN protocol

CANopen®

according to CiA 301 (version 4.02) and CiA 404. Other protocol stacks are available on request.

space-saving and compact

The µCAN.6.ai-SNAP is designed for direct use in DIN-rail mounting applications. The casing is also available with internal bus and power connector for stacking of several modules. Due to the space-saving and compact design of the casing the module can be installed virtually everywhere in the field.

inexpensive and servicefreundlich

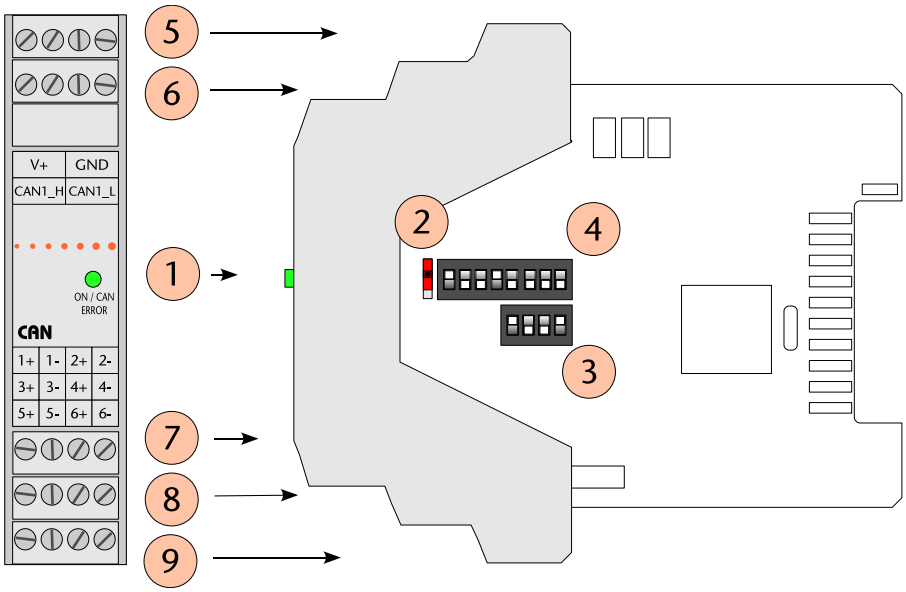
The quick and easy integration of the µCAN.6.ai-SNAP into your applications reduces development efforts and expenses. Costs for material and personnel are reduced to a minimum. The modules can be maintained or replaced easily.

3. Project Planning

The chapter Project Planning contains important information for systems engineers using the µCAN.6.ai-SNAP. This information includes case dimensions and optimum conditions of use.

3.1 Functional Units of a µCAN Module

The following figure shows the different functional units of a µCAN.6.ai-SNAP module illustrating the structure and position of its adjustment and operating facilities.



1: Bi-color LED for status indication
2: Switch for CANbus termination
3: Baudrate setting
4: Module address setting
5: Terminal block for power
6: Terminal block for CAN
7: Terminal block input (1 + 2)
8: Terminal block input (3 + 4)
9: Terminal block input (5 + 6)

Abb. 2: View of Functional Units

3.2 General Description

Our µCAN.6.ai-SNAP is a robust device for acquisition and linearisation of analogue signals for industrial purposes.
The following types of signals can be acquired:
±10 V DC,
0..20 mA,
4..20 mA
Further signal ranges are available on request.

This µCAN module acquires and linearises analogue signals which are transmitted as voltage or current values via CANbus.
If the values fall below or exceed preset limits, the user will receive an alarm message. See also chapter siehe “Überwachung der Grenzwerte” auf Seite 31.

The µCAN module is surrounded by a casing in protection class IP 20. µCAN modules can be used virtually everywhere, it can either be mounted to a machine or placed in a switching cabinet.

The µCAN module may be powered by a supply voltage of 9..36V DC and should be connected to power supply and CANbus via four core wire, thus reducing wiring effort of the µCAN.6.ai-SNAP. Adequate CANbus 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. Each µCAN module is an active node.

A maximum of 127 modules can be managed logically on one bus line. Each module must be assigned a unique address or node-ID (NID), which can be adjusted via a DIP switch at the µCAN module. The single µCAN module can be connected through the CANbus.

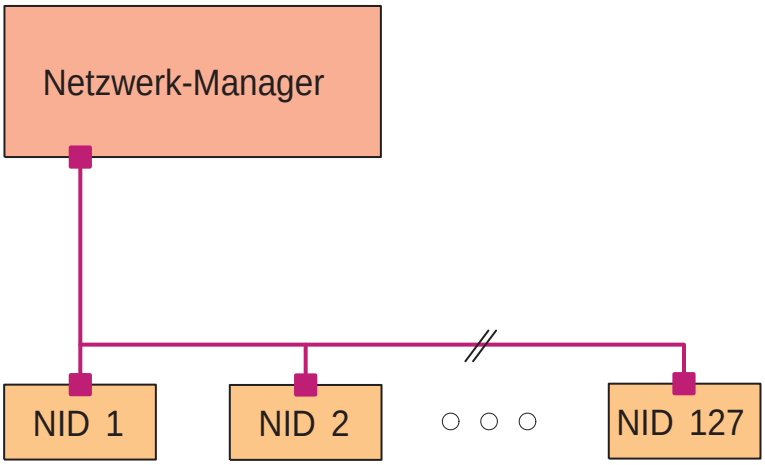


Abb. 3: Maximum system configuration

The maximum cable length depends on the selected bit rate as they are defined in the following table. The values are based on the recommendations of the CAN in Automation (CiA) and may be realized with the µCAN.6.ai-SNAP.

Baudrate	Cable lenth
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

Tabelle 1: Baud rates in relation to cable length

The CAN in Automation recommends not to use a baud rate of 100 kBit/s in new systems.

3.4 Case Dimensions

The case dimensions of the µCAN.6.ai-SNAP are given in the drawings below. Due to its casing in protection class IP 20 the µCAN module can be installed virtually everywhere. The µCAN modules can be installed either directly to the machine or in a control cabinet.

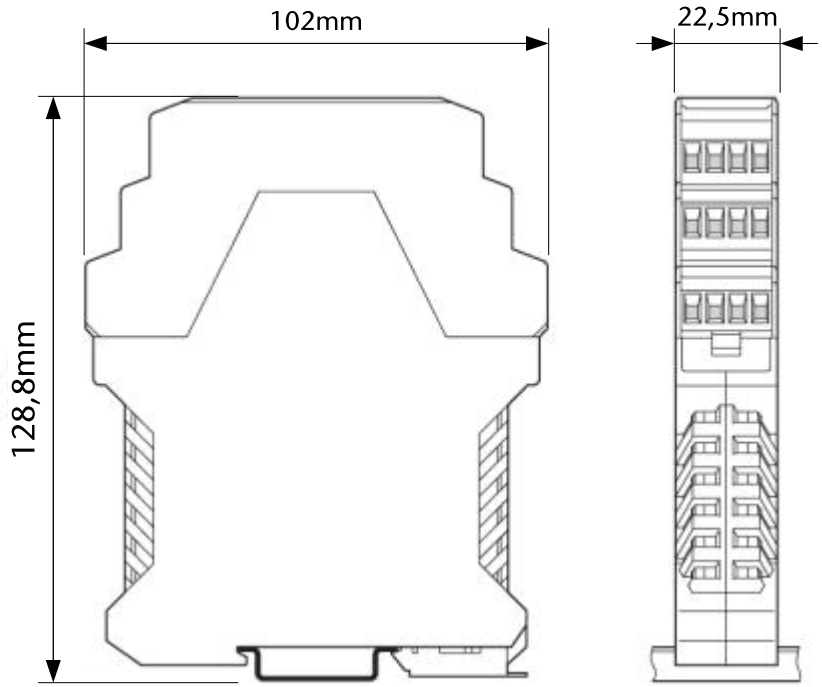


Abb. 4: Case dimensions

For detailed information on the environmental conditions for the installation of the module please see chapter “Technische Daten” auf Seite 89.

3

4. Assembly and Disassembly

4.1 General Information

Assembly

The µCAN.6.ai-SNAP modules can be mounted onto a standard DIN rail TS35. They are equipped with a snap lock system and can easily be fixed to the rail without tools.

Power supply

Power can be supplied via a two-wire cable which is connected to the respective terminals. However, application of four wire cables is more convenient as the CANbus can use the same connection .

The PE is supplied by an integrated functional earthing conductor which is situated in the back of the casing. Snapping the module onto the rail simultaneously connects the contact to the rail. PE supply within the casing is not allowed due to EMC regulations.

The PE protective conductor must not lead into the casing or connected to one of the terminals.

Abb. 5: Supply of the PE protective conductor

When operating the µCAN.6.ai-SNAP the casing must be closed.

4.2 Assembly

To facilitate identification of the µCAN modules during operation the casings should be labelled on the lid after assembly. We recommend using the set module ID for labelling.

When mounting several µCAN modules on a bus line, please make sure that the last module installed to the bus is terminated with a resistor. For more detailed information on termination of bus lines, please refer to “Termination” on page 31.

4.3 Disassembly

4

Please make sure that the device is disconnected from power supply first!

Disconnect all signal wires from the connectors. Then, disconnect the CANbus and the power supply line from the connector.

5. Installation

5.1 Potential Basics

The potential environment of our µCAN.6.ai-SNAP modules is characterized by the following features:

- The CAN bus potential is not isolated from the power supply. (Optionally available: galvanic insulation of the CAN bus)
- The individual µCAN.6.ai-SNAP modules are not isolated from the power supply. µCAN.6.ai-SNAP.
- All µCAN modules can be supplied separately.
- The I/O signals are not isolated from each other.

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 an inadmissible way.

All µCAN modules fit these requirements and are tested for electromagnetic compatibility in an EMC laboratory. However, an EMC plan for the system should be set up 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 of the noise source the following kinds of coupling can be distinguished:

DC Coupling:

If two electronic circuits use the same conductor this is called a DC coupling. Potential noise sources in these cases may be: starting motors, frequency converters (switching devices in general) and different potentials of casings and/or common power supply.

Inductance Coupling:

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

Capacitive Coupling:

A capacitive coupling is given between two conductors which have a different potential (principle of a capacitor). Potential noise sources in these cases may be: parallel running conductors, static discharge and contactors.

RF Coupling:

RF coupling is given when electromagnetic fields hit a conductor. This conductor acts as an antenna for the electromagnetic field and induces noise to the system. Typical noise sources are spark plugs and electric motors. Radio sets situated near the system may also cause interference.

To reduce the impact of noise sources please ensure that the basic EMC rules are observerd.

5.2.1 Earthing of inactive Metal Parts

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

The ground potential must not carry any dangerous voltage and must be connected to a protective earthing conductor.

The µCAN modules are equipped with an integrated functional earthing conductor which is situated in the back of the casing. Snapping the module onto the rail simultaneously connects the conductor to the rail. The protective conductor must not lead into the casing.

5.2.2 Shielding of Cables

Any noise signal which works on a cable shield will be grounded to earth by appropriate conductors. The cable shields have to be connected to the grounding conductor with low impedance to avoid interference from the shields as well.

Cable Types

For installation of the µCAN modules, please only use cables with a shield covering at least 80% of the core. Do not use cables with a metallized foil shield as this can be easily damaged on assembly and, therefore, does not guarantee proper shielding.

Cable Layout

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. In addition, grounding on both ends is not possible for certain measurement sensors. In these cases, grounding on one end would be an advantage if:

- an equipotential bonding is not possible,
- analogue signals of several mV or mA are to be transmitted (e.g. through the sensor).

The shield of the CAN bus cable must not lead inside the µCAN modules. Never connect the shield to one of the terminals inside the casing.

For stationary applications the shield of the CAN bus cable should be connected to an earthing conductor by metal clamps.

5.3 General Information on Wiring

All wires used within the system should be grouped in different categories. These could be e.g.: signal lines, data lines, power lines.

Power lines and data or signal lines should be arranged in seperate cable ducts or groups (ref. inductance coupling).

Data and signal lines should lead along ground planes as near as possible.

Observing the rules of proper wiring layout will avoid or impede interference of parallel wires to a large extend.

5.3.1 Groups of Wires

In order to achieve a EMC-compliant wiring layout the wires should be categorized as follows:

Group 1: shielded bus and data wires,
shielded analogue wires,
unshielded DC wires < 60V,
unshielded AC wires < 25V,
coaxial wires for monitors.

Group 2: unshielded DC wires > 60V
and < 400V,
unshielded AC wires > 25V
and < 400V

Group 3: unshielded AC / DC wires
< 400V

Combination of Groups

Based on this categorization the following combinations for arrangement in groups or cable ducts are possible:

Group 1 with Group 1, Group 2 with Group 2, Group 3 with Group 3

The following groups may be combined in separate cable ducts or groups without a minimum spacing:

Group 1 with Group 2

Other combinations of groups can only be realised if they are arranged in separate cable ducts or groups observing the admissible limit values.

5.4 CAN Cable

To connect the bus devices to the CAN bus an ISO11898-2 compliant cable must be used. The wires must comply with the following electrical specifications:

Characteristics	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

5.5 Power Supply

The µCAN.6.ai-SNAP modules are designed for industrial applications. The supply voltage may vary within a range from 9V..36V. The input is protected against reverse voltage.

Please make sure that the power supply is correctly connected to the respective screw terminals of the COMBICON plug. The positive line of the power supply for the µCAN module has to be connected to terminal **V+**. The terminal is internally bridged, thus permitting to connect through the power supply line.

The reference potential of the power supply is connected to one of the **GND** terminals. The GND terminals are internally bridged.

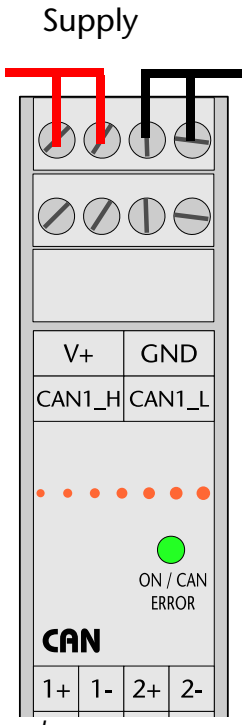


Abb. 6: Connection of power supply

Electronical parts are designed for supply voltages of up to 36V DC. Application of higher vorltages will destroy the µCAN module.

The shielding must not lead into the µCAN module or connected to one of the terminals. Shielding must be connected to the appropriate potential via special connectors outside the casing.

5.6 CAN Bus

The CAN-Bus is connected to the appropriate terminal of the COMBICON plug via a two-stranded wire.

To avoid electromagnetic interference, please ensure that the CAN bus cable does not cross the signal wires.

The CAN bus line with high potential must be connected to the terminal **CAN_H**. The CAN bus line with low protential must be connected to the terminal **CAN_L**.

The terminals CAN_H and CAN_L are internally bridged.

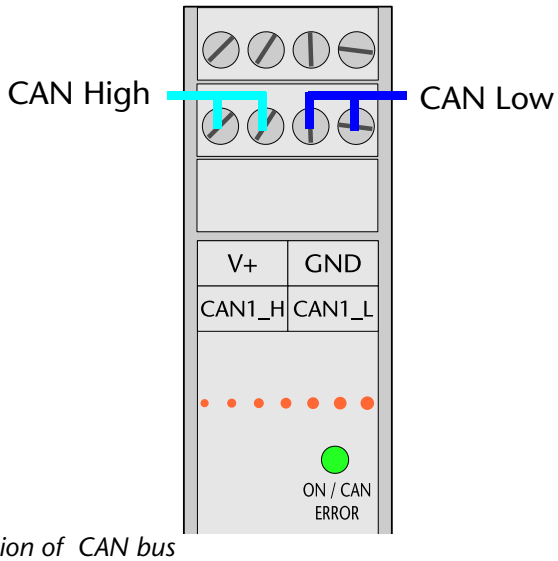


Abb. 7: Connection of CAN bus

Any reversal connection of the bus potentials will prevent communication on the bus.

A shielding must not lead into the µCAN module or connected to one of the terminals. Shielding must be connected to the appropriate potential via special connectors outside the casing.

If you use a 9-pin Sub-D connector, the conductor CAN-H has to be connected to pin 7 and the conductor CAN-L to pin 2 (according to CiA standards).

5.7 Module Address

The address of the µCAN modules is selected via an 8-pin DIP-switch which is located at the top in the middle of the board.

To select the address (i.e. node-ID, NID) you should use a small-size screw driver.

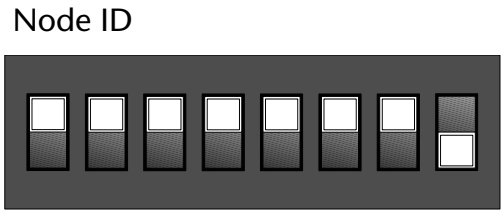


Abb. 8: Setup of module address (address 9 is shown here)

5

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

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

The selected address is read during initialization of the µCAN module. The µCAN module runs with the selected node ID until an new node ID is selected and a reset is performed.

If all "Node ID" switches are in OFF position and the DIP switches of the baudrate are also switched to OFF, the µCAN.6.ai-SNAP will be started in LSS mode.

The DIP switch 8 must always be in OFF position.

The µCAN module will not start if the switches are in a wrong position, which will be indicated by the "Error" LED (refer to "Diagnosis" on page 37).

5.8 Baud Rates

The Baud rates of the µCAN modules are selected via a 4-pin DIP switch which is located on the left of the DIP switch for selecting the module address at the top of the board.

To select the Baud rate you should use a small-size screw driver.

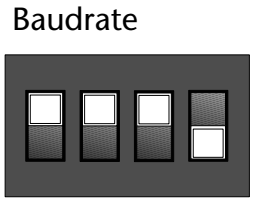


Abb. 9: Setup of Baud rate (1 MBit/s is shown here)

The supported Baud rates of the µCAN field modules are listed in the tabel below. The values are recommended by the CiA.

Baud Rate	DIP switch position			
	1	2	3	4
Autobaud / LSS	0	0	0	0
Autobaud	1	0	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: Configuration of Baud rate

The Baud rates 10 kBit/s and 20 kBit/s are not supported by the µCAN.6.ai-SNAP module. LSS will be used only if all module ID switches are turned to OFF position.

If the module is switched to Autobaud, it will automatically detect the Baud rate used on the CAN bus.

If the module is switched to LSS mode, the pre-set Baud rate and module address of the device will be used.

If the Baud rate is set to a value which is not supported by the module, this will be indicated by the "Error" LED (refer to “Diagnosis” on page 37).

5.9 Termination

The last module on a CAN line has to be terminated with a resistor (120 Ohm). Thus, the CAN bus line is properly terminated and does not reflect back to the communication lines.

For termination of the µCAN module the slide switch has to be switched to the respective position using a small-size screw driver.

Please ensure that only µCAN modules are switched to termination which are at the end of a CAN bus line. If the module is disconneted from power supply, you will be able to measure a value of 60 Ohm between the lines CAN-H and CAN-L.

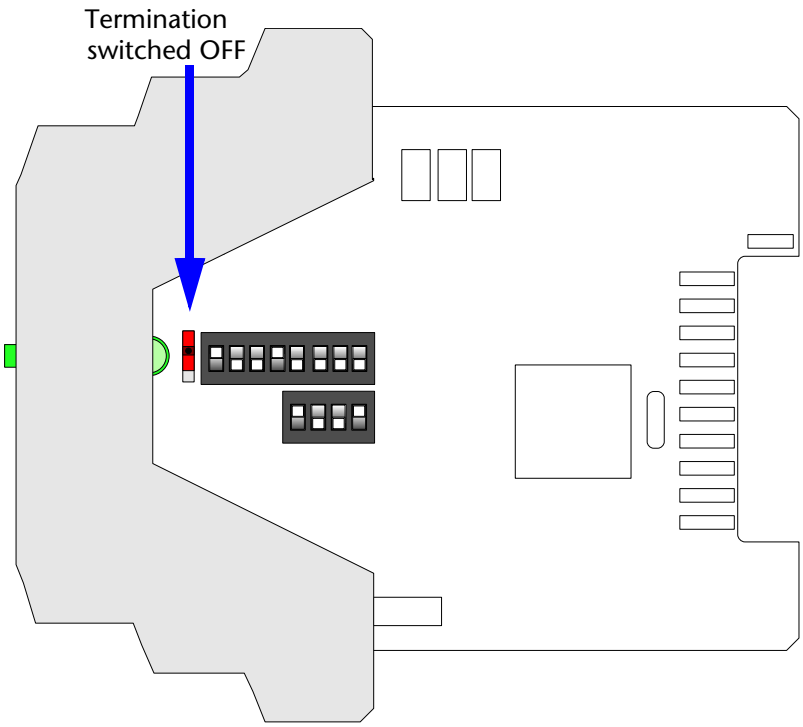


Abb. 10: Switching of termination

In the drawing above the termination is switched off. The module is used as T-piece on a CAN bus line. Therefore, the line has to be terminated through another module with 120 Ohm.

5

6. Analogue Inputs

This part of the manual is aimed at demonstrating how to connect the different types of signals to the µCAN.6.ai-SNAP.

The µCAN.6.ai-SNAP is equipped with six analogue measurement inputs which are numbered in ascending order.

When connecting signal lines, please observe the respective EMC rules. Only proper, EMC-compliant wiring ensures smooth operation of the µCAN module.

Switch power off before connecting signal lines to the µCAN module to avoid serious damage to the electronic components.

Depending on the type of your µCAN.6.ai-SNAP module (voltage or current measurement) the input impedance is dimensioned accordingly.

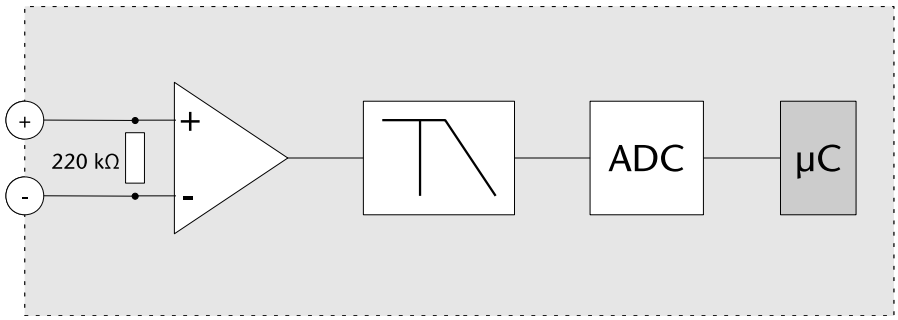


Abb. 11: Switching for voltage measurement

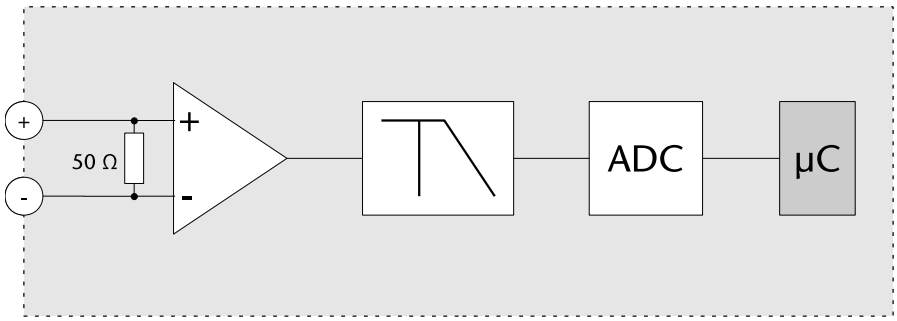


Abb. 12: Switching for current measurement

6.1 Connection of Signal Lines

When connecting the lines for standard signal evaluation, only terminal + and - of the respective channel are to be used. The measurement channels must not be bridged.

The positive signal of the respective measurement zone is connected to terminal "+" of the selected channel, the negative signal must be connected to the corresponding terminal "-" of the same channel. The zones are numbered 1 to 6.

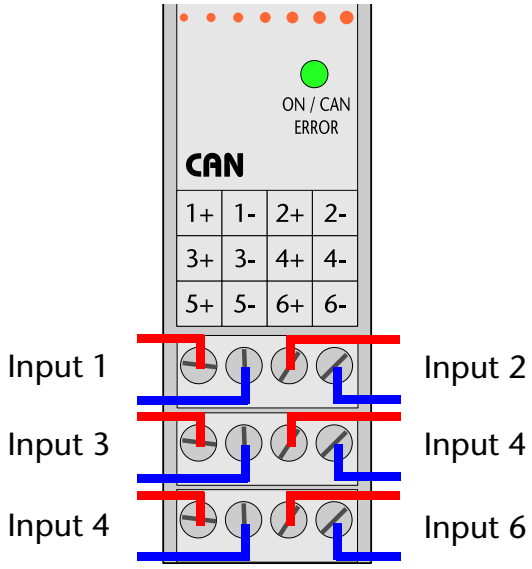


Abb. 13: Measurement inputs of the µCAN.6.ai-SNAP module

To avoid interference with electronic parts the shielding of the signal lines must not lead into the µCAN module. The shielding must be connected to the appropriate potential via special connectors outside the casing.

Please ensure that the signals do not exceed ±50 V for voltage input and ±40 mA for current input. Exceeding these limit values may destroy the electronic components of the module.

6.2 Monitoring of Limit Values

The µCAN.6.ai-SNAP module monitors measurement signals with regard to whether they exceed the limit values considering the preset signal type.

The following table shows an overview of these limit values:

Sensor type	Measurement limit value	Signal limit values
±10 V DC	- 10,001 V +10,001 V	- 10,200 V +10,200 V
0..20 mA	- 0,001 mA +20,001 mA	- 0,200 mA +20,200 mA
4..20 mA	+ 3,999 mA +20,001 mA	+ 3,800 mA +20,200 mA

Tabelle 4: Overview of monitored limit values

If the measurement limit value is exceeded, the measured value will be entered into the status information of the respective measuring channel (siehe “AI Status” auf Seite 64).

If the signal limit values are exceede, an emergency message will be sent (siehe “Emergency-Botschaft” auf Seite 85) and the status of the measuring channel will be reset.

Furthermore, an invalid measurement value EEEEh = -4370d (signed) = 61166d (unsigned) will be emitted.

The error message will remain until the measuring value falls below the signal limit value again.

6

7. Diagnosis

All µCAN modules of the µCAN.6.ai-SNAP type are equipped with an LED for indication of device status and errors.

The µCAN.6.ai-SNAP has a bi-colour LED (green/red) marked "ON / CAN ERROR".

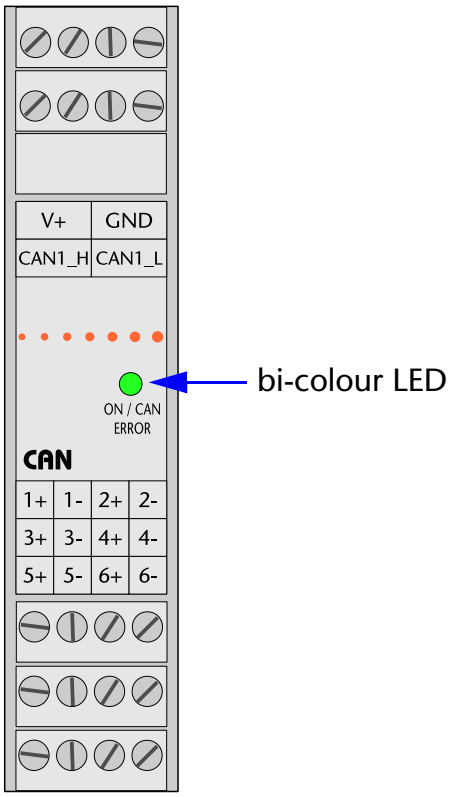


Abb. 14: Position of the bi-colour LED on the µCAN module

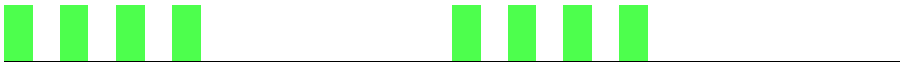
In normal operation the LED should either have a green colour or be blinking. A red steady light or a red blinking of the LED indicates an error condition.

7.1 Status of Network

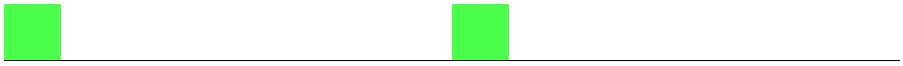
The LED indicates the condition of the CANopen NMT state machine and the error condition of the CAN controller.

7.1.1 Representation of CANopen NMT states

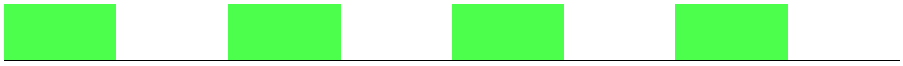
The green light of the LED indicates the state of the CANopen Network Management (NMT).



Device setup (Autobaud Detection)



NMT state: Device in "Stopped" state



NMT state: Device in "Pre-operational" state

NMT state: Device in "Operational" state

7.1.2 Representation of CAN state

The red light of the LED indicates the error state of the CAN controller and the µCAN module. In non-fault condition the LED is green.

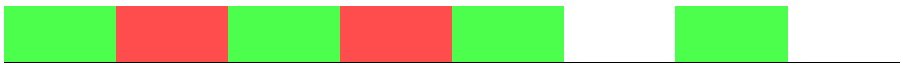
CAN state: Controller in "Warning" state

CAN state: Controller in "Error Passive" state

CAN state: Controller in "Bus-Off" state or
error in preset Baud rate/ module address

7.1.3 Combined Representation

The combination of green and red lights allows representation of both the CAN controller condition and the CANopen NMT state.



Device in "Pre-operational" state, CAN Controller in "Warning" state

Device in "Operational" state, Controller in "Error Passive" state

7

8. CANopen Protokoll

This chapter provides important information for the user on how to connect the µCAN module to a CANopen-Manager and put it into operation. A CANopen-Manager can be a PC with a CAN interface, a PLC or any other control unit such as an actuator.

For more information about the CANopen-Manager please refer to the supplied documentation of your devices.

This documentation describes the currently implemented functions and services of the µCAN modules.

8.1 General Information

The identifiers of the µCAN module are set up after initial start-up according to the **Predefined Connection Set** which is described in detail in the CANopen communication profile CiA 301. The following table provides an overview of the supported services.

Object	COB-ID (dez.)	COB-ID (hex)
Network Management	0	0x000
SYNC	128	0x080
EMERGENCY	129 - 255	0x081 - 0x0FF
PDO 1 (transmit)	385 - 511	0x181 - 0x1FF
PDO 1 (receive)	513 - 639	0x201 - 0x27F
SDO (transmit)	1409 - 1535	0x581 - 0x5FF
SDO (receive)	1537 - 1663	0x601 - 0x67F
Heartbeat / Boot-up	1793 - 1919	0x701 - 0x77F

Table 5: Distribution of identifiers

The above mentioned directions (transmit / receive) signify the transfer directions from the µCAN.6.ai-SNAP to adjacent devices.

8

8.2 Network Management

The device state (Stop / Pre-Operational / Operational) can be changed by means of Network Management (NMT) messages.

Start Node

Start Node

ID	*DLC*	*B0*	*B1*
0	2	01h	NID

NID = module address, 00h = all CAN modules

The „Start Node“ command sets the CAN module to operation mode which will enable the CAN module to communicate via PDOs.

Stop Node

Stop Node

ID	*DLC*	*B0*	*B1*
0	2	02h	NID

NID = module address, 00h = all CAN modules

The „Stop Node“ command sets the CAN-node to stop mode which will prevent any communication via SDOs or PDOs.

Pre-Operational

Enter Pre-Operational

ID	*DLC*	*B0*	*B1*
0	2	80h	NID

NID = module address, 00h = all CAN modules

The „Enter Pre-Operational“ command sets the CAN-node to pre-operational mode which will prevent communication via PDOs.

Reset Node

Reset Node

ID	*DLC*	*B0*	*B1*
0	2	81h	NID

NID = module address, 00h = all CAN modules

The „Reset Node“ command will execute a hardware reset of the module. After reset the CAN module will be set to pre-operational mode automatically and will send a „Boot-up Message“.

8

8.3 SDO Communication

All parameters of the CAN module (object dictionary) are accessed via SDO (Service Data Object). A SDO message has the follwing contents:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
	8	CMD	Index		Sub-Index	Data bytes			

The Command Byte (**CMD**) is signified as follows:

SDO-Client (CANopen Master)	**SDO-Server (CANopen Slave)**	**Function**
22_h	60_h	write, undefined size
23_h	60_h	write, 4 bytes
27_h	60_h	write, 3 bytes
$2B_h$	60_h	write, 2 bytes
$2F_h$	60_h	write, 1 byte
40_h	42_h	read, undefined size
40_h	43_h	read, 4 bytes
40_h	47_h	read, 3 bytes
40_h	$4B_h$	read, 2 bytes
40_h	$4F_h$	read, 1 byte

Table 6: Commands for SDO expedited messages

In case of **Index** and **Data bytes** the LSB is transmitted first!

The minimum time delay between two successive SDO messages must not be lower than10 ms.

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-Index	Abort code			

The ID of the message as well as the index and sub-index refer to the ID which emitted the access error.

The abort code may have the following values:

Error code	Description
0504 0001h	Client / Server command specifier unknown / invalid
0601 0000h	Access to object not supported
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 object dictionary
0609 0011h	Sub-Index does not exist in object dictionary

Table 7: SDO abort codes

8.4 Object Dictionary

This chapter describes the objects implemented in the µCAN.6.ai-SNAP module. For further information, please refer to the CANopen communication profile CiA 301 and the device profile CiA 404.

EDS

The objects implemented in the µCAN.6.ai-SNAP are listed in an "Electronic Data Sheet" (EDS). You can download the respective EDS file from the MicroControl homepage.

8.4.1 Communication Profile

The µCAN.6.ai-SNAP module comprises the following objects of the communication profile CiA 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
1800h	1st Transmit PDO Parameters
1801h	2nd Transmit PDO Parameters
1802h	3rd Transmit PDO Parameters
1803h	4th Transmit PDO Parameters
1A00h	1st Transmit PDO Mapping
1A01h	2nd Transmit PDO Mapping
1A02h	3rd Transmit PDO Mapping

Table 8: Supported objects of the communication profile

Index	Name
1A03h	4th Transmit PDO Mapping
1F80h	NMT Startup

Table 8: Supported objects of the communication profile

Device Profile

Index 1000h

The object at index 1000h describes the device profile.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned32	ro	Device Profile	0002 0194h

The object is read-only. Only sub-index 0 is supported. Access to other sub-indices will result in 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 a response the µCAN.6.ai-SNAP will send:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	43h	00h	10h	00h	94h	01h	02h	00h

Byte 4 + Byte 5 = 0194h = 404d (Device Profile Number)
Byte 6 + Byte 7 = 0002h = 2 (Additional Information)

8

Error Register

Index 1001h

The object at Index 1001h is the error register of the device.

Sub-Index	Data Type	Access	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 result in 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	00h	00h	00h	00h	00h

As a response you will receive the status of the error register of the device.

The following error types are supported and displayed:

B4	*B5*	*B6*	*B7*	Description
02h	00h	00h	00h	Current Error: due to an error in current measurement.
04h	00h	00h	00h	Voltage Error: due to an error in voltage measurement.
10h	00h	00h	00h	Communication Error: due to communication error on the CAN-Bus. For detailed descriptions of errors please refer to refer to “Emergency Message” on page 101.

Table 9: Supported errors in the error register

Manufacturer Status Register

Index 1002h

The object at Index 1002h is the state register of the device.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Manufacturer Status Register	00h

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

Example: read parameter, module-ID = 2, Index = 1002h

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	02h	10h	00h	00h	00h	00h	00h

As a response you will receive information on the state of the device.

This register provides status information for the AD converter and the EEPROM.

In the register you may set different bits which are explained in the following table:

B4	*B5*	*B6*	*B7*	Description
x1h	xxh	xxh	xxh	EEPROM Error: error in communication with the EEPROM detected.
x2h	xxh	xxh	xxh	EEPROM Error: error in writing the data occurred..
1xh	xxh	xxh	xxh	AD Converter Error: communication with AD converter could not be established.
2xh	xxh	xxh	xxh	AD Converter on hold: AD converter has been stopped.

Table 10: Status information in the Manufacturer Status Register

Error Register

Index 1003

The object at Index 1003h will provide access to the error history. The last 4 errors occurred can be read through the sub-indices 1 to 10.

Sub-Index	Data Type	Access	Name	DefaultValue
0	Unsigned8	rw	Number of errors	00h
1 .. 4	Unsigned32	ro	Standard error field	0000 0000h

Sub-indices 0 to 4 are supported. Access to other sub-indices will result in an error message. Writing access to sub-index 0 will clear the error register.

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

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

As a response the module will return the error value at third-last position in the history.

Device Name

Index 1008

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

Sub-Index	Data Type	Access	Name	Default Value
0	Visible String	ro	Device name	mCAN.6.ai-SNAP

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

Hardware Version

Index 1009h

The object at Index 1009h contains the hardware version.

Sub-Index	Data Type	Access	Name	Default Value
0	Visible String	ro	Hardware version	1.02

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

Software Version

Index 100Ah

The object at Index 100Ah contains the software version.

Sub-Index	Data Type	Access	Name	Default Value
0	Visible String	ro	Software version	1.00

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

Store Parameter

Index 1010h

The object at Index 1010h saves parameters into a non-volatile memory.

Sub-Index	Data Type	Access	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

The storage function is initiated by sending index 1010h with the message "save" (in ASCII) to sub-index 1. The message is structured as follows:

Example: save all parameters, module address = 2,
Index = 1010h

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

As a response the device will send:

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

After initiating the storage function the parameters are stored in the non-volatile memory (EEPROM).

Restore Default Parameters

Index 1011h

The object at Index 1011h restores the default parameters of the device.

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

Restoring of default parameters is initiated by sending index 1011h with the message "load" (in ASCII) to sub-index 1. The message is structured as follows:

Example: load default parameters, module address = 2,
Index = 1011h

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

As a response the device will send the following message:

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

8

COB-ID Emergency Message

Index 1014h

This object defines the COB-ID of emergency messages.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned32	rw	COB-ID EMCY	80h + NID

The default value for the identifier of the emergency message is 80h + set node-ID (1 to 127).

Identity Object

Index 1018h

The object at Index 1018h provides information about the identity object of the device.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	4
1	Unsigned32	ro	Vendor ID	0000 000Eh
2	Unsigned32	ro	Product Code	0010 7AC0h
3	Unsigned32	ro	Revision Number	0306 0100h
4	Unsigned32	ro	Serial Number	-

Only sub-indices 0 to 4 are supported. An access to other sub-indices will result in an error message.

Vendor ID

The Vendor ID contains a unique value allocated to each manufacturer of CAN-open devices. The numbers are centrally assigned and managed by the CAN in Automation (CiA). The Vendor-ID 0x0000000E has been assigned to MicroControl.

Product Code

The Product Code is a manufacturer-specific code which in case of MicroControl products corresponds to the order number stated in our product catalogue.

Revision Number

The Revision Number states the software version. The number consists of two 16 bit values. The upper 16 bit values signify a modification in the CAN part of the software, the lower 16 bit values signify a modification of the application software of the device.

Serial Number

The Serial Number identifies a specific device.

Error Behaviour

Index 1029h

If a device failure (CANopen NMT Error Protocol) is detected and the device is in operational mode, the device will be switched to pre-operational mode automatically. The error behaviour of individual devices may be changed via index 1029h.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	number of entries	1
1	Unsigned8	rw	Communication error	00h

The following values are possible:

Value	Description
00h	Default setting, change to pre-operational
01h	Keep current NMT mode
02h	Change to NMT mode "Stopped"

The following failures are considered:
- Error in node guarding
- Error in heartbeat

NMT Startup

8

Index 1F80h

This object defines the NMT startup behaviour of the device.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned32	rw	NMT Startup	0000 0000h

Only sub-index 0 is supported. An access to other sub-indices will result in an error message. The object defines the startup behaviour of the device after initialisation (Power-Up / Reset-Node). The following values are possible:

Value	Description
00h	Default setting, change to pre-operational
02h	Send NMT "Start All Nodes"
08h	Change to NMT mode "Operational"

8.4.2 Manufacturer-Specific Objects

The module µCAN.6.ai-SNAP comprises the following manufacturer-specific objects:

Index	Name
2010h	Customer Data
201Ah	COB-ID Storage
2E00h	PDO Data Format
2E10h	Disable BootUp Message
2E22h	Bus Statistic

Table 11: Manufacturer-specific objects

Customer Data

Index 2010h

The object at Index 2010h can save up to 8 words in the EEPROM of the µCAN module.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	04h
1	Unsigned32	rw	Customer Data 1	-
2	Unsigned32	rw	Customer Data 2	-
..	..	..	..	..
8	Unsigned32	rw	Customer Data 8	-

Only sub-indices 0 to 8 are supported. An access to other sub-indices will result in an error message.

Writing access to sub-indices 1 to 8 will automatically save the value in the EEPROM. Access to object 1010h is not necessary.

COB-ID Storage

Index 201Ah

This object defines the behaviour of the stored identifiers for PDO and EMCY services when changing the module address.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	rw	COB-ID Storage	00h

Only sub-index 0 is supported. An access to other sub-indices will result in an error message. The following values are possible:

Value	Description
00h	Keep stored identifiers (PDO/EMCY) when changing module address
01h	Discard stored identifiers (PDO/EMCY) when changing module address. Change to pre-defined connection set
02h	Calculate identifier for PDO/EMCY from module addresse + stored value

The object 201Ah is used in combination with the objects 1010h, 1014h, 1800h and 1801h.

PDO Data Format

Index 2E00h

This object determines the format, Intel (Little-Endian) or Motorola (Big-Endian), used to send PDO data.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	rw	PDO Data Format	00h

Only sub-index 0 is supported. An access to other sub-indices will result in an error message. The following values are possible:

Value	Description
00h	PDO data will be sent in Intel format
01h	PDO data will be sent in Motorola format

Changing to Motorola format for transmission of PDO data does not comply with CANopen specifications.

Disable BootUp Message

Index 2E10h

This object determines whether or not the µCAN.6.ai-SNAP module will send a BootUp message after startup or reset node.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	rw	Disable BootUp Message	00h

Only sub-index 0 is supported. An access to other sub-indices will result in an error message. The following values are possible:

Value	Description
00h	BootUp message after startup or reset node
01h	No BootUp message

Disabling the BootUp message does not comply with CANopen specifications.

8

Bus Statistics

Index 2E22h

The object at Index 2E22h can call up the bus statistics.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Number of entries	03h
1	Unsigned32	ro	CAN Receive Count	-
2	Unsigned32	ro	CAN Transmit Count	-
3	Unsigned32	ro	CAN Error Count	-

Only sub-indice 0 to 3 are supported. Access to other sub-indices will result in an error message.

The number of messages received is in sub-index 1, the number of messages sent is in sub-index 2. The number of CAN Error-Frames is in sub-index 3.

8.4.3 Device Profile CiA 404

The module µCAN.6.ai-SNAP comprises the following objects of the device profile CiA 404:

Index	Name
6110h	AI Sensor Type
6112h	AI Operating Mode
6131h	AI Physical Unit Process Value
6132h	AI Decimal Digits Process Value
6150h	AI Status
61A0h	AI Filter Type
61A1h	AI Filter Constant
7100h	AI Field Value
7120h	AI Input Scaling 1 Field Value
7121h	AI Input Scaling 1 Process Value
7122h	AI Input Scaling 2 Field Value
7123h	AI Input Scaling 2 Process Value
7124h	AI Input Offset
7130h	AI Process Value

Table 12: Supported objects of the Device Profile

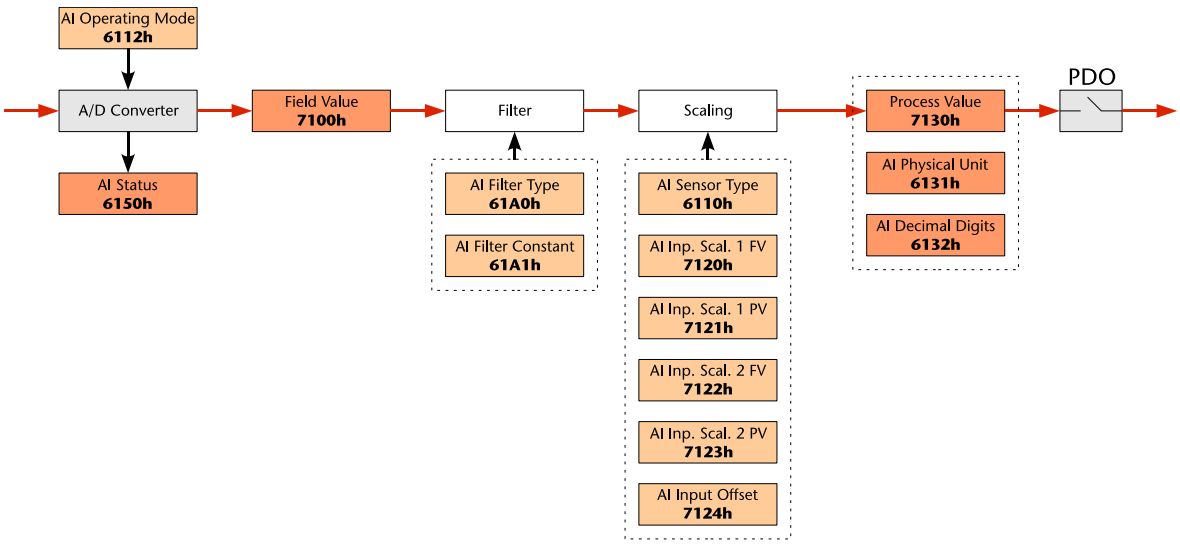


Abb. 15: Blockdiagram of supported functions of an analogue input

AI Sensor Type

Index 6110h

The object at Index 6110h can adjust and/or call up the sensor type.

Sub-Index	Data Type	Access	Name	Default Value [V] | [mA]
0	Unsigned8	ro	Largest Sub-Index	06h
1	Unsigned16	rw	AI Sensor type of Channel 1	0029h | 0034h
2	Unsigned16	rw	AI Sensor type of Channel 2	0029h | 0034h
3	Unsigned16	rw	AI Sensor type of Channel 3	0029h | 0034h
4	Unsigned16	rw	AI Sensor type of Channel 4	0029h | 0034h
5	Unsigned16	rw	AI Sensor type of Channel 5	0029h | 0034h
6	Unsigned16	rw	AI Sensor type of Channel 6	0029h | 0034h

The object is read and write. Only sub-indices 0 to 6 are supported. Access to other sub-indices will result in an error message.

The sensor type is according to the version of the µCAN.6.ai-SNAP module. µCAN modules for acquisition of voltage signals can only be equipped with sensor type ±10 V at the moment. The standard setting of modules for acquisition of current signals is sensor type 0...20 mA.

8

The following table shows the supported sensor types and their corresponding values.

Value	*Sensor Type*
29h	±10V
33h	4..20mA
34h	0..20mA

Table 13: Supported sensors

Example: Read sensor type of first measuring channel (Sub-Index 1), module address is 2.

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	10h	61h	01h	00h	00h	00h	00h

As a response you will receive the following message:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
581h	8	4Bh	10h	61h	01h	33h	00h	00h	00h

In this example, the value of the sensor type is 33h (byte 4). This means that the current sensor type is set to acquire values of 4..20 mA.

The setting of a sensor type always applies for two channels (1/2 as well as 3/4).

Example: Set sensor type of first (and second) measuring channel (Sub-Index 1) to acquisition of 4..20 mA signals, module address is 2

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
602h	8	2Bh	10h	61h	01h	34h	00h	00h	00h

Setting will be confirmed by the following message:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
582h	8	60h	10h	61h	01h	00h	00h	00h	00h

If the µCAN module receives values other than those stated in table 13 on page 64, an error message will be sent.

The set sensor type is not automatically stored in a non-volatile storage. Storage has to be triggered via Index 1010h (refer to “Store Parameter” on page 55).

AI Operating Mode

Index 6112h

The object at Index 6112h is used for setting and enquiring operating modes of the respective measuring inputs. Individual measuring channels can be switched on or off via the operating mode.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Unsigned8	rw	AI Operating Mode of Channel 1	01h
2	Unsigned8	rw	AI Operating Mode of Channel 2	01h
3	Unsigned8	rw	AI Operating Mode of Channel 3	01h
4	Unsigned8	rw	AI Operating Mode of Channel 4	01h
5	Unsigned8	rw	AI Operating Mode of Channel 5	01h
6	Unsigned8	rw	AI Operating Mode of Channel 6	01h

The object is read and write. Only sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message.

To switch a channel on, you have to transmit "1" to the device via the respective sub-index, and "0" to switch the channel off.

Example: Switch off measurement at channel 3 (sub-index 3) (byte 4 = 00h), module address is 2

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	2Fh	12h	61h	03h	00h	00h	00h	00h

As a response the µCAN.6.ai-SNAP will send the following message:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	60h	12h	61h	03h	00h	00h	00h	00h

Now, channel 3 is switched off, signals will not be evaluated . If this channel has caused an error before, it will now be reset. If you enquire data from this channel, you will receive the value 0.

The set operating mode is not automatically stored in a non-volatile storage. Storage has to be triggered via Index 1010h (refer to “Store Parameter” on page 55).

AI Physical Unit Process Value

Index 6131h

Index 6131h is read only. It indicates the physical unit of measure of the process values.

Sub-Index	Data Type	Access	Name	Default Value [V] | [mA]
0	Unsigned8	ro	Largest Sub-Index	06h
1	Unsigned32	rw	AI Physical Unit PV of Channel 1	00260000h | FD040000h
2	Unsigned32	rw	AI Physical Unit PV of Channel 2	00260000h | FD040000h
3	Unsigned32	rw	AI Physical Unit PV of Channel 3	00260000h | FD040000h
4	Unsigned32	rw	AI Physical Unit PV of Channel 4	00260000h | FD040000h
5	Unsigned32	rw	AI Physical Unit PV of Channel 5	00260000h | FD040000h
6	Unsigned32	rw	AI Physical Unit PV of Channel 6	00260000h | FD040000h

The object is read and write. Only sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message.

Example: Read physical unit of measure of channel 3 (sub-index 3), module address is 2

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	31h	61h	03h	00h	00h	00h	00h

As a response the µCAN.6.ai-SNAP will send the following message:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	31h	61h	03h	00h	00h	26h	00h

Data byte 6 contains the value 26h, which corresponds to a unit of measure in V.

For a complete table of codes please refer to the CiA 303-2.

AI Decimal Digits Process Value

Index 6132h

Index 6132h is read only. It indicates the number of decimal places of the process values.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Unsigned8	rw	AI Decimal Digits PV of Channel 1	03h
2	Unsigned8	rw	AI Decimal Digits PV of Channel 2	03h
3	Unsigned8	rw	AI Decimal Digits PV of Channel 3	03h
4	Unsigned8	rw	AI Decimal Digits PV of Channel 4	03h
5	Unsigned8	rw	AI Decimal Digits PV of Channel 5	03h
6	Unsigned8	rw	AI Decimal Digits PV of Channel 6	03h

The object is read and write. Only sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message.

Example: Read number of decimal places of channel 3 (sub-index 3), module address is 2

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	32h	61h	03h	00h	00h	00h	00h

As a response the µCAN.6.ai-SNAP will send the following message:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	40h	32h	61h	03h	03h	00h	00h	00h

Data byte 4 contains the value 03h, i.e. the process values are stated to three decimal places.

AI Status

Index 6150h

The object at Index 6150h reads the state of the respective measuring channel.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Unsigned8	ro	AI Status of Channel 1	00h
2	Unsigned8	ro	AI Status of of Channel 2	00h
3	Unsigned8	ro	AI Status of of Channel 3	00h
4	Unsigned8	ro	AI Status of of Channel 4	00h
5	Unsigned8	ro	AI Status of of Channel 6	00h
6	Unsigned8	ro	AI Status of of Channel 6	00h

The object is read and write. Only sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message.

The value in the state register determines whether the measuring values have reached or exceeded the limit values. The following values are defined:

Wert	*Status*
00h	No failure
01h	Measuring Value not valid
02h	Positive Overload
04h	Negative Overload

Table 14: Possible values indicating the state of the measuring channel

Example: Read state of channel 3 (sub-index 3), module address is 2

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	50h	61h	03h	00h	00h	00h	00h

As a response you will receive the follwing message:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
582h	8	4Bh	50h	61h	03h	03h	00h	00h	00h

Value 03h of data byte 4 signifies a positive overflow. The measuring values of this channel are not valid.

The limit values of the measuring inputs are listed in chapter refer to “Überwachung der Grenzwerte” on page 31 of this manual.

AI Filter Type

Index 61A0h

The object at Index 61A0h is used for setting and reading the filter type of the respective measuring channel.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Unsigned8	rw	AI Filter Type of Channel 1	00h
2	Unsigned8	rw	AI Filter Type of of Channel 2	00h
3	Unsigned8	rw	AI Filter Type of of Channel 3	00h
4	Unsigned8	rw	AI Filter Type of of Channel 4	00h
5	Unsigned8	rw	AI Filter Type of of Channel 5	00h
6	Unsigned8	rw	AI Filter Type of of Channel 6	00h

The object is read and write. Only sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message.

The following values are defined for selection of filter type:

Value	*Filter*	*Calculation*
00h	No Filter	-
01h	Moving average	$Data_N = Data_{N-1} + \frac{NewData - Data_{N-1}}{\text{filter constant}}$

Table 15: Possible filters

Example: Read filter type of channel 3 (sub-index 3), module address is 2

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	A0h	61h	03h	00h	00h	00h	00h

As a response you will receive the following message:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	4Bh	A0h	61h	03h	01h	00h	00h	00h

Value 01h of data byte 4 signifies selection of a "moving average" filter.

The set filter type is not automatically stored in a non-volatile storage. Storage has to be triggered via Index 1010h (see refer to “Store Parameter” on page 55).

AI Filter Constant

Index 61A1h

The object at Index 61A1h is used for setting and enquiring the filter constant of the respective channel.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Unsigned8	rw	AI Filter Constant of Channel 1	01h
2	Unsigned8	rw	AI Filter Constant of Channel 2	01h
3	Unsigned8	rw	AI Filter Constant of Channel 3	01h
4	Unsigned8	rw	AI Filter Constant of Channel 4	01h
5	Unsigned8	rw	AI Filter Constant of Channel 5	01h
6	Unsigned8	rw	AI Filter Constant of Channel 6	01h

The object is read and write. Only sub-indices 0 to 6 are supported. Access to other sub-indices will result in an error message.

The value of the filter constant may be selected from 1 to 50. If a value other than 1 to 50 is set for filter constant an error message will be sent.

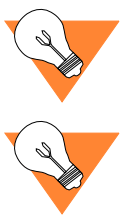
Example: Write filter constant 5 to channel 3 (sub-index 3), module adress is 2

8

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	2Bh	A1h	61h	03h	05h	00h	00h	00h

As a response the µCAN.6.ai-SNAP will send the following message:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	60h	A1h	61h	03h	00h	00h	00h	00h



Please ensure that the selected filter is set to Index 61A0h as well. If e.g. no filter is selected, writing of the filter constant will not have any effect.
The set filter constand is not automatically stored in a non-volatile storage. Storage has to be triggered via Index 1010h (see refer to “Store Parameter” on page 55).

AI Field Value

Index 7100h

Index 7100h is read only. It indicates the measuring value of the selected channel. The measuring value is not linearized, however it may be filtered. It represents the current (and possibly filtered) value sent by the AD converter.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Signed16	ro	AI Field Value of Channel 1	0000h
2	Signed16	ro	AI Field Value of Channel 2	0000h
3	Signed16	ro	AI Field Value of Channel 3	0000h
4	Signed16	ro	AI Field Value of Channel 4	0000h
5	Signed16	ro	AI Field Value of Channel 5	0000h
6	Signed16	ro	AI Field Value of Channel 6	0000h

The object is read only. Sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message.

Example: Read AD value of channel 3 (sub-index 3), module address is 2

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	00h	71h	03h	00h	00h	00h	00h

As a possible response you will receive the following message:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	4Bh	50h	61h	03h	11h	0Ah	00h	00h

Data byte 4 (LowByte) and 5 (High Byte) indicate the measuring value 0A11h of the AD converter.

Simultaneous inquiry of all four channels is possible via PDOs, but not via SDOs (see refer to “PDO Communication” on page 89).

AI Input Scaling 1 Field Value

Index 7120h

The object at Index 7120h in combination with objects 7121h, 7122h and 7123h are used to scale the measuring values of each channel.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Signed16	rw	AI Imput Scaling 1 FV of Channel 1	0000h
2	Signed16	rw	AI Imput Scaling 1 FV of Channel 2	0000h
3	Signed16	rw	AI Imput Scaling 1 FV of Channel 3	0000h
4	Signed16	rw	AI Imput Scaling 1 FV of Channel 4	0000h
5	Signed16	rw	AI Imput Scaling 1 FV of Channel 5	0000h
6	Signed16	rw	AI Imput Scaling 1 FV of Channel 6	0000h

The object is read and write. Only sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message. For further information on how to use this object with regard to scaling, please refer to chapter refer to “Scaling” on page 83 of this manual.

8

AI Input Scaling 1 Process Value

Index 7121h

The object at Index 7121h in combination with objects 7120h, 7122h and 7123h are used to scale the measuring values of each channel.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Signed16	rw	AI Imput Scaling 1 PV of Channel 1	0000h
2	Signed16	rw	AI Imput Scaling 1 PV of Channel 2	0000h
3	Signed16	rw	AI Imput Scaling 1 PV of Channel 3	0000h
4	Signed16	rw	AI Imput Scaling 1 PV of Channel 4	0000h
5	Signed16	rw	AI Imput Scaling 1 PV of Channel 5	0000h
6	Signed16	rw	AI Imput Scaling 1 PV of Channel 6	0000h

The object is read and write. Only sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message. For further information on how to use this object with regard to scaling, please refer to chapter refer to “Scaling” on page 83 of this manual.

AI Input Scaling 2 Field Value

Index 7122h

The object at Index 7122h in combination with objects 7120h, 7121h and 7123h are used to scale the measuring values of each channel.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Signed16	rw	AI Imput Scaling 2 FV of Channel 1	0000h
2	Signed16	rw	AI Imput Scaling 2 FV of Channel 2	0000h
3	Signed16	rw	AI Imput Scaling 2 FV of Channel 3	0000h
4	Signed16	rw	AI Imput Scaling 2 FV of Channel 4	0000h
5	Signed16	rw	AI Imput Scaling 2 FV of Channel 5	0000h
6	Signed16	rw	AI Imput Scaling 2 FV of Channel 6	0000h

The object is read and write. Only sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message. For further information on how to use this object with regard to scaling, please refer to chapter refer to “Scaling” on page 83 of this manual.

AI Input Scaling 2 Process Value

Index 7123h

The object at Index 7123h in combination with objects 7120h, 7121h and 7122h are used to scale the measuring values of each channel.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Signed16	rw	AI Imput Scaling 2 PV of Channel 1	0000h
2	Signed16	rw	AI Imput Scaling 2 PV of Channel 2	0000h
3	Signed16	rw	AI Imput Scaling 2 PV of Channel 3	0000h
4	Signed16	rw	AI Imput Scaling 2 PV of Channel 4	0000h
5	Signed16	rw	AI Imput Scaling 2 PV of Channel 5	0000h
6	Signed16	rw	AI Imput Scaling 2 PV of Channel 6	0000h

The object is read and write. Only sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message.

For further information on how to use this object with regard to scaling, please refer to chapter refer to “Scaling” on page 83 of this manual.

AI Input Offset

Index 7124h

The object at Index 7124h adds an offset to the process value of each channel.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Signed16	rw	AI Imput Offset of Channel 1	0000h
2	Signed16	rw	AI Imput Offset of Channel 2	0000h
3	Signed16	rw	AI Imput Offset of Channel 3	0000h
4	Signed16	rw	AI Imput Offset of Channel 4	0000h
5	Signed16	rw	AI Imput Offset of Channel 5	0000h
6	Signed16	rw	AI Imput Offset of Channel 6	0000h

The object is read and write. Only sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message.

For further information on how to use this object with regard to scaling, please refer to chapter refer to “Scaling” on page 83 of this manual.

8

AI Process Value

Index 7130h

Index 7130h is read only. It indicates the linearized process values of the selected channel. Linearization depends on the selected process size. The index is structured as follows:

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	06h
1	Signed16	ro	AI Process Value of Channel 1	0000h
2	Signed16	ro	AI Process Value of Channel 2	0000h
3	Signed16	ro	AI Process Value of Channel 3	0000h
4	Signed16	ro	AI Process Value of Channel 4	0000h
5	Signed16	ro	AI Process Value of Channel 5	0000h
6	Signed16	ro	AI Process Value of Channel 6	0000h

The object is read only. Sub-indices 0 to 6 are supported. An access to other sub-indices will result in an error message.

Example: Read process value of channel 3 (sub-index 3), module address is 2

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	30h	71h	03h	00h	00h	00h	00h

As a possible response you will receive the following message:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	4Bh	30h	61h	03h	45h	03h	00h	00h

Data byte 4 (LowByte) and 5 (High Byte) indicate the process value 0345h = 837d = 0,837 V in case of the voltage measurement-version and = 0,837 mA in case of current measurement version of the µCAN.6.ai-SNAP module.

If a measuring value is not within the valid measurement range, the value EEEEh = -4370d (signed) = 61166d (unsigned) will be transmitted.

At the same time, the objects error state of the device (see refer

to “Error Register” on page 51) and state of measuring inputs (see refer to “AI Status” on page 70) will show corresponding values.

If an analogue input error is detected, an emergency message will be sent on the bus (see refer to “Emergency Message” on page 101).

8.5 Scaling

In many cases voltage and current values are measured which represent different process variables.

The µCAN.6.ai-SNAP module facilitates scaling to desired process variables directly on the module.

To do so, the line $y = mx + b$ is used which passes through the points $(x_1 \mid y_1)$ and $(x_2 \mid y_2)$.
The slope is defined as follows:

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{PV_2 - PV_1}{FV_2 - FV_1}$$

By adjusting the slope m and the offsets b scaling and representation of process variables can be varied individually.
The following table represents the objects in their corresponding values.

Variable	Abbrev.	Object	Name	Default Value [d] [V] | [mA]
m	$x_1 = FV_1$	7120h	AI Input Scaling 1 FV	0000h | 0000h
	$y_1 = PV_1$	7121h	AI Input Scaling 1 PV	0000h | 0000h
	$x_2 = FV_2$	7122h	AI Input Scaling 2 FV	0000h | 0000h
	$y_2 = PV_2$	7123h	AI Input Scaling 2 PV	0000h | 0000h
b	b	7124h	AI Input Offset	0000h | 0000h

Table 16: Variables for scaling

To scale the process values, at least one of the "AI Input Scaling" values must unequal "0".
In standard configuration the Field Value is 0 ,000 V = 0d and bei +10,000 V = 32000d.
In the current measurement version the configuration is: 0 mA = 0d and +20,000 mA = 31360d.

To complete the setting, objects 6131h and 6132h should be adjusted accordingly, so that process values are correctly represented in their physical unit as well as in the number of decimal places.

The set parameters are not automatically stored in a non-volatile storage. Storage has to be triggered via index 1010h (see refer to “Store Parameter” on page 55).

8.6 CANopen Device Monitoring

For monitoring the CANopen device two mechanisms (protocols) are available:

- Heartbeat Protocol
- Node Guarding

CAN in Automation recommends to exclusively use the Heartbeat Protocol for monitoring purposes (CiA AN 802 V1.0: CANopen statement on the use of RTR-messages).

8.6.1 Heartbeat Protocol

Other users in the network may use the Heartbeat Protocol to check the operational state of the module and its current condition.

Heartbeat ID

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

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

Netzwerk state	Code (dez.)	Code (hex)
Bootup	0	00h
Stopped	4	04h
Operational	5	05h
Pre-Operational	127	7Fh

Table 17: Status information for Heartbeat

After Power-on the module will send a so-called „Boot-up Message“.

Example: Power-on of µCAN module with address 2

ID	*DLC*	*B0*
702h	1	00h

Consumer Heartbeat Time

Index 1016h

The object at Index 1016h defines the Consumer Heartbeat Time.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Number of objects	02h
1	Unsigned32	rw	Heartbeat Cons. 1	0000 0000h
2	Unsigned32	rw	Heartbeat Cons. 2	0000 0000h

The µCAN.6.ai-SNAP is able to monitor two other devices (Heartbeat Producer). If a failure of a Heartbeat Producer occurs within a preset time, an emergency message with the value 8130h (Life guard error or heartbeat error) will be emitted.
The 32 bit value sets the time and the device address of the device to be monitored.

Bit 31 ... 24	*Bit 23 ... 16*	*Bit 15 ... 0*
reserved (00h)	device address	Heartbeat Producer time

The time value is stated in milliseconds. If the time value is set to "0" or the device address is set to "0" or to a value larger than 127, the Consumer Heartbeat Time will not be used or activated. The Consumer Heartbeat Time will be activated after receipt of the first Producer Heartbeat.

8

Producer Heartbeat Time

Index 1017h

The object at Index 1017h sets the Producer Heartbeat Time. The time is stated in milliseconds. If the time is set to 0 ms the Heartbeat Protocol will be switched off.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned16	rw	Producer Time	0000h

The object is read and write. Only sub-index 0 is supported. Access to other sub-indices will result in an error message.

Example: Producer Time 1000 ms, device adress 1

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

As a response you will receive the following message:

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

The Producer Heartbeat Time is not automatically stored in an non-volatile storage. Storage has to be triggered via index 1010h.

8.6.2 Node Guarding

In cyclic device monitoring (Node Guarding) the NMT master polls each NMT slave at regular intervals. The polled NMT slaves will check if the "Node Guarding" is performed within the predefined interval (Life Guarding). This is necessary to ascertain whether the master is still active.

If the device is not polled within the predefined time interval, an emergency message with the value 8130h (Life guard error or heartbeat error) will be sent.

Time intervals for Device Monitoring

Index 100Ch

The object at Index 100Ch defines the interval of device monitoring by multiplying the time by the value from Index 100Dh.

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

The time is stated in milliseconds. The value 0000h deactivates device monitoring.

Factor for Device Monitoring

Index 100Dh

The object at Index 100Dh registers the factor for defining the interval of device monitoring, which is defined via Index 100Ch.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	rw	Life time factor	00h

The value 00h deactivates the device monitoring function.

8.7 PDO Communication

PDOs (Process Data Objects) are used for process data transmission.

PDO communication is possible only in "Operational" mode of the devices.

8.7.1 Transmission Modes

Synchronous Transmission

Synchronous transmission modes will be applied if a user is able to trigger the SYNC message in the CAN-open network. The synchronous transmission is defined by the "PDO transmission type" in the communications parameter of the respective process data object. e.g. a "transmission type" of 5 means that the device will send a process data object after each 5 SYNC messages received. For further details please refer to the CiA Draft Standart 301.

Cyclic Asynchronous Transmission

The input information can be transmitted cyclically (e.g. every 100 ms) via the transmit PDO. The duration of a transmission cycle is defined via the object 1800h or 1801h of the respective transmit PDO.

8.7.2 Transmit PDO 1 Parameter

Index 1800h

The object at Index 1800h defines the communication parameters of the transmit PDO 1.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	05h
1	Unsigned32	rw	COB-ID for PDO	180h+Node-ID
2	Unsigned8	rw	Transmission Type	01h
5	Unsigned16	rw	Event Timer	0000h

The object is read and write. Only sub-indices 0 to 2 and 5 are supported. Access to other sub-indices will result in an error message.

COB-ID for PDO

Subindex 1 sets the ID which receives the PDO. It is defined as follows:

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,

Table 18: Definition of COB-ID for PDO

To enable the PDO the most significant bit (b31) must be deleted. To disable the PDO the most significant bit must be set. In default setting the PDO active.

Transmission Type

Subindex 2 sets the transmission type of a message.

Transmission Type	Description
00h	acyclic synchronous, the module replies to each SYNC message
01h - F0h (1 - 240 dez)	cyclic synchronous, the module replies to each nth SYNC message
FFh (255 dez)	event-driven, PDO is transmitted at expiration of the Event Timer

Table 19: Setup of Transmission Type

The transmit PDO 1 transmits an identifyer with 8 byte user data. The content of the message is copied from object 7130h, subindex 1 to 4.

8.7.3 Transmit PDO 2 Parameter

Index 1801h

The object at Index 1801h defines the communication parameters of the transmit PDO 2.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	05h
1	Unsigned32	rw	COB-ID for PDO	280h+Node-ID
2	Unsigned8	rw	Transmission Type	01h
5	Unsigned16	rw	Event Timer	0000h

The object is read and write. Only sub-indices 0 to 2 and 5 are supported. Access to other sub-indices will result in an error message.

COB-ID for PDO

Subindex 1 sets the ID which receives the PDO. It is defined as follows:

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,

Table 20: Definition of COB-ID for PDO

To enable the PDO the most significant bit (b31) must be deleted. To disable the PDO the most significant bit must be set. In default setting the PDO active.

Transmission Type

Subindex 2 sets the transmission type of a message.

Transmission Type	Description
00h	acyclic synchronous, the module replies to each SYNC message
01h - F0h (1 - 240 dez)	cyclic synchronous, the module replies to each nth SYNC message
FFh (255 dez)	event-driven, PDO is transmitted at expiration of the Event Timer

Table 21: Setup of Transmission Type

The transmit PDO 2 transmits an identifyer with 8 byte user data. The content of the message is copied from object 7130h, subindex 5 to 6.

8.7.4 Transmit PDO 3 Parameter

Index 1802h

The object at Index 1802h defines the communication parameters of the transmit PDO 3.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	05h
1	Unsigned32	rw	COB-ID for PDO	380h+Node-ID
2	Unsigned8	rw	Transmission Type	01h
5	Unsigned16	rw	Event Timer	0000h

The object is read and write. Only sub-indices 0 to 2 and 5 are supported. Access to other sub-indices will result in an error message.

COB-ID for PDO

Subindex 1 sets the ID which receives the PDO. It is defined as follows:

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,

Table 22: Definition of COB-ID for PDO

To enable the PDO the most significant bit (b31) must be deleted. To disable the PDO the most significant bit must be set. In default setting the PDO active.

Transmission Type

Subindex 2 sets the transmission type of a message.

Transmission Type	Description
00h	acyclic synchronous, the module replies to each SYNC message
01h - F0h (1 - 240 dez)	cyclic synchronous, the module replies to each nth SYNC message
FFh (255 dez)	event-driven, PDO is transmitted at expiration of the Event Timer

Table 23: Setup of Transmission Type

The transmit PDO 3 transmits an identifyer with 8 byte user data. The content of the message is copied from object 7100h, subindex 1 to 4.

8.7.5 Transmit PDO 4 Parameter

Index 1803h

The object at Index 1803h defines the communication parameters of the transmit PDO 4.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	05h
1	Unsigned32	rw	COB-ID for PDO	380h+Node-ID
2	Unsigned8	rw	Transmission Type	01h
5	Unsigned16	rw	Event Timer	0000h

The object is read and write. Only sub-indices 0 to 2 and 5 are supported. Access to other sub-indices will result in an error message.

COB-ID for PDO

Subindex 1 sets the ID which receives the PDO. It is defined as follows:

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,

Table 24: Definition of COB-ID for PDO

To enable the PDO the most significant bit (b31) must be deleted. To disable the PDO the most significant bit must be set. In default setting the PDO active.

Transmission Type

Subindex 2 sets the transmission type of a message.

Transmission Type	Description
00h	acyclic synchronous, the module replies to each SYNC message
01h - F0h (1 - 240 dez)	cyclic synchronous, the module replies to each nth SYNC message
FFh (255 dez)	event-driven, PDO is transmitted at expiration of the Event Timer

Table 25: Setup of Transmission Type

The transmit PDO 4 transmits an identifyer with 8 byte user data. The content of the message is copied from object 7100h, Sub-Index 5 to 6.

8.7.6 Transmit PDO 1 Mapping Parameter

Index 1A00h

The object at Index 1A00h shows the mapping objects of transmit PDO 1.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	04h
1	Unsigned32	ro	Mapped application object 1	7130 0110h
2	Unsigned32	ro	Mapped application object 2	7130 0210h
3	Unsigned32	ro	Mapped application object 3	7130 0310h
4	Unsigned32	ro	Mapped application object 4	7130 0410h

The object is read and write. Only sub-indices 0 to 4 are supported. An access to other sub-indices will result in an error message.

Objects which are transmitted by PDO can be read here. The structure is as follows.

Bit 31 - Bit 16	Bit 15 - Bit 8	Bit 7 - Bit 0
Index	Sub-Index	Länge

Table 26: Structure of transmit PDO Mapping Parameter

8

In PDO 1 sub-indices 1 to 4 of object 7130h are transmitted (siehe “AI Process Value” auf Seite 81).

8.7.7 Transmit PDO 2 Mapping Parameter

Index 1A01h

The object at Index 1A01h shows the mapping objects of transmit PDO 2.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	02h
1	Unsigned32	ro	Mapped application object 1	7130 0510h
2	Unsigned32	ro	Mapped application object 2	7130 0610h

The object is read only. Sub-indices 0 to 2 are supported. Access to other sub-indices will result in an error message.

Objects which are transmitted by PDO can be read here. The structure is as follows.

Bit 31 - Bit 16	Bit 15 - Bit 8	Bit 7 - Bit 0
Index	Sub-Index	Länge

Table 27: Structure of transmit PDO Mapping Parameter

In PDO 2 sub-indices 5 to 6 of object 7130h are transmitted (siehe “AI Process Value” auf Seite 81).

8.7.8 Transmit PDO 3 Mapping Parameter

Index 1A02h

The object at Index 1A02h shows the mapping objects of transmit PDO 3.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	04h
1	Unsigned32	ro	Mapped application object 1	7100 0110h
2	Unsigned32	ro	Mapped application object 2	7100 0210h
3	Unsigned32	ro	Mapped application object 3	7100 0310h
4	Unsigned32	ro	Mapped application object 4	7100 0410h

The object is read only. Sub-indices 0 to 4 are supported. Access to other sub-indices will result in an error message.

Objects which are transmitted by PDO can be read here. The structure is as follows.

Bit 31 - Bit 16	Bit 15 - Bit 8	Bit 7 - Bit 0
Index	Sub-Index	Laenge

Table 28: Structure of transmit PDO Mapping Parameter

In PDO 3 sub-indices 1 to 4 of object 7100h are transmitted (siehe “AI Field Value” auf Seite 75).

8.7.9 Transmit PDO 4 Mapping Parameter

Index 1A03h

The object at Index 1A03h shows the mapping objects of transmit PDO 4.

Sub-Index	Data Type	Access	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	02h
1	Unsigned32	ro	Mapped application object 1	7100 0510h
2	Unsigned32	ro	Mapped application object 2	7100 0610h

The object is read only. Sub-indices 0 to 2 are supported. Access to other sub-indices will result in an error message.

Objects which are transmitted by PDO can be read here. The structure is as follows.

Bit 31 - Bit 16	Bit 15 - Bit 8	Bit 7 - Bit 0
Index	Sub-Index	Laenge

Table 29: Structure of transmit PDO Mapping Parameter

In PDO 4 sub-indices 5 to 6 of object 7100h are transmitted (siehe “AI Field Value” auf Seite 75).

8.7.10 Example of Transmit PDO

In default setting all 4 transmit PDOs are set to transmission type 1 (cyclic, synchronous, each SYNC). Transmission of PDOs is triggered by the SYNC message (object 1005h).

Example: module address 1, send SYNC

ID	DLC
80h	0

As a response you will receive the following message:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
181h	8	Index 7130h, Sub 01h		Index 7130h, Sub 02h		Index 7130h, Sub 03h		Index 7130h, Sub 04h	

ID	DLC	B0	B1	B2	B3
281h	4	Index 7130h, Sub 05h		Index 7130h, Sub 06h	

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
381h	8	Index 7100h, Sub 01h		Index 7100h, Sub 02h		Index 7100h, Sub 03h		Index 7100h, Sub 04h	

ID	DLC	B0	B1	B2	B3
481h	8	Index 7100h, Sub 05h		Index 7100h, Sub 06h	

Sending PDOs is possible in operational mod of the devices only. Independent from the transmission type, the transmit PDOs are sent once after NMT "Start all Nodes".

Example: A device with module address 2 sends a boot-up message, is switched to operational mode via NMT "Start all Nodes" and will then send the first two PDOs at each SYNC.

```
     1)      Rx          0702   1   00
     2)      Tx          0000   2   01 00
     3)      Rx          0182   8   00 00 00 00 00 00 00 00
     4)      Rx          0282   4   00 00 00 00
     5)      Rx          0382   8   00 00 00 00 00 00 00 00
     6)      Rx          0482   4   00 00 00 00
     7)      Tx          0080   0
     8)      Rx          0182   8   00 00 00 00 00 00 00 00
     9)      Rx          0282   4   00 00 00 00
    10)      Rx          0382   8   00 00 00 00 00 00 00 00
    11)      Rx          0482   4   00 00 00 00
    12)      Tx          0080   0
    13)      Rx          0182   8   00 00 00 00 00 00 00 00
    14)      Rx          0282   4   00 00 00 00
    15)      Rx          0382   8   00 00 00 00 00 00 00 00
    16)      Rx          0482   4   00 00 00 00
```

Example 1: Trace: A device is set to operational and sends PDOs

8.8 Synchronization Message

Index 1005h

The object at Index 1005h sets the identifier for the synchronization message (SYNC). Sending of a PDO can be triggered via the SYNC message.

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

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

Example: COB-ID to 10, setup of module address 1

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

As a response you will receive 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 value of the SYNC identifier is 80h. This assigns high priority to SYNC messages on the CAN bus.

8

The set SYNC-ID is not automatically stored in an non-volatile storage. Storage has to be triggered via index 1010h

8.9 Emergency Message

In case of a device error the µCAN module automatically generates an Emergency Message (EMCY). Please note the difference between an SDO error message in case of faulty access to a SDO object and "real" error messages which are sent as Emergency messages. If an error occurs for the first time an error message will be sent. As soon as the error has been removed, another error message will be sent stating error code 0000h.

The identifier of the EMCY message is calculated from the value of the preset module address + 128_d.

An emergency message has the following structure:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
80h+NID	08h	Error Code		ER	Manufacturer Specific Error Field				

The error is classified by the "Error Code".

In field „ER" (error register) of the emergency message the current content of CANopen object 1001h is displayed.

The „Manufacturer Specific Error Field" contains further manufacturer specific information to clearly identify the cause of the error.

8.9.1 Overview of Error Codes

The following Error Codes are supported:

Error Code	*Error Field [hex] [b3 b4 b5 b6 b7]*	*Name*
0000h	00 00 00 00 00	Error eliminated or no error
5030h	xx xx 00 00 00	"Sensor fault"
8100h	xx 00 00 00 00	CAN Controller is in "Warning" condition
8110h	00 00 00 00 00	CAN Controller is in "Overrun" condition, too many messages
8120h	xx 00 00 00 00	CAN Controller is in"Error Passive" condition
8130h	00 00 00 00 00	Heartbeat / Node-Guarding Event
8140h	00 00 00 00 00	Recover from Bus-Off
8150h	00 00 00 00 00	Identifier collision (Transmit identifier has been received)

Table 30: Error codes of Emergency Message

The "Error Field" column indicates whether or not the „Manufacturer Specific Error Field" is in use. The data in this field may be used to provide more detailed information on the error code indicated.

The µCAN module stores an error history containing all sent emergency messages in object 1003h within the CANopen object dictionary.

8

Error Field of "5030h Error Code"

5030h Error Code

If an error occurs on a measuring channel, it will be masked with the value of object 6150h (siehe refer to “AI Status” on page 70) and displayed in this error field.

B3	*B4*	*B5*	*B6*	*B7*
xxh	xxh	00h	00h	00h
Number of measuring channel	Error code	-	-	-

Table 31: Manufacturer specific error field of the 5030h Error Code

Example: Emergency message with the following content:
The sent emergency message indicates an error at a sensor in the µCAN module with the module address 2 at measuring channel 3. The displayed error code 05h indicates a "negative overload"

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
82h	8	50h	30h	08h	03h	05h	00h	00h	00h

and "break of sensor" condition at the sensor.

Error Field of "8100h Error Code" and "8120h Error Code"

8100h and 8120h Error Code

If an EMCY message is sent with an error code as mentioned above, the internal value of the CAN controller will be displayed in data byte 3. The following values are possible.

B3	Description
00h	no error
10h	bit error
20h	stuffing error
30h	form error
40h	CRC error
50h	ACK error

Table 32: Error codes of 8100h and 8120h Error Code

8

9. Technical Data

Power Supply	
Supply Voltage	9 .. 36V DC, reverse voltage protected
Power consumtion	1,86 W (155 mA @ 12 V DC) 1,92 W (80 mA @ 24 V DC) 2,08 W (65 mA @ 32 V DC)
Connection	Screw terminals at COMBICON plug

CAN Bus	
Baud rates	50 kBit/s .. 1 MBit/s
State of Bus	active node
Protocol	CANopen acc. to CiA 301 V4.02, CiA 404 V1.02
Connection	Screw terminals at COMBICON plug

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 EN 61000-4-3
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 55011, class A

Measurement technology	
Operating temperatures	-40°C bis +85°C
Signal type	± 10 V 0(4)..20,000 mA
Resolution	16 bit
Sampling rate	80 Hz on each measuring channel

Housing	
Plastic	Polyamid
Temperature resistance	-40°C to +105° C
Combustibility class	V0 (acc. to UL 94)
Mounting	on DIN rail/supporting rail TS 35 according to DIN EN 50022
Dimensions	128,8 * 22,5 * 102 mm (T * B * H)
Weight	approx. 150 g
Protection class	IP20

Appendix

Voltage Supply	
Operating voltage	8 .. 60 V DC, reverse voltage protected
Power consumption	1,5 W (60 mA @ 24 V DC) no load
Potential isolation	fieldbus/contol voltage: 500 Veff
Connection	spring-cage connectors

9.1 Drilling Template

9.2 Index

Mechanics	
Housing	Aluminium die cast
Dimensions	125 * 80 * 57 mm (L * W * H)
Weight	540 g
Protection class	IP65

2

Digital Inputs	
Input impedance	24,2 kOhm
Input Low	$U_{in} < 0,4 * U_{PWR}$
Input High	$U_{in} > 0,6 * U_{PWR}$

Digital Outputs	
Type	Highside Power-MOSFET
Maximum output voltage	50 V
Maximum output current	1,4 A
Short circuit detection	ab 5 A
Max. total current	6 A

MicroControl übernimmt keine Haftung für die Übereinstimmung des Inhalts der Bedienungsanleitung mit den jeweiligen geltenden gesetzlichen Vorschriften, ebensowenig für Fehler und technische Angaben.

Diese Bedienungsanleitung ist urheberrechtlich geschützt. Alle Rechte, wie Übersetzung, Nachdruck und Vervielfältigung auch in Auszügen, behält sich MicroControl GmbH & Co. KG vor.

© 1999 - 2011 MicroControl GmbH & Co. KG, Troisdorf

MicroControl GmbH & Co. KG
Lindlaustr. 2c
D-53842 Troisdorf
Fon: +49 / 2241 / 25 65 9 - 0
Fax: +49 / 2241 / 25 65 9 - 11
http://www.microcontrol.net