MicroControl

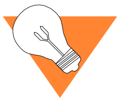
Systemhaus für Automatisierung

µCAN.4.sg-BOX

Manual Strain Gauge Acquisition Module

Explanation of Symbols

To facilitate reading and understanding of the document several symbols will be seen on the left side of the manual.



This symbol marks a paragraph which contains useful information for working with the device or which gives useful hints.

This symbol marks a paragraph which explains possible sources of danger which might cause damage to the system or operating personnel.

Keywords

Important keywords appear in the border column to facilitate navigation through the text.

MicroControl GmbH & Co. KG
Junkersring 23
D-53844 Troisdorf
Fon: +49 / 2241 / 25 65 9 - 0
Fax: +49 / 2241 / 25 65 9 - 11
http://www.microcontrol.net

1. Safety Regulations

Please read the following chapter in any case to ensure safe handling of electrical devices.

1.1 General Safety Regulations

This paragraph contains important information about the usage of µCAN modules. It was written for personnel which is qualified and trained on the use of 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 are familiar with the handling of the machine.
- You are the operator of the machines 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 service and are trained on repairing automated machines. In addition, you are trained in 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 during set-up of the devices.

If devices are to be integrated into stationary machines without a mains switch for all phases or fuses, this equipment must be installed 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 the devices are supplied by 24V DC auxiliary supply, please ensure isolation of the low-voltage lines from other voltages.

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

Devices or machines used in 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 a mechanical locking device.

2. Operation of µCAN.4.sg-BOX

2.1 Overview

The µCAN.4.sg-BOX is ideally suitable for acquisition and linearization of analogue standard signals from full bridge strain gauges which are transmitted as voltage [mV] via CAN bus. The measured values may be scaled and the measurement units may be changed according to your requirements.

Fig.1: µCAN.4.sg-BOX Module for acquisition of analogue signals

While evaluating the signals, the module automatically monitors the limit values and sends a CAN message if deviations are detected.

Separated from the main system, the µCAN modules are supposed to acquire data at the source which will reduce wiring and costs.

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

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

All these requirements are fulfilled by the µCAN.4.sg-BOX module. The µCAN.4.sg-BOX runs on the standard field bus CAN.

Typical applications of the µCAN.4.sg-BOX are industrial automation, automotive technology, food industry and environmental technology.

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

CANopen®

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

Space saving and compact

The µCAN.4.sg-BOX is ideally suited to work in harsh industrial environments due to its rugged casing in protection class IP 66. The compact and space-saving casing offers the opportunity to apply the module virtually everywhere.

Inexpensive and Service friendly

The quick and easy integration of the µCAN.4.sg-BOX in your application reduces the development effort. Costs for material and personnel are reduced to a minimum. The easy installation of the module facilitates maintenance and replacement.

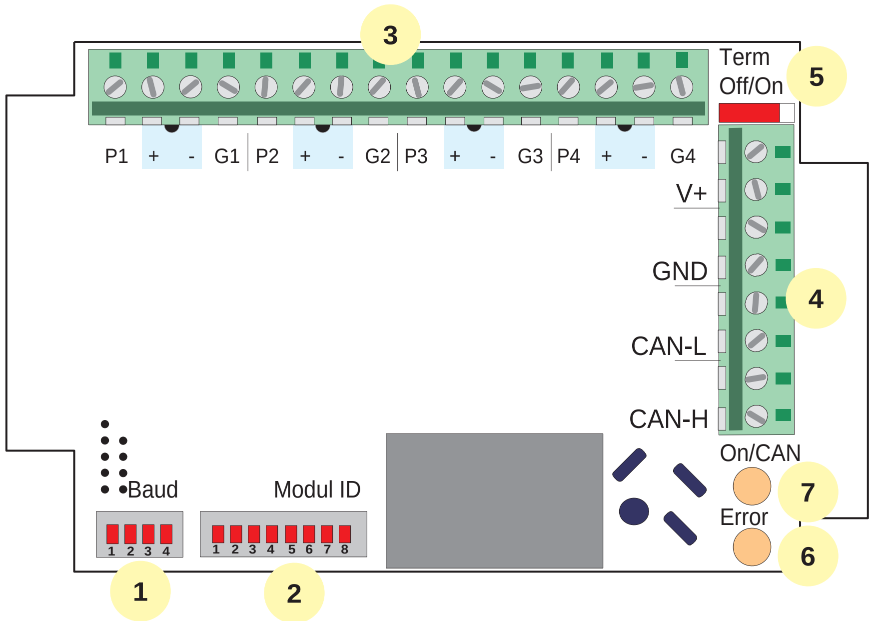
3. Project Planning

The chapter Project Planning contains information which is important for systems engineers and users of the µCAN.4.sg-BOX. This information include case dimensions and optimum conditions of use.

3.1 Functional units of the µCAN module

3

The following illustration shows the different functional groups of the µCAN module. This illustrates the structure and position of the settings and application options available.



1: Bit rate setting
2: Module ID setting
3: Terminal block analogue signals
4: Terminal block power supply / CAN
5: Termination switch
6: Bi-colour LED for module status
7: Bi-colour LED for network status

Fig.2: Overview of functional units

3.2 General Description

The µCAN.4.sg-BOX is a rugged device for acquisition and linearization of analogue signals designed for industrial purposes.
The following signal types can be acquired:
- full bridge strain gauge with min. 300 Ohm impedance

The strain gauge can be sourced directly from the module with a 4,096V reference voltage.

Further signal ranges are optionally available.

This µCAN module acquires and linearizes analogue signals which are transmitted as voltage or current values via CAN bus.
If the values fall below or exceed pre-set limits, the user will receive an alarm message, siehe “Monitoring Limit Values” auf Seite 35.

The µCAN module is protected by a casing in protection class IP 66, and so the µCAN.4.sg-BOX well-suited to be installed outside a switching cabinet.
The idea behind installing the module outside a switching cabinet is to acquire measurement data at the source, long signal and extension lines are not necessary. In addition, electro magnetic interference with long signal lines will be avoided.

The µCAN module can be supplied with voltage of 9..36V DC. The µCAN.4.sg-BOX should be connected to power supply and CAN bus via four core wires, thus reducing wiring to a minimum. Adequate CAN cables are also available.

3.3 Maximum System Layout

For an executable bus system at least one network manager must exist on the CAN bus. This network manager may be a PLC or PC equipped with an adequate CAN card. Every µCAN module is and active node.

A bus line can **logically be** controlled from a maximum of 127 modules. Each module is assigned a module address (node ID, NID) which can be set via a DIP switch on the µCAN module. The CAN bus can be connected through the individual µCAN modules.

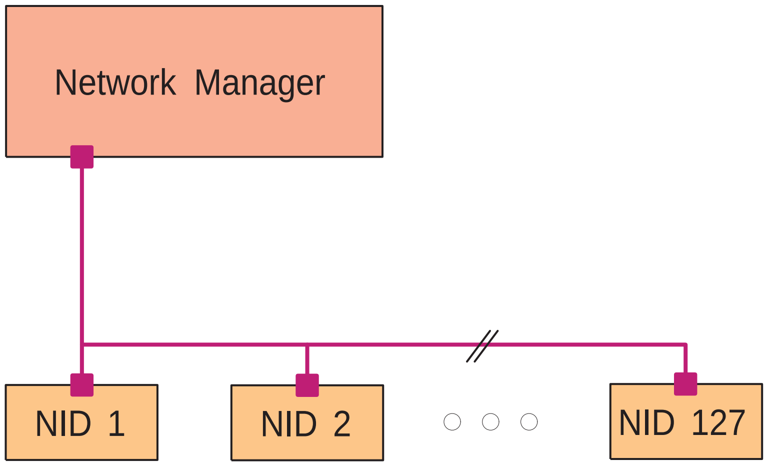


Fig.3: Maximum system layout

The maximum cable lengths depend on the selected bit rate and are listed in the table below. The values are recommended by CAN in Automation (CiA) and can be realized with the µCAN.4.sg-BOX.

Bit rate	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

Table 1: Bit rates in relation to cable length

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

3.4 Case Dimensions

The case dimensions of the µCAN.4.sg-BOX are given in the drawing below. Due to its casing in protection class IP 66 the µCAN module can be installed virtually everywhere. The µCAN modules can either be installed directly on the machine or inside a switching cabinet. Please check the technical data section for detailed information about the maximum environment conditions of the µCAN module.

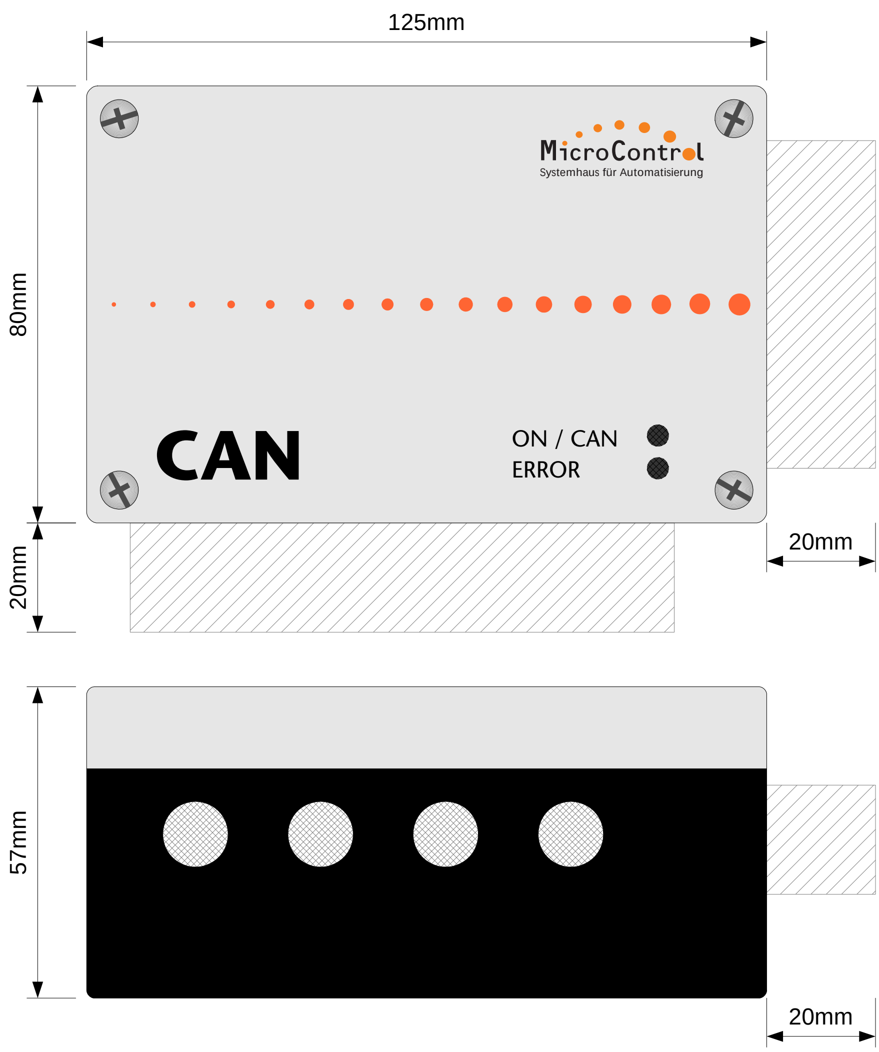
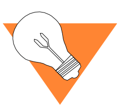


Fig.4: Case dimensions

The hatched areas mark the space required for plugs or cable glands. As the kind of connection differs according to the module version, the dimensions of connectors in the illustration are only approximate values.



For further information about the casing, please refer to chapter “Technical Data” page 97.

3

4. Installation and Dismantling

4.1 General Information

Installation

We recommend to mount the µCAN modules on a mounting plate of at least 2 mm or directly on the machine. The modules are fixed by two M4 screws which fit in the holes in the bottom part of the casing.

Power Supply

Power can be supplied via a two core wire which is fixed to the respective terminals. However, application of four core wires is more convenient as the CAN bus can use the same connection.

PE must be supplied via a ground or earth connector outside the casing (see illustration 5, “Supply of the PE protective conductor”). PE supply within the casing is not allowed due to EMC regulations.

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

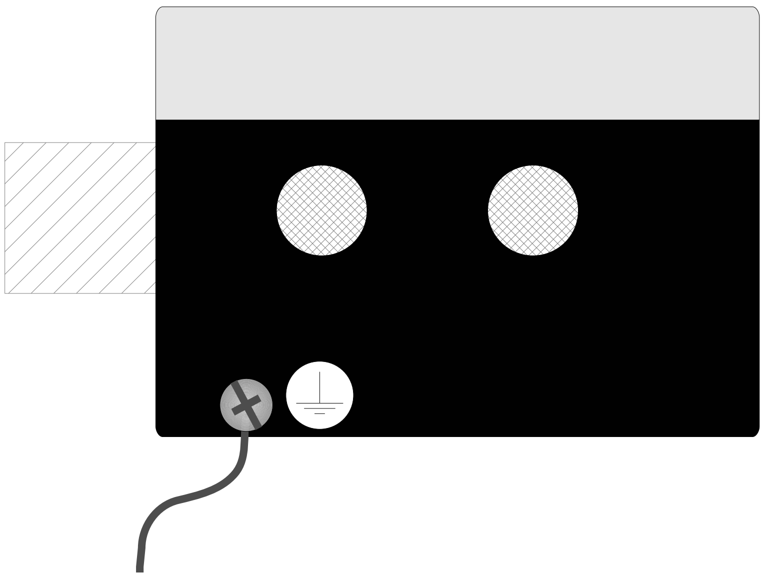


Fig.5: Supply of the PE protective conductor

When operating the µCAN.4.sg-BOX the casing must be closed.

4.2 Installation

The casing is fixed via two separate screw channels outside the sealed compartment of the module.

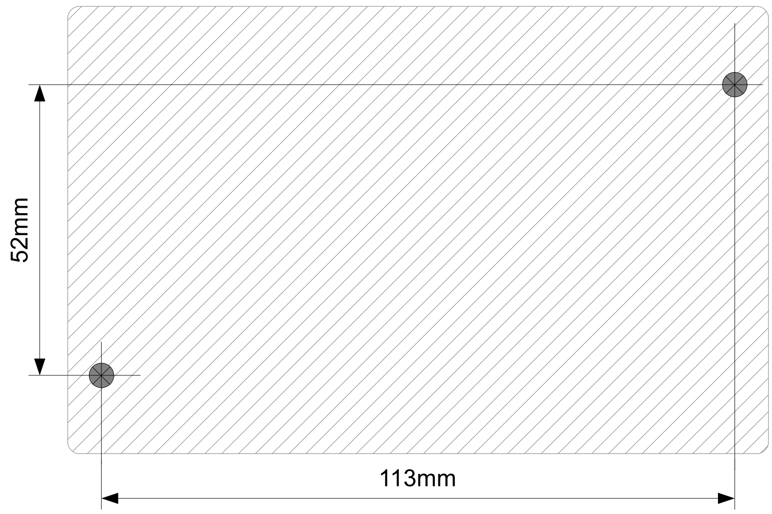


Fig.6: Position of the fixing points of the casing

When installing several µCAN modules, please make sure to leave enough space between the modules for cable glands or plugs.

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

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

4.3 Dismantling

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

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

4

5. Installation

5.1 Potential Basics

The potential environment of our µCAN.4.sg-BOX modules is characterised by the following features:

- The CAN bus potential is isolated from the power supply.
- The individual µCAN modules are not isolated from the power supply.
- All µCAN modules can be supplied separately.
- The analogue inputs 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 with regard to the limit values stipulated by law. The µCAN modules are tested through an officially recognized EMC laboratory. However, an EMC plan for the system should be set up in order to exclude potential noise sources.

In automation and measurement technology noise signals can couple in different ways. Depending on that way (guided wave propagation or non-guided wave propagation) and the distance between the noise source and the µCAN modules, 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) as well as different potentials of casings of components and the 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, parallel 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 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 observed.

5.2.1 Earthing of inactive Metal Parts

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

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

The µCAN modules are earthed via a cable lug which is connected to the PE terminal outside the modules. The protective conductor must not lead into the casing of the modules.

µCAN modules which are not equipped with metal or aluminium casing do not have to be connected to a common earthing potential.

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 protective 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 these can be easily damaged on assembly and, therefore, do not guarantee proper shielding.

Cable Layout

In general, the cable shield should be earthed on both ends. The cable shield should only be earthed on one end if an attenuation is necessary in the low frequency range. In addition, earthing on both ends is not possible for certain measurement sensors. In these cases, earthing 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 sensors).

The shield of the CAN bus cable must not lead inside casing of the µCAN module. 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 terminals.

5.3 General Information on Wiring

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

High-voltage power lines and data or signal lines should be arranged in separate 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 lines 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 necessary:

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

5.4 CAN Cable

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

Cable characteristics	Value
Impedance	108 - 132 Ohms (nom. 120 Ohms)
Specific impedance	70 mOhms/meter
Specific signal delay	5 ns/meter

Table 2: Characteristics CAN cable

The bus cable is connected to the µCAN modules via terminal block inside the casing. For terminal configuration please refer to the respective chapter of this manual.

STOP

Please ensure that potential of signal lines are not confused, as this will prevent communication on the bus.

5.5 Power Supply

The µCAN.4.sg-BOX module is designed for industrial applications. By using a DC/DC converter the CAN bus is galvanically isolated from the power supply. The supply voltage may vary within a range from 9..36V. The input of the power supply is protected against reverse polarity.

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

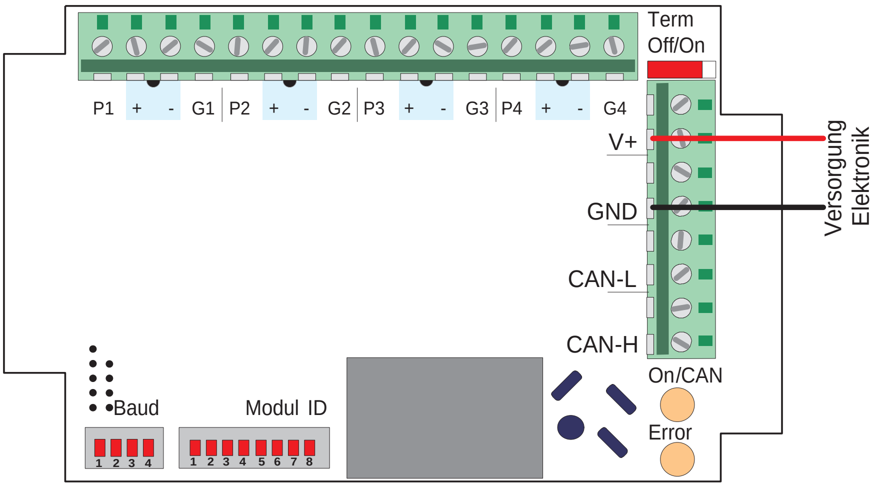
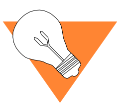


Fig.7: Connection of the power supply

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



A shielding must not lead into the µCAN module or connected to one of the terminals. The 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 via a two core wire.

To avoid electromagnetic interference, please ensure that the CAN bus line does not cross the signallines.

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

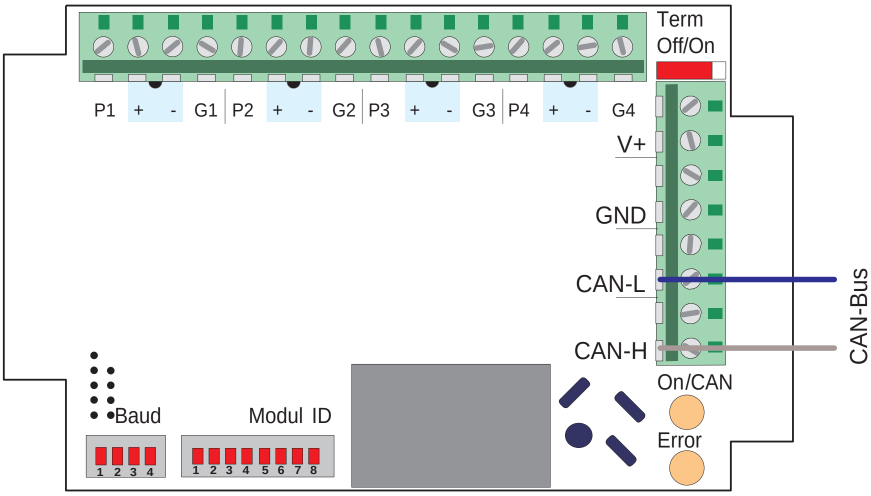
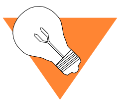
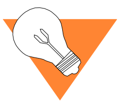


Fig.8: Connection of CAN bus

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. The shielding must be connected to the appropriate potential via special connectors outside the casing.



If you use a 9-pin Sub-D connector, the high potential has to be connected to pin 7 and the low potential 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 bottom left of the board and is labelled "Module ID". For setting the module address (node ID) we recommend using a small screw driver.

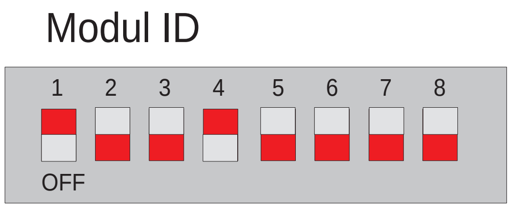


Fig.9: Set-up of module address (address 9 is shown here)

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 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 module address until a new module address is selected and a reset is performed.

If all "Module ID" switches are in OFF position and the DIP switches of the bit rate are switched to OFF as well, the µCAN.4.sg-BOX module will be started in LSS mode.

The DIP switch 8 must always be in OFF position.

If the position of the switches are not configured correctly, the µCAN module will not start. This will be indicated by the "Error" LED (refer to see “Module state” page 40).

5.8 Bit Rates

The bit rates of the µCAN modules are selected via a 4-pole 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 bit rate you should use a small-size screw driver.

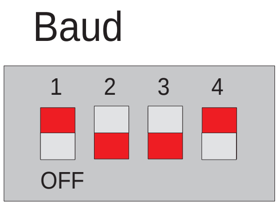


Fig.10: Set-up of bit rate (1 MBit/s is shown here)

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

Bit rate	**DIP switch position**			
	1	**2**	**3**	**4**
Autobit / LSS	0	0	0	0
Autobit	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

Table 3: Configuration of the bit rate

A bit rate of 10 kBit/s or 20 kBit/s is not supported by the µCAN.4.sg-BOX. LSS will only be used if all module ID switches are switched to OFF position.

If the module is set to Autobit, the bit rate on the CAN bus will automatically be detected.

If the module is set to LSS mode, the bit rate and module address stored in the module will be applied.

If an inadmissible bit rate has been set on the module, this will be indicated via the "Error" LED (see “Diagnosis” page 37).

5.9 Termination

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

To terminate a µCAN.4.sg-BOX set the DIP switch named "Term" from position Term "Off" to position Term "On" by using a small screw driver.

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

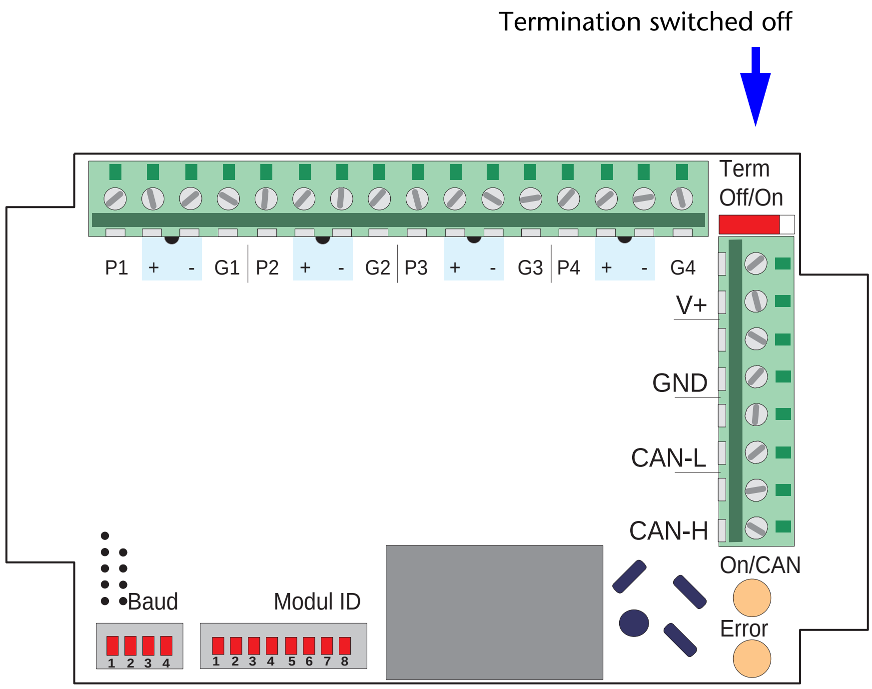


Fig.11: Switching of termination

In the illustration the termination is switched off. This µCAN module is used as a T-piece on a CAN bus line. Therefore, the line has to be terminated through another µCAN module with 120 Ohms.

6. Analogue Inputs

This part of the manual illustrates the various kinds of signals which may be connected to the µCAN.4.sg-BOX module. Again it is important to observe the basic rules of EMC wiring. Only proper, EMC-compliant wiring ensures smooth operation of the modules.

The µCAN.4.sg-BOX module is equipped with four analogue measurement inputs which are numbered in ascending order. The terminal named P1 is assigned measurement channel 1. In the same way, the following three terminals named + , - and G1 are assigned the same measurement channel. The last measurement channel (channel 4) is named P4, +, - and G4.

To connect the signal lines the µCAN modules must be disconnected from power supply to avoid destruction of electronic components.

The input impedance of the µCAN.4.sg-BOX will be dimensioned as seen in the follwing drawing.

Strain gauge measurement

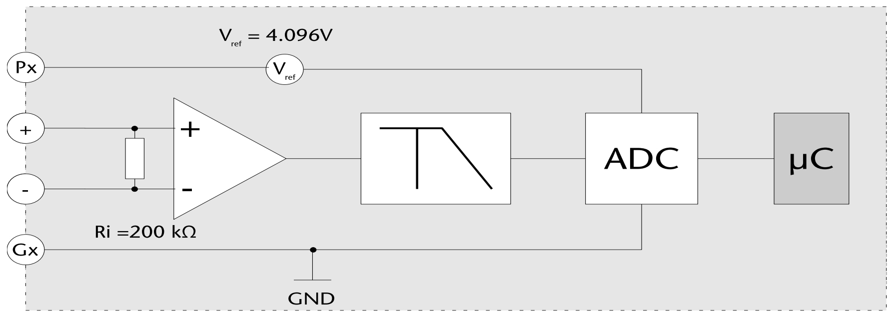


Fig.12: Input settings for voltage measurement

6.1 Linking of Signal Wires

When connecting wires for evaluating standard signals, please make sure that are linked only with terminals + and - of the respective channel. Measurement channels must not be linked with each other.

The positive signal of the respective measurement zone has to be linked with terminal "+" of the selected channel, the negative signal has to be linked with terminal "-".

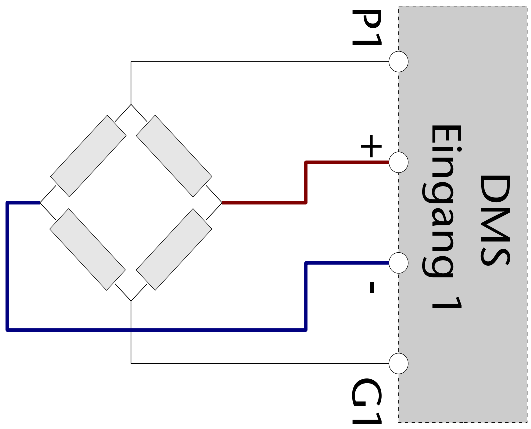


Fig.13: Connection of signal lines

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

Make sure that the signals do not exceed the stipulated limit values of ±60 mV. Exceeding the limit values may destroy the electronic components of the module.

6.2 Monitoring Limit Values

Depending on the version of the the µCAN.4.sg-BOX will automatically checked whether the measurement signals are within the stipulated limit values. The µCAN.4.sg-BOX is available in two different version with 16-bit resolution or 24-bit resolution.

Only in the 24-bit version the measuring signals are monitored. The following table shows the respective limit values.

Sensor type	Limit value measurement range	Signal limit value
Strain gauge	- 55,00000 mV V +55,00000 mV	- 60,00000 mV +60,00000 mV

Table 4: Overview of monitored limit values

Exceeding the limit values of the measurement range will set the respective status values of the measurement channel (see “AI State” page 64).

Only if the stipulated signal limit values are exceeded, an emergency message (see “Emergency Message” page 85) will be sent, the flash code of the "ERROR" LED (see “Module state” page 40) will be adjusted and the status of the measurement channel will be reset.

Moreover, an invalid measurement value 8888h = -30584d (signed) = 34952d (unsigned) will be displayed.

The error will remain until the measurement value falls below the signal limit value.

The current status of the respective measurement input may be read via the object 6150h (see “AI State” page 64)

6

7. Diagnosis

All µCAN modules are equipped with LEDs for indicating the module state and error conditions.

The µCAN.4.sg-BOX is equipped with two bi-colour LEDs (green/red) named "On/CAN" (network status) and "Error" (module status) located on the board.

The designations **ON/CAN** of the network status LED and **ERROR** of the module status LED are printed on the casing.

The following illustration they are marked 1 and 2.

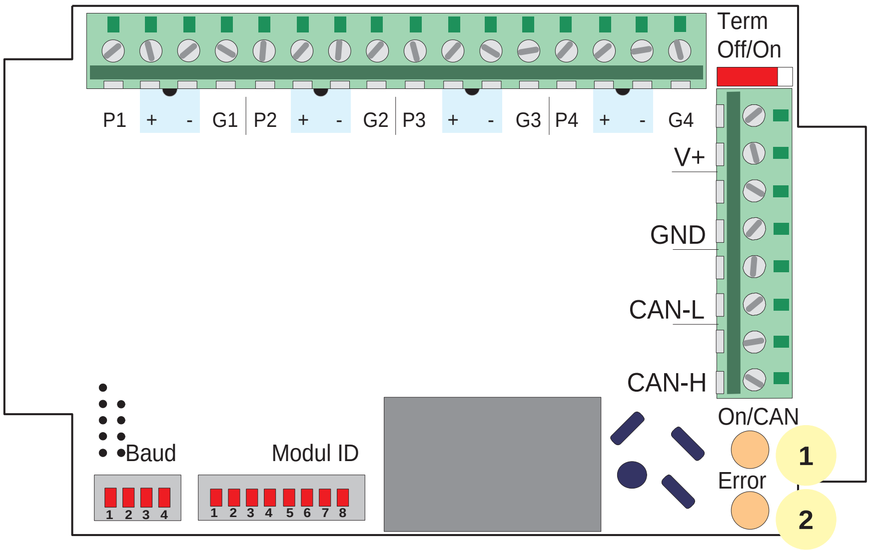


Fig.14: Position of the LEDs on the µCAN module

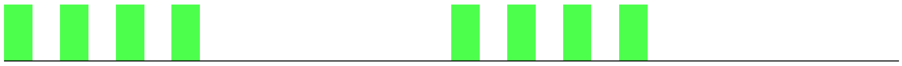
In normal operation all LEDs should have a green colour. A red light, either steady or blinking indicates an error condition.

7.1 Network State

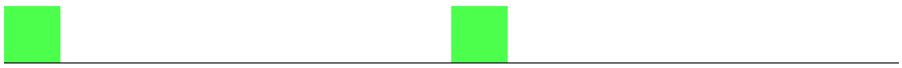
The LED marked "network status" (ON/CAN on the casing) indicates the condition of the CANopen NMT state machine and the error condition of the CAN controller.

7.1.1 Indication CANopen NMT State

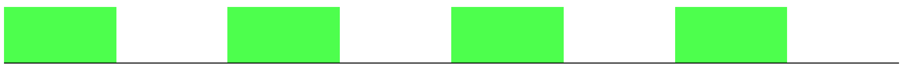
The green light of the LED indicates the CANopen Network Management (NMT) state.



Initialization (Autobit Detection)



NMT state: Device in "Stopped" state

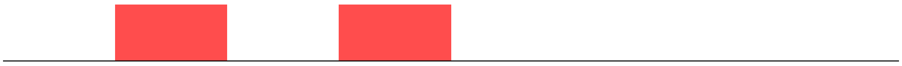


NMT state: Device in "Pre-operational" state

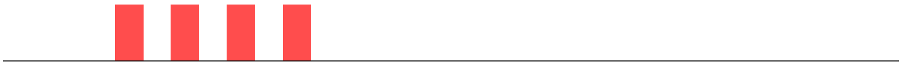
NMT state: Device in "Operational" state

7.1.2 Indication CAN Controller State

The red LED indicates the state of the CAN controller. In error-free condition the red LED is switched off.



CAN state: Controller in "Warning" state

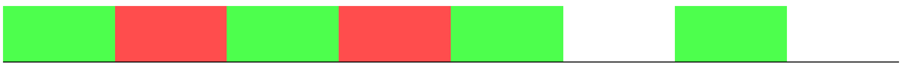


CAN state: Controller in "Error Passive" state

CAN state: Controller in "Bus-Off" state

7.1.3 Combined Indication

The combination of red and green LED indicates the condition of the CAN controller (red LED).



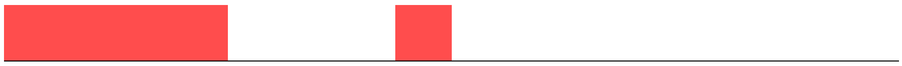
Device in "Pre-operational" state, CAN Controller in "Warning" state

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

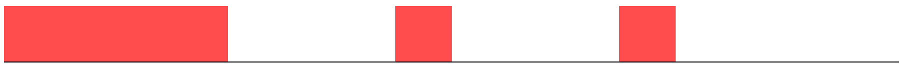
7.2 Module state

The LED marked "module status" (named ERROR on the casing) indicates the state of the module.

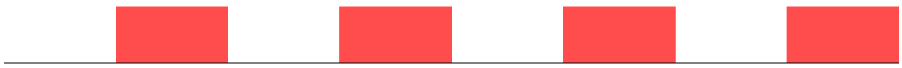
Module state: Operation/Power OK



Module state: Wrong configuration of bit rate



Module state: Wrong configuration of address



Module state: Measured signal exceeds limit values

8. CANopen Protocol

The chapter CANopen Protocol contains the most important information for the user on connecting the µCAN modules to a CANopen manager and putting them into operation. The CANopen manager can be a PC with CAN card, a PLC or even a control unit or actuator.

For detailed information on the CANopen manager, please refer to the respective documentation of the devices in use.

This operating manual provides the currently implemented functions of the µCAN module.

8.1 General Information

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

Object	**COB-ID (hex)**	**Offset (hex)/ Node Address**
Network Management	0x000	--
SYNC	0x080	configurable
EMERGENCY	0x080 + Offset	0x01 .. 0x7F
PDO 1 (Transmit)	0x180 + Offset	0x01 .. 0x7F
PDO 1 (Receive)	0x200 + Offset	0x01 .. 0x7F
PDO 2 (Transmit)	0x280 + Offset	0x01 .. 0x7F
PDO 2 (Receive)	0x300 + Offset	0x01 .. 0x7F
SDO (Transmit)	0x580 + Offset	0x01 .. 0x7F
SDO (Receive)	0x600 + Offset	0x01 .. 0x7F
Heartbeat / Boot-up	0x700 + Offset	0x01 .. 0x7F

Table 5: Distribution of identifiers

The directions (Transmit / Receive) signify the transfer directions from the µCAN.4.sg-BOX module to adjacent devices.

The "Offset" is the selected node address of the device. Please refer to “Module address” page 28.

8.2 Network Management

The µCAN module state (Stop / Pre-Operational / Operational) can be changed by means of Network Management 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 Operational mode. The CAN module is now able 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“ sets the CAN module to Stop mode which will prevent 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“ sets the CAN module 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“ will execute a hardware reset of the CAN module. After reset the CAN module will be set to Pre-Operational mode automatically and will send a „Boot-up Message“.

8.3 SDO Communication

The parameters of the CAN module (object dictionary) are accessed via an SDO channel (Service Data Object). An SDO message is structured as follows:

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)	Functions
22_h	60_h	write, size not defined
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 bytes
40_h	42_h	read, size not defined
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 bytes

Table 6: Command for SDO expedited message

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

STOP

The minimum time delay between two successive SDO messages must not be lower than 10ms.

8.3.1 SDO Abort Protocol

In case of an erroneous access to indices the system will send an abort code. The abort code is always structured as follows:

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 the sub-index refer to the ID which has been accessed erroneously.

The error messages may have the following contents:

Abort code	Description
0504 0001h	Client / server command specifier unknown / not valid
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-indes does not exist in the object dictionary

Table 7: SDO Abort Protocol

8.4 Object Dictionary

This chapter describes the objects implemented in the µCAN.4.sg-BOX 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.4.sg-BOX module are listed in an "Electronic Data Sheet" (EDS). This EDS file may be downloaded from the MicroControl Homepage.

8.4.1 Communications Profile

The µCAN.4.sg-BOX module comprises the following objects of the communications profile CiA 301:

Index	Name
1000h	Device Profile
1001h	Error Register
1002h	Manufacturer State
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

Table 8: Supported objects of the communications profile

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

Table 8: Supported objects of the communications profile

Device Profile

Index 1000h

Index 1000h contains the device profile.

Sub-index	Data type	Access	Description	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 parameters, module address = 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.4.sg-BOX 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 = 2d (Additional Information)

Error Register

Index 1001h

Index 1001h contains the error register of the device.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Error Register	00h

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

Example: read parameters, module address = 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 the module will send the state 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: caused by an error in current measurements.
04h	00h	00h	00h	Voltage Error: caused by an error in voltage measurement.
10h	00h	00h	00h	Communications Error: caused by interference of communications on the CAN bus. A solution to the cause of error is explained in chapter “Emergency Message” page 91.

Table 9: Supported error types in the error register

8

Manufacturer State Register

Index 1002h

Index 1002h contains the state register of the device.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Manufacturer State Register	00h

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

Example: read parameters, module address = 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 the module will send the state of the device.

This register displays state information on the AD converters and the EEPROM.

Different bits may be set in the register which are explained in the following table:

B4	B5	B6	B7	Description
x1h	xxh	xxh	xxh	EEPROM error: an error in communication with the EEPROM has been detected.
x2h	xxh	xxh	xxh	EEPROM error: an error occurred while writing the data.
1xh	xxh	xxh	xxh	Error AD converter 1: communication with AD converter 1 could not be established.
2xh	xxh	xxh	xxh	AD converter 1 stopped: AD converter 1 (channel 1 and 2) has been stopped.
xxh	x1h	xxh	xxh	Error AD converter 2: communication with AD converter 2 could not be established.
xxh	x2h	xxh	xxh	AD converter 2 stopped: AD converter 2 (channel 3 and 4) has been stopped.

Table 10: State information in the manufacturer state register

8

Error Register

Index 1003

Via index 1003h the user may access the error history. Sub-indices 1...4 contain the last 4 errors occurred.

Sub-index	Data type	Access	Description	Default value
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. A writing access to sub-index 0 will delete the error register.

Example: read parameters, module address = 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 send the the state of the error register of the 3rd to last error of the device.

Device Name

Index 1008

Index 1008h contains the device name.

Sub-index	Data type	Access	Description	Default value
0	Visible String	ro	Device name	mCAN.4.sg-BOX

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

Hardware Version

Index 1009h

Index 1009h contains the hardware version.

Sub-index	Data type	Access	Description	Default value
0	Visible String	ro	Hardware version	4.02

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

Software Version

Index 100Ah

Index 100Ah contains the software version.

Sub-index	Data type	Access	Description	Default value
0	Visible String	ro	Software version	4.00

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

Store Parameters

Index 1010h

Index 1010h may trigger storing of parameters in a non-volatile memory.

Sub-index	Data type	Access	Description	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

Storage is triggered by sending index 1010h with the message "save" (in ASCII) on sub index 1. Therefore, the message is structured as follows:

Example: read 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 module will send:

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

8

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

Load default parameter settings

Index 1011h

Via Index 1011h the default parameter settings of the device may be loaded.

Sub-index	Data type	Access	Description	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

Loading of default parameter settings is triggered by sending index 1011h with the message "load" (in ASCII) on sub index 1. Therefore, the message is structured as follows:

Example: Load default settings, 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 module will send:

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

COB-ID emergency message

Index 1014h

This object defines the COB-ID for the emergency messages (EMCY).

Sub-index	Data type	Access	Description	Default value
0	Unsigned32	rw	COB-ID EMCY	80h + NID

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

Module Identity

Index 1018h

Index 1018h contains the identity object of the device.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Unsigned32	ro	Vendor ID	0000 000Eh
2	Unsigned32	ro	Product Code	0012 F779h
3	Unsigned32	ro	Revision Number	0302 0400h
4	Unsigned32	ro	Serial Number	-

Sub-indices 0 to 4 are supported. Access to other sub-indices will result in an error message.

Vendor ID

The vendor ID is a unique manufacturer specific identification number which is centrally assigned and managed by the CAN in Automation (CiA). 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

In response to a query you will receive the serial number of the 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 automatically be switched to pre-operational mode. Via index 1029h this error behaviour may be changed.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	number of entries	01h
1	Unsigned8	rw	Communication error	00h

The following values are admissible:

Value	Description
00h	Standard behaviour, change over to pre-operational
01h	The current NMT mode is not changed
02h	Change over to NMT mode "stopped"

The following device failures are considered:
- error in node guarding
- error in heartbeat

NMT Startup

Index 1F80h

This object defined the NMT startup behaviour after the device has been switched on.

Sub-index	Data type	Access	Description	Default value
0	Unsigned32	rw	NMT Startup	0000 0000h

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

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

8.4.2 Manufacturer specific objects

The µCAN.4.sg-BOX modules contain 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

There are some additional manufacturer objects for configuration of the analogue frontend of the module.

Index	Name
5380h	Minimal Value Scaling
5381h	Maximum Value Scaling

Table 12: Analogue configuration specific objects

Customer Data

Index 2010h

Via the index 2010h up to 8 words may be stored in the EEPROM of the µCAN module.

Sub-index	Data type	Access	Description	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	-

Sub-indices 0 to 8 are supported. Access to other sub-indices will result in an error message.

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

COB-ID Storage

Index 201Ah

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

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	rw	COB-ID Storage	00h

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

8

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

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

PDO Data Format

Index 2E00h

This object stipulates which format, Intel (Little-Endian) or Motorola (Big-Endian), PDO data will have.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	rw	PDO Data Format	00h

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

Value	Description
00h	PDO data are sent in Intel formate
01h	PDO data are sent in Motorola formate

Change over to Motorola formate for sending PDO data does not comply with the CANopen specifications.

Disable BootUp Message

Index 2E10h

This object stipulates whether the µCAN.4.sg-BOX module sends a BootUp Message or not after switching on or Reset Node.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	rw	Disable BootUp Message	00h

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

Value	Description
00h	BootUp Message is sent after switching on or Reset Node
01h	The BootUp Message is not sent

Switching off the Bootup Message does not comply with CANopen specifications.

Bus Statistics

Index 2E22h

Via the index 2E22h the bus statistics may be retrieved.

Sub-index	Data type	Access	Description	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	-

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

The number of received messages is recorded in sub-index 1, the number of sent messages is in sub-index 2. The number of CAN error frames is recorded in sub-index 3.

8

8.4.3 Device profile CiA 404

The µCAN.4.sg-BOX module contains the following objects of the device profile CiA 404:

Index	Name
6110h	AI Sensor Type
6112h	AI Operating Mode
6126h	AI Scale Factor
6127h	AI Sacle Offset
6131h	AI Physical Unit Process Value
6132h	AI Decimal Digits Process Value
6150h	AI State
61A0h	AI Filter Type
61A1h	AI Filter Constant
9100h	AI Field Value
9130h	AI Process Value

Table 13: Supported objects of the device profile

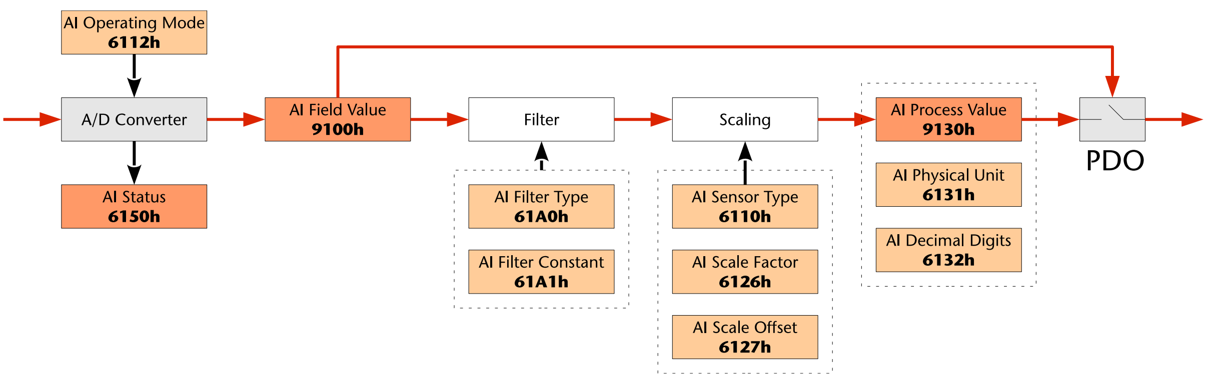


Fig.15: Block diagram of the supported functions of an analogue input

The µCAN.4.sg-BOX supports some additional features for customer specific scaling of measuring values. Please refer to “User-defined Scaling” page 94.

AI Sensor Type

Index 6110h

Via the index 6110h the sensor type may be set and queried.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Unsigned16	ro	AI Sensor Type of Channel 1	0046h
2	Unsigned16	ro	AI Sensor Type of Channel 2	0046h
3	Unsigned16	ro	AI Sensor Type of Channel 3	0046h
4	Unsigned16	ro	AI Sensor Type of Channel 4	0046h

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

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

Value	*Sensor type*
46h	STRAIN_GAUGE

Table 14: Supported sensors

Example: read sensor type of measurement channel 1 (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 the module will send:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
581h	8	4Bh	10h	61h	01h	46h	00h	00h	00h

In this example, the value of the sensor type is 46h (byte 4). i.e. the current sensor type is set to acquire data in units of +/-50 mV.

8

AI Operating Mode

Index 6112h

Via index 6112h the operating mode of the measurement inputs is set and retrieved. Via operating mode measurement channels may individually be switched on or off.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
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

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

To switch on a channel "1" has to be sent to the module via the respective sub-index, to switch off a channel "0" has to be sent.

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.4.sg-BOX module 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

Channel 3 is now switched off and will not be evaluated. If an error occurred at this channel, it is now reset. If you retrieve a measurement value from this channel, you will receive the value 0.

The set operating mode is not automatically saved in a non-volatile memory. Storage has to be triggered via index 1010h (see “Store Parameters” page 54).

AI Physical Unit Process Value

Index 6131h

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

Sub-index	Data type	Access	Description	Default value [mV]
0	Unsigned8	ro	Largest sub-index	04h
1	Unsigned32	rw	AI Physical Unit PV of Channel 1	FD260000h
2	Unsigned32	rw	AI Physical Unit PV of Channel 2	FD260000h
3	Unsigned32	rw	AI Physical Unit PV of Channel 3	FD260000h
4	Unsigned32	rw	AI Physical Unit PV of Channel 4	FD260000h

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

Example: read physical measurement unit of measurement 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.4.sg-BOX module will send the following message:

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

The data contains the value FD260000h which corresponds to the measurement unit in mV.

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

AI Decimal Digits Process Value

Index 6132h

Index 6132h is read only. It states the number of decimal digits of the process values.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Unsigned8	rw	AI Decimal Digits PV of Channel 1	05h
2	Unsigned8	rw	AI Decimal Digits PV of Channel 2	05h
3	Unsigned8	rw	AI Decimal Digits PV of Channel 3	05h
4	Unsigned8	rw	AI Decimal Digits PV of Channel 4	05h

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

Example: Read number of decimal digits of measurement 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.4.sg-BOX module will send the following message:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
582h	8	40h	32h	61h	03h	05h	00h	00h	00h

Data byte 4 contains the value 05h, i.e. process values will have fife decimal digits.

AI State

Index 6150h

Via index 6150h the state of the measurement channel may be read.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Unsigned8	ro	AI Status of Channel 1	00h
2	Unsigned8	ro	AI Status of Channel 2	00h
3	Unsigned8	ro	AI Status of Channel 3	00h
4	Unsigned8	ro	AI Status of Channel 4	00h

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

Whether the measurement values reach or go beyond the limit values may be stipulated via the value in the state register. The values are defined as follows:

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

Table 15: Possible values of the state of the measurement channel

Example: Read state of measurement 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 the module will send:

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 has occurred and the measurement values at this channel are not valid.

The limit values of the measurement inputs are stated in chapter “Monitoring Limit Values” page 35.

AI Filter Type

Index 61A0h

Via index 61A0h the filter type of the measurement channel may be set or read.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Unsigned8	rw	AI Filter Type of Channel 1	00h
2	Unsigned8	rw	AI Filter Type of Channel 2	00h
3	Unsigned8	rw	AI Filter Type of Channel 3	00h
4	Unsigned8	rw	AI Filter Type of Channel 4	00h

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

The following values for filter types selection have been defined:

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

Table 16: Possible filters

Example: Read filter type of measurement 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 the module will send:
Value 01h of the data byte 4 signifies a "Moving average" filter has been selected.

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

The set filter type is not automatically saved in a non-volatile memory. Storage has to be triggered via index 1010h (see "Store Parameters" page 54).

AI Filter Constant

Index 61A1h

Via index 61A1h the filter constant of the respective measurement channel may be ste and read.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
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

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

The value of the filter constant may be selected from a value between 1 and 50. Writing a different value to filter constant will cause an error message.

Example: Write filter constant 5 for measurement channel 3 (subindex 3), module address is 2

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.4.sg-BOX module 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 make sure that the filter has been set correctly via index 61A0h previously. If no filter has been selected, writing a filter constant will have no effect.

The set filter constant is not automatically saved in a non-volatile memory. Storage has to be triggered via index 1010h (see “Store Parameters” page 54).

AI Field Value

Index 9100h

Index 9100h is read only. It states the measurement value of the selected channel. This value, however, has not been linearized but may have been filtered. It shows the actual value of the AD converter (which may have been filtered).

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Signed32	ro	AI Field Value of Channel 1	0000 0000h
2	Signed32	ro	AI Field Value of Channel 2	0000 0000h
3	Signed32	ro	AI Field Value of Channel 3	0000 0000h
4	Signed32	ro	AI Field Value of Channel 4	0000 0000h

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

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

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
602h	8	40h	00h	91h	03h	00h	00h	00h	00h

As a possible response the module will send:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
582h	8	43h	00h	91h	03h	11h	0Ah	00h	00h

Data byte B4 (Low Byte) and B7 (High Byte) indicates the measurement value 00000A11h of the AD converter.

Retrieving data from all four channels simultaneously is possible via PDOs but not via SDOs (see “PDO Communication” page 77).

AI Scale Factor

Index 6126h

Via index 6126h measurement values at each channel may be scaled.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Float	rw	AI Scale Factor of Channel 1	1.0
2	Float	rw	AI Scale Factor of Channel 2	1.0
3	Float	rw	AI Scale Factor of Channel 3	1.0
4	Float	rw	AI Scale Factor of Channel 4	1.0

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

For using this object for scaling purposes please refer to chapter “User-defined Scaling” page 94.

AI Scale Offset

Index 6127h

Via index 6127h the process value of each channel may be combined with an Offset.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Float	rw	AI Scale Offset of Channel 1	0.0
2	Float	rw	AI Scale Offset of Channel 2	0.0
3	Float	rw	AI Scale Offset of Channel 3	0.0
4	Float	rw	AI Scale Offset of Channel 4	0.0

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

For using this object for scaling purposes please refer to chapter “User-defined Scaling” page 94.

AI Process Value

Index 9130h

Index 9130h is read only. It states the linearized process values of the selected channel. Linearization depends on the selected process value. The index is structured as follows:

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Signed32	ro	AI Process Value of Channel 1	0000 0000h
2	Signed32	ro	AI Process Value of Channel 2	0000 0000h
3	Signed32	ro	AI Process Value of Channel 3	0000 0000h
4	Signed32	ro	AI Process Value of Channel 4	0000 0000h

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

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

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
602h	8	40h	30h	91h	03h	00h	00h	00h	00h

As a possible response the module will send:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
582h	8	43h	30h	91h	03h	45h	03h	00h	00h

Data byte B4 (Low Byte) and B7 (High Byte) states the process value 0345h = 837d = 0.00837 V in case of the current version of the µCAN.4.sg-BOX module.

At the same time, the objects for error status of the device (see "Error Register" page 50) and state of the measurement inputs (see "AI State" page 67) will show the corresponding values.

If an analogue input error is detected, an emergency message is sent via the bus (see "Emergency Message" page 91).

8.5 CANopen Device Monitoring

To monitor a CANopen device two mechanisms (protocols) are available:

- Heartbeat protocol
- Node guarding

CAN in Automation recommends using the heartbeat protocol for monitoring only (acc. to CiA AN 802 V1.0: CANopen statement on the use of RTR-messages).

8.5.1 Heartbeat Protocol

Via Heartbeat protocol other devices on the network are able to check proper functioning and condition of the module.

Heartbeat ID

The identifier through which the module sends a heartbeat is set to 700h + module ID and cannot be changed. The message repetition time (called Producer Heartbeat Time) may be set via index 1017h.

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

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

Table 17: State information for heartbeat

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

Example: Switching on µCAN module with module address 2

ID	DLC	B0
702h	1	00h

Consumer Heartbeat Time

Index 1016h

Via index 1016h the consumer heartbeat time may be set.

Sub-index	Data type	Access	Description	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

Via the µCAN.4.sg-BOX another two devices (heartbeat producer) may be monitored. A failure of a heartbeat producer within a pre-set time will trigger an emergency message with the value 8130h (Life guard error or heartbeat error).
Time and module address of the module to be monitored is set via the 32 bit value.

Bit 31 ... 24	Bit 23 ... 16	Bit 15 ... 0
reserved (00h)	Module address	Heartbeat producer time

The time is stated in milliseconds. If the value 0 is selected for time or the value 0 or higher than 127 for module ID, the consumer heartbeat time will not be used or activated. The consumer heartbeat time will be activated after the first producer heartbeat has been received.

8

Producer Heartbeat Time

Index 1017h

Via index 1017h the producer heartbeat time may be set. The time is set in milliseconds. Selecting the time 0 ms will switch off the heartbeat protocol.

Sub-index	Data type	Access	Description	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, module address 1

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

As a response the module will send:

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

The set producer heartbeat time is not automatically saved in a non-volatile memory. Storage has to be triggered via index 1010h.

8.5.2 Node Guarding

Where a module is monitored cyclically (Node Guarding) the NMT master regularly retrieves the NMT condition from the NMT slaves. The NMT slaves which participate in the monitoring process internally check whether the "Node Guarding" is performed within the defined time period (Life Guarding). This is necessary to ascertain whether a NMT master is still "alive".

If there is not any data retrieved within a pre-defined time period, the module will send an emergency message with the value 8130h (Life guard error or heartbeat error).

Time for monitoring the module

Index 100Ch

Via index 100Ch the time value may be set which is multiplied by the value of index 100Dh to set the module guarding.

Sub-index	Data type	Access	Description	Default value
0	Unsigned16	rw	Guard time	0000h

The time is set in milliseconds. Setting the value 0000h disables monitoring of the module.

Factor for monitoring the module

Index 100Dh

Via index 100Dh the factor for the monitoring time of the module is defined which is set via index 100Ch.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	rw	Life time factor	00h

Setting the value 00h disables monitoring of the module.

8.6 PDO Communication

PDOs (Process Data Objects) are used to transmit the process data.

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

8.6.1 Transmission Modes

Sychronous Transmission Modes

Synchronous transmission modes will be applied if a user is able to trigger the SYNC message in the CANopen network. The synchronous transmission is defined by the „PDO transmission type" in the communications parameter of the respective process data object. For further details please refer to the CiA Draft Standard 301.

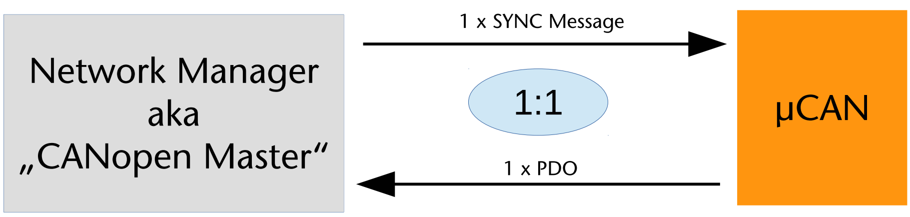


Fig.16: SYNC driven PDO communication

This figure shows the behaviour of the µCAN when each SYNC will trigger a PDO transmission.

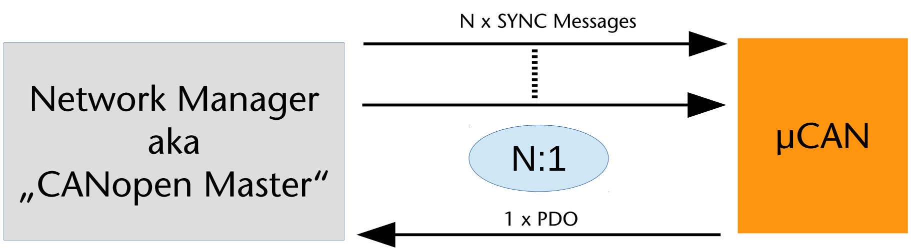


Fig.17: Factor N x SYNC driven PDO communication

This figure shows the when every N x SYNC will trigger a PDO transmission.

Cyclic or Asynchronous Transmission

The input information may be transmitted cyclically (e.g. every 100 ms) via transmit PDO. The cycle time for transmission may be set via the objects 18xxh of the respective transmit PDO.

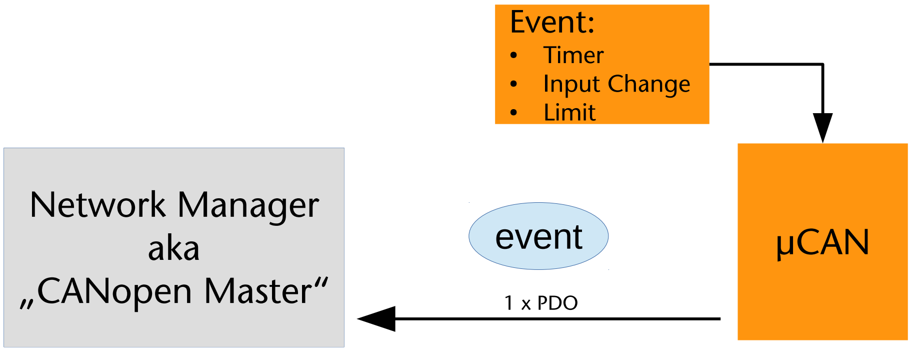


Fig.18: Event driven PDO communication

There are several events which can be configured by the user to trigger a µCAN-module to send a PDO.

8.6.2 PDO Communication parameter

Each PDO can be configured individually by a set of parameters. This refers to each transmit PDO as well as for receive PDOs (if supported). The modules are shipped with a default configuration which eventually may fit into your application. But in case of any customers settings the parameter must be changed.

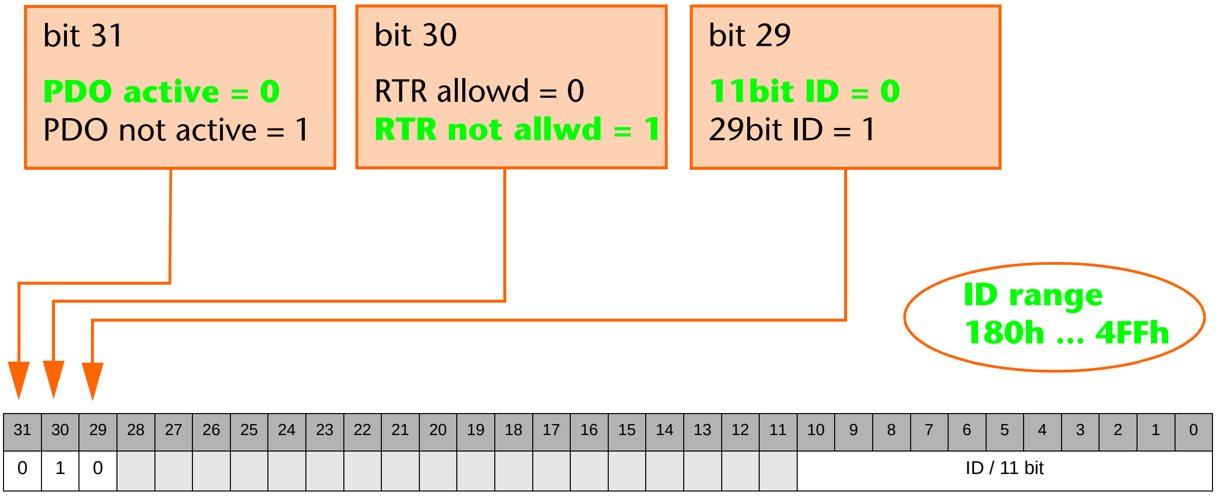
Sub-index	Access	Description	Value range
0	read only	Largest sub-index	05 / fix
1	read/write	COB-ID for PDO	0 .. 2^32
2	read/write	Transmission type	0 .. 255
3	read/write	Inhibit time	0 .. 65536 ms
4	--	n. A.	--
5	read/write	Event timer	0 .. 65536 ms

8

COB-ID for PDO

Via sub index 1 the ID is set which is to receive the PDO. It is defined as follows:

When using PDOs with 11-bit IDs the parameter is represented as follows:

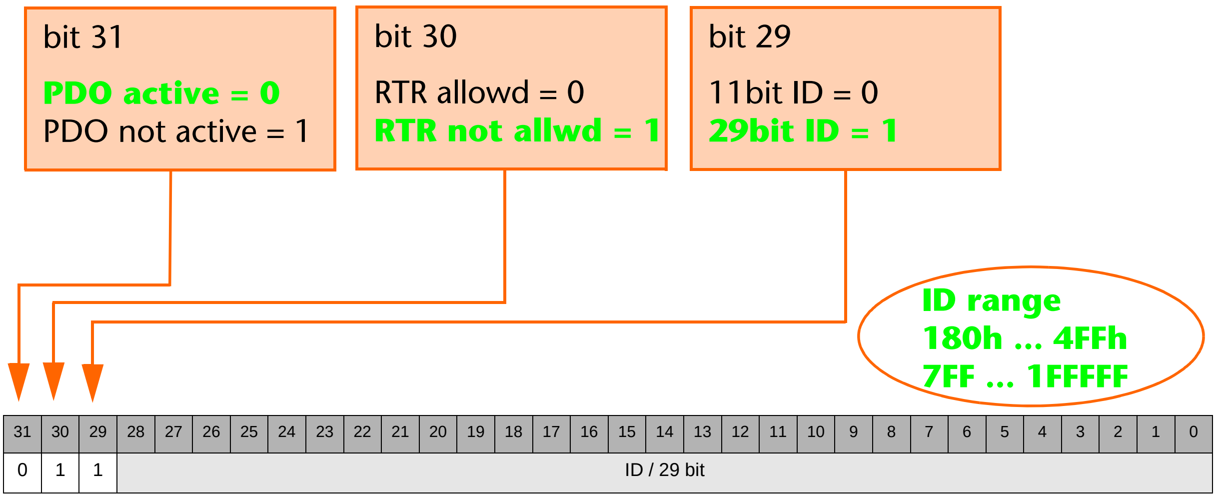


For example, the parameter value for a PDO ID = 330h is:

40000330 hex

When using PDOs with 29-bit IDs the parameter is represented as follwos:

8



For example, the parameter value for a 29bit PDO ID = 1FF000h is:

601FF000 hex

Transmission type

Via sub index 2 the transmission type of the message may be set.

Transmission type	Description
00h or 01h	1:1 each SYNC will trigger a PDO
02 ... F0h (2 ... 240)	N:1 each N SYNC will trigger a PDO
FEh	manufacturer event please refer to PDO parameters
FFh	timer driven PDO is sent after expiry of event timer

Table 18: Transmission type setting

Inhibit time

Via sub index 3 the inhibit time can be set. When using the PDO event driven (manufacturer event), it might happen that the event occurs several times per milli-second. This will block the CAN bus with multiple PDO messages. To avoid the unintended/multiple sending of PDOs, the user can set a "delay" or inhibit time between sending of 2 PDOs.

Transmission type	Description
0 ... FFFF h	Event timer in ms range 5ms ... 65535ms

Table 19: Transmission type setting

Event timer

Via sub index 5 the event timer can be set.

Event timer	Description
0 ... FFFF h	Event timer in ms range 5ms ... 65535ms

Table 20: Transmission type setting

The µCAN products support a multiple of 5ms for the event timer. The minimum timer value is 5ms. If faster transmission rates are needed, please refer to manufacturer events.

8.6.3 Transmit PDO 1 Parameter

Index 1800h

Via index 1800h the communication parameters of the transmit-PDO 1 may be set.

Sub-index	Data type	Access	Description	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
3	Unsigned16	rw	Inhibit time	0000h
5	Unsigned16	rw	Event timer	0000h

Example: Change COB ID of PDO to 330h, node address 1:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
601h	8	22h	00h	18h	01h	30h	03h	00h	04h

As a response the module will send:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
581h	8	60h	00h	18h	00h	00h	00h	00h	00h

The new COB-ID is not saved in a non-volatile memory. Storage has to be triggered via index 1010h.

When changing and storing the PDO parameters, the COB-ID will not change dynamically anymore when changing the node address by DIP-switches and re-power the unit. To keep the "dynamically" PDO ID, please refer to “COB-ID Storage” page 59.

8.6.4 Transmit PDO 2 Parameter

Index 1801h

Via index 1801h the communication parameters of the transmit-PDO 1 may be set.

Sub-index	Data type	Access	Description	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
3	Unsigned16	rw	Inhibit time	0000h
5	Unsigned16	rw	Event timer	0000h

Example: Change COB ID of PDO to 330h, node address 1:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
601h	8	22h	01h	18h	01h	30h	03h	00h	04h

As a response the module will send:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
581h	8	60h	01h	18h	00h	00h	00h	00h	00h

The new COB-ID is not saved in a non-volatile memory. Storage has to be triggered via index 1010h.

When changing and storing the PDO parameters, the COB-ID will not change dynamically anymore when changing the node address by DIP-switches and re-power the unit. To keep the "dynamically" PDO ID, please refer to “COB-ID Storage” page 59.

8.6.5 Transmit PDO 3 Parameter

Index 1802h

Via index 1802h the communication parameters of the transmit-PDO 1 may be set.

Sub-index	Data type	Access	Description	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
3	Unsigned16	rw	Inhibit time	0000h
5	Unsigned16	rw	Event timer	0000h

Example: Change COB ID of PDO to 330h, node address 1:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
601h	8	22h	02h	18h	01h	30h	03h	00h	04h

As a response the module will send:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
581h	8	60h	02h	18h	00h	00h	00h	00h	00h

The new COB-ID is not saved in a non-volatile memory. Storage has to be triggered via index 1010h.

When changing and storing the PDO parameters, the COB-ID will not change dynamically anymore when changing the node address by DIP-switches and re-power the unit. To keep the "dynamically" PDO ID, please refer to “COB-ID Storage” page 59.

8.6.6 Transmit PDO 4 Parameter

Index 1803h

Via index 1801h the communication parameters of the transmit-PDO 1 may be set.

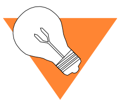
Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	05h
1	Unsigned32	rw	COB-ID for PDO	480h + Node-ID
2	Unsigned8	rw	Transmission type	01h
3	Unsigned16	rw	Inhibit time	0000h
5	Unsigned16	rw	Event timer	0000h

Example: Change COB ID of PDO to 330h, node address 1:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
601h	8	22h	03h	18h	01h	30h	03h	00h	04h

As a response the module will send:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
581h	8	60h	03h	18h	00h	00h	00h	00h	00h



The new COB-ID is not saved in a non-volatile memory. Storage has to be triggered via index 1010h.

When changing and storing the PDO parameters, the COB-ID will not change dynamically anymore when changing the node address by DIP-switches and re-power the unit. To keep the "dynamically" PDO ID, please refer to “COB-ID Storage” page 59.

8.6.7 Transmit PDO 1 Mapping Parameter

Index 1A00h

Via index 1A00h the mapping objects of the first transmit PDO will be displayed.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Unsigned32	ro	Mapped application object 1	9130 0120h
2	Unsigned32	ro	Mapped application object 2	9130 0220h

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

Objects transmitted via PDO may be read here. The structure is shown in the following table.

bit 31 - bit 16	bit 15 - bit 8	bit 7 - bit 0
Index	Sub-index	Length

Table 21: Structure of transmit PDO mapping parameter

The first PDO transmits sub indices 1 to 2 of object 9130h (see “AI Process Value” page 72).

8

8.6.8 Transmit PDO 2 Mapping Parameter

Index 1A01h

Via index 1A01h the mapping objects of the first transmit PDO will be displayed.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Unsigned32	ro	Mapped application object 1	9130 0320h
2	Unsigned32	ro	Mapped application object 2	9130 0420h

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

Objects transmitted via PDO may be read here. The structure is shown in the following table.

bit 31 - bit 16	bit 15 - bit 8	bit 7 - bit 0
Index	Sub-index	Length

Table 22: Structure of transmit PDO mapping parameter

The first PDO transmits sub indices 3 to 4 of object 9130h (see "AI Process Value" page 72).

8.6.9 Transmit PDO 3 Mapping Parameter

Index 1A02h

Via index 1A02h the mapping objects of the second transmit PDO will be displayed.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h
1	Unsigned32	ro	Mapped application object 1	9100 0120h
2	Unsigned32	ro	Mapped application object 2	9100 0220h

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

Objects transmitted via PDO may be read here. The structure is shown in the following table.

8

Bit 31 - Bit 16	Bit 15 - Bit 8	Bit 7 - Bit 0
Index	Sub-index	Length

Table 23: Structure of transmit PDO mapping parameter

The second PDO transmits sub indices 1 to 2 of object 9100h (see "AI Field Value" page 70).

8.6.10 Transmit PDO 4 Mapping Parameter

Index 1A03h

Via index 1A03h the mapping objects of the second transmit PDO will be displayed.

Sub-index	Data type	Access	Description	Default value
0	Unsigned8	ro	Largest sub-index	04h

Sub-index	Data type	Access	Description	Default value
1	Unsigned32	ro	Mapped application object 1	9100 0320h
2	Unsigned32	ro	Mapped application object 2	9100 0420h

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

Objects transmitted via PDO may be read here. The structure is shown in the following table.

Bit 31 - Bit 16	Bit 15 - Bit 8	Bit 7 - Bit 0
Index	Sub-index	Length

Table 24: Structure of transmit PDO mapping parameter

The second PDO transmits sub indices 3 to 4 of object 9100h (see “AI Field Value” page 70).

8

8.6.11 Transmit-PDO Example

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

Example: Module address 1, send SYNC

ID	DLC
80h	0

As a response the module will send the following messages:

<table>
<tr><th>ID</th><th>DLC</th><th>B0</th><th>B1</th><th>B2</th><th>B3</th><th>B4</th><th>B5</th><th>B6</th><th>B7</th></tr>
<tr><td>181h</td><td>8</td><td colspan="4">Index 9130h,
Sub 01h
PV Ch1 +/-50,00000 mV</td><td colspan="4">Index 9130h,
Sub 02h
PV Ch2 +/-50,00000 mV</td></tr>
</table>

<table>
<tr><th>ID</th><th>DLC</th><th>B0</th><th>B1</th><th>B2</th><th>B3</th><th>B4</th><th>B5</th><th>B6</th><th>B7</th></tr>
<tr><td>281h</td><td>8</td><td colspan="4">Index 9130h,
Sub 03h
PV Ch3 +/-50,00000 mV</td><td colspan="4">Index 9130h,
Sub 04h
PV Ch4 +/-50,00000 mV</td></tr>
</table>

<table>
<tr><th>ID</th><th>DLC</th><th>B0</th><th>B1</th><th>B2</th><th>B3</th><th>B4</th><th>B5</th><th>B6</th><th>B7</th></tr>
<tr><td>381h</td><td>8</td><td colspan="4">Index 9100h,
Sub 01h
FV Ch1 / ADC raw data</td><td colspan="4">Index 9100h,
Sub 02h
FV Ch2 / ADC raw data</td></tr>
</table>

<table>
<tr><th>ID</th><th>DLC</th><th>B0</th><th>B1</th><th>B2</th><th>B3</th><th>B4</th><th>B5</th><th>B6</th><th>B7</th></tr>
<tr><td>481h</td><td>8</td><td colspan="4">Index 9100h,
Sub 03h
FV Ch3 / ADC raw data</td><td colspan="4">Index 9100h,
Sub 04h
FV Ch4 / ADC raw data</td></tr>
</table>

Sending of PDOs is possible only in "Operational" mode of the devices. The transmit PDOs are sent once after NMT "Start all Nodes" regardless of the transmission type.

Example: A module with module address 2 sends a boot-up message, is then switched to Operational via NMT "Start all Nodes" and transmits both PDOs for 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 8 00 00 00 00 00 00 00 00
 5)  Rx 0382 8 00 00 00 00 00 00 00 00
 6)  Rx 0482 8 00 00 00 00 00 00 00 00
 7)  Tx 0080 0
 8)  Rx 0182 8 00 00 00 00 00 00 00 00
 9)  Rx 0282 8 00 00 00 00 00 00 00 00
10)  Rx 0382 8 00 00 00 00 00 00 00 00
11)  Rx 0482 8 00 00 00 00 00 00 00 00
12)  Tx 0080 0
13)  Rx 0182 8 00 00 00 00 00 00 00 00
14)  Rx 0282 8 00 00 00 00 00 00 00 00
15)  Rx 0382 8 00 00 00 00 00 00 00 00
16)  Rx 0482 8 00 00 00 00 00 00 00 00
```

Example 1: Trace: A module is set to Operational and sends PDOs

8

8.7 Synchronization Message

Index 1005h

Via Index 1005h the identifier for the synchronization message (SYNC) may be defined. Through the SYNC message transmission of a PDO may be triggered.

Sub-index	Data type	Access	Description	Default value
0	Unsigned32	rw	COB-ID SYNC	80h

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

Example: Set COB-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 a response the module will send:

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 which will ensure high priority of the SYNC messages on the CAN bus.

The set SYNC-ID is not automatically saved in a non-volatile memory. Storage has to be triggered via index 1010h.

8.8 Emergency Message

Emergency messages (EMCY) will automatically be sent by the µCAN module every time an error occurs. Please note the difference between SDO error messages, indicating erroneous access to a SDO object, and a "real" emergency message. If an error occurs for the first time, an error message will be sent. If the error has been eliminated and does not exist any-more, an error message will be sent as well (error code 0000h).

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

The 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				

Error classified by "Error Code".

In field "ER" (error register) of the emergency message shows the content of the CANopen object 1001h.

The „Manufacturer Specific Error Field" contains further manufacturer specific information for detecting errors.

8.8.1 Overview of Error Codes

The following error codes are supported:

Error code	Error Field [hex] [b3 b4 b5 b6 b7]	Description
0000h	00 00 00 00 00	Error reset or no error
5030h	0x 0x 0x 0x 00	"Sensor fault", a sensor error occurred
8100h	xx 00 00 00 00	CAN controller in "Warning" mode
8110h	00 00 00 00 00	CAN Controller in "Overrun" mode, too many messages
8120h	xx 00 00 00 00	CAN controller in "Error Passive" mode
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	Collision of identifiers (transmit identifier has been detected)

Table 25: Error codes of the emergency message

The column "Error Field" indicates whether a "Manufacturer Specific Error Field" is used or not. In some cases, this field is used to give additional information about the error code displayed.

The µCAN module will save all emergency messages in an error history which may be accessed via object 1003h of the CANopen object dictionary.

Error field "5030h Error Code"

5030h Error Code

In case of an error the respective measurement channel will be masked with 01h and displayed accordingly in this field.

B3	**B4**	**B5**	**B6**	**B7**
0xh	0xh	0xh	0xh	00h
Measurement channel 1	Measurement channel 2	Measurement channel 3	Measurement channel 4	-

Table 26: Manufacturer specific error field of 5030h Error Code

Example: Emergency message has the following content:

ID	**DLC**	**B0**	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**
82h	8	50h	30h	08h	00h	01h	01h	00h	00h

The sent Emergency indicates that a sensor error has occurred at the µCAN module with the module address 2, measurement channels 2 and 3.

Error field "8100h Error Code" and "8120h Error Code"

8100h and 8120h Error Code

If an EMCY message containing one of the listed error codes is sent, the internal values of the CAN controller will also be indicated in data byte 3. The following table gives an overview of the error codes.

B3	**Description**
00h	no error
10h	Bit error

Table 27: Error codes of 8100h and 8120h Error Code

B3	Description
20h	Stuffing error
30h	Form error
40h	CRC error
50h	ACK error

Table 27: Error codes of 8100h and 8120h Error Code

8.9 User-defined Scaling

The µCAN.4.sg-BOX is able to scale your measurement values as desired. To do so, factor and offset may be pre-set via the objects 6126h and 6127h. The sub index of the object corresponds with the analogue input.

Scaling is switched off if the value *1.0* is set for factor and the value *0.0* is set for offset.

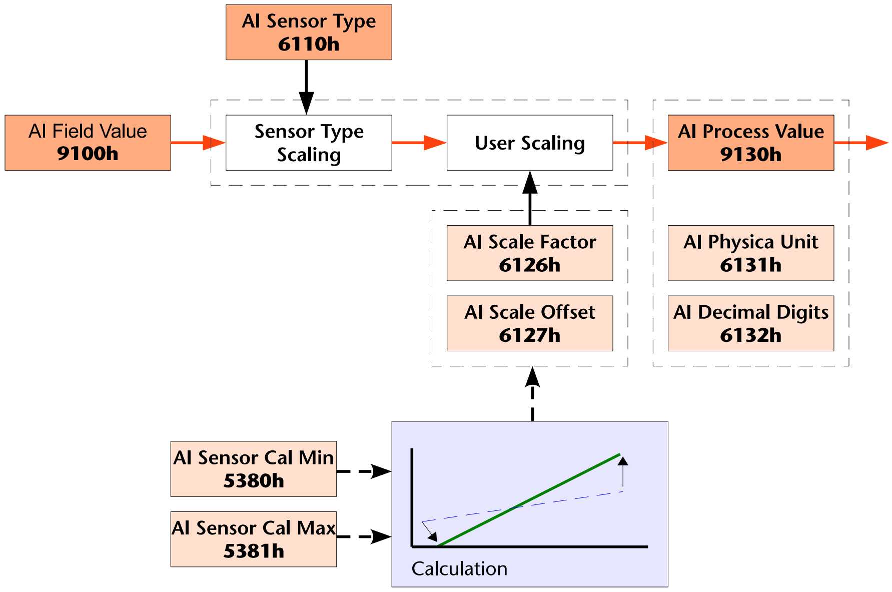


Fig.19: Realization of user defined scaling

Another way of scaling is via objects 5380h "AI Sensor Calibrate Minimal Value" and 5381h "AI Sensor Calibrate Maximal Value".

In this case, a signal has to be entered to the measurement input of the µCAN.4.sg-BOX which equals the minimum value. Then this minimum value has to be entered to the sub index of the respective analogue input in object 5380.

After this, the signal has to be entered which equals the maximum value. And the maximum value has to be entered to the sub index of the respective analogue input in object 5381.

This will automatically determine the scaling parameters and enter them to objects 6126h and 6127h.

If scaling is enabled, the original (unscaled) process value will be transmitted as "field value".

The set scaling parameters are not automatically saved in a non-volatile memory. Storage has to be triggered via index 1010h (see “Store Parameters” page 54).

The following table represents the unscaled measurement values of the µCAN.4.sg-BOX module:

	Signal on +/-	"Field Value"	"Process Value"
strain gauge 32Bit	-50mV	FF9C 0000$_h$ -6553600$_d$	FFB3 B4C0$_h$ -50,00000$_d$
	0 mV	0000 0000$_h$ 0$_d$	0000 0000$_h$ 0,00000$_d$
	+50mV	0064 0000$_h$ 6553600$_d$	004C4B40$_h$ +50,00000$_d$

Table 28: Measurement values of µCAN.4.sg-BOX

Example step by step for user defined scaling:
The input of channel 1 is supposed to output a signal of 0 to 100,0 kN.

Step 1:
Apply the strain gauge with the physical minimum value to channel 1. This shall be no force on strain gauge and the module will read the inital offset of your system.

Index 5380h Minimal Value Scaling

Step 2:
Now you can write the corresponding process value of 0,0 kN. This will be used as the first parameter for scaling:

Example: write 0 to channel 1, module address 1

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
601h	8	22h	80h	53h	01h	00h	00h	00h	00h

As a response the module will send:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
581h	8	60h	80h	53h	01h	00h	00h	00h	00h

Step 3:
Now apply the maximum physical force to the strain gauge connected to channel 1. This should be the maximum value and the system will read the current measuring value.

Index 5381h Max Value Scaling

Step 4:
Now you can write the corresponding process value of 100,0 kN. This will be used as second parameter for scaling:

Example: write 1000 (=100,0) to channel 1, module address 1

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
601h	8	22h	81h	53h	01h	E8h	03h	00h	00h

As a response the module will send:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
581h	8	60h	81h	53h	01h	00h	00h	00h	00h

Now the scaling on channel 1 is active and working. You can perform some measurement now. If satisfied with the results, you should store your scaling to EEPROM.

The set customer scaling is not automatically saved in a non-volatile memory. Storage has to be triggered via index 1010h.

9. Technical Data

Power Supply	
Supply Voltage	9..36V DC, reverse polarity protected
Power consumption	1,86 W (155 mA @ 12 V DC) 1,92 W (80 mA @ 24 V DC) 2,08 W (65 mA @ 32 V DC)
Isolation	Field bus / control voltage 500 Veff
Connection	Screw terminals

CAN bus	
Bit 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

EMC	
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 61000-4-4
Surge	according to EN 61000-4-5
Conducted RF disturbance	10 V, according to EN 61000-4-6
Electromagnetic emission	according to EN 55011, class A

Measurement technology	
Operating temperature	-40°C to +85°C
Signal type	±50 mV
Resolution	16 bit / 24 bit (diff. articles)
Sampling rate	200 Hz at each measurement channel

Casing	
Aluminium die-cast casing	EN AC-44300 DIN EN 1706 (GD Al Si 12 / DIN 1725)
Draft angle	1° - clear interior dimensions are circumferentially reduced to the bottom of the casing by 1°
Screws	Stainless steel captive screws
Protection Class	IP 66 / EN 60529
Sealing	Tongue and groove system, with groove in casing lid, silicone sealing resistant to liquid fuel and oil
Mounting	Through separate screw channels.
Coating	Standard powder coating Pebble grey RAL 7032, silver grey RAL 7001 at no extra charge. Structured baking varnish and individual colour available on request.
Dimensions	125 * 80 * 57 mm (L * W * H) without high-strength cable glands or plugs
Weight	540 g
Weight with connections	640 g

Index

MicroControl shall neither assume liability for compliance of the content of this manual with the respective legal provisions nor for errors and technical specifications.

MicroControl GmbH & Co. KG
Junkersring 23
D-53844 Troisdorf
Fon: +49 / 2241 / 25 65 9 - 0
Fax: +49 / 2241 / 25 65 9 - 11
http://www.microcontrol.net