MicroControl

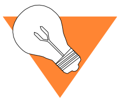
Systemhaus für Automatisierung

µCAN.4.ti-BOX

Manual for 4-channel temperature measurement modules
Version 4.00

Document conventions

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



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

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

Keywords

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

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, because it contains important information about the secure handling of electrical devices.

1.1 General Safety Regulations

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

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

- You know the safety regulations for automated machines and you are familiar with the machine.
- You are the operator for the machine and you have been trained on operation modes. You are familiar with the operation of devices described in this manual.
- You are responsible for setting into operation or service and you are trained on repairing automated machines. In addition you are trained in setting electrical devices into operation, to connect the earthing conductor and to label these devices.

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

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

Please take care to observe the actual local safety regulations.

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

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

1.2 Safety Notice

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

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

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

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

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

2. Operation of µCAN.4.ti-BOX

2.1 Overview

The µCAN.4.ti-BOX is the right solution to measure und to linearise temperatures. This device supports resistance thermometers and thermocouple elements. Measured temperatures are transmited via CAN fieldbus in degree Celsius [°C].

2

Fig. 1: Temperature module µCAN.4.ti-BOX

Resistance thermometers (RTDs) can be connectetd in two-wire, three-wire or four-wire configuration.

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

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

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

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

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

The µCAN.4.ti-BOX 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.4.ti-BOX is designed for direct use on DIN-rail mounting applications. The housing is also available with internal bus and power connector for stacking of several modules. The compact, space saving case gives the freedom to mount the module in many places.

inexpensive and service friendly

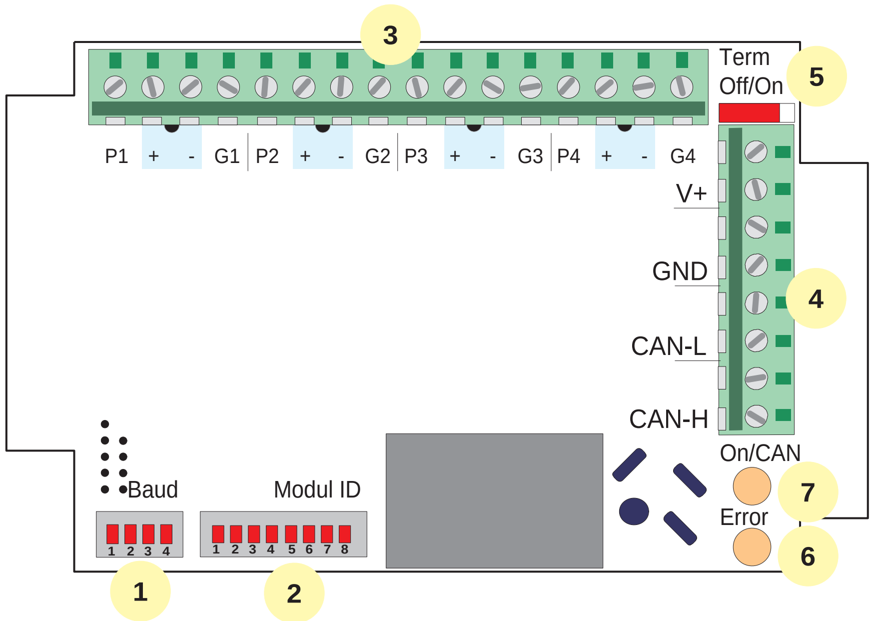
The quick and easy integration of the µCAN.4.ti-BOX in your application reduces the development effort. Costs for material and personnel are reduced. The easy installation makes maintenance and replacement quite simple.

3. Project Planning

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

3.1 Module Layout

The following figure shows the top view of the µCAN.4.ti-BOX PCB. Use the figure to identify the terminal blocks, LED's and DIP-switches.



1: Switch to configure baudrate
2: Switch to configure node ID
3: Terminal block for temperatur sensors
4: Terminal block for voltage supply and CAN
5: Switch for CANbus termination
6: Bi-color LED for device status
7: Bi-color LED for network status

Fig. 2: Top view of µCAN.4.ti-BOX PCB

3.2 Operation Area

The µCAN.4.ti-BOX is a robust field module for acquisition and linearisation of temperatures in industrial applications. Temperatures can be acquired by different kinds of sensors.

The following resistance thermometers can be connected to the µCAN.4.ti-BOX:

- PT100
- PT200
- PT500
- PT1000

The following thermocouples can be connected to the µCAN.4.ti-BOX:

- Type J
- Type K
- Type R
- Type T

Other temperature sensors are available on request.

The module gathers the analogue signal of temperature sensors and performs a linearisation. The temperature is transmitted in degree Celsius via CAN bus. Fraction of sensor (thermocouple / resistance thermometers) and short circuit of sensor (resistance thermometers) are detected.

The PCB is incorporated in a robust case of protection class IP65. The µCAN.4.ti-BOX is suited for mounting outside the switch cabinet. The idea behind that concept is to acquire the signals direct at the test point. Long wires for the sensors are not longer necessary. Influence of EMI is reduced.

The suppported supply voltage of µCAN.4.ti-BOX is 9..36V. The µCAN module needs a four core cable for connection of power supply and CAN bus, in order to reduce the amount of cabling. Special CAN bus cables are available as accessories.

3.3 Maximum System Layout

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

A CAN network may have one network manager and up to 127 network slaves (refer to Fig. 3, "Maximum System Layout"). Every module gets a unique address, which is set up via a DIP switch. The CAN bus is connected through the µCAN modules. The last module in the network must be terminated by a termination switch (refer to "Termination" on page 27).

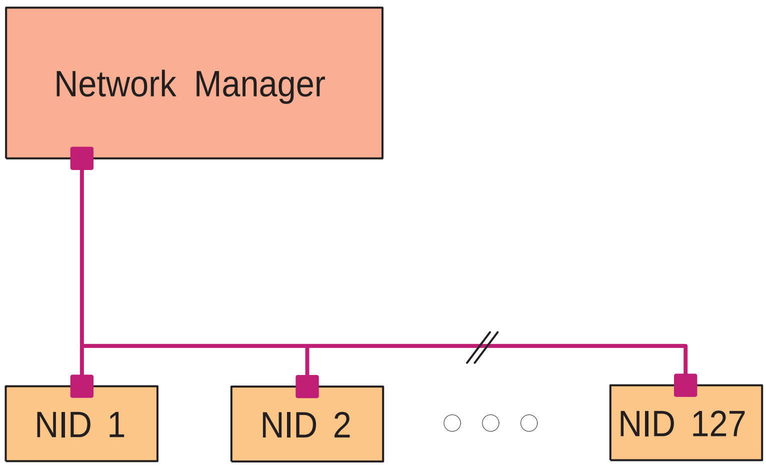


Fig. 3: Maximum System Layout

The maximum cable length depends on the selected bitrate. The following table shows the maximum cable length recommended by CiA[1]. These distances can be realized with the µCAN.4.ti-BOX.

Bitrate	Cable Length
1000 kBit/s	25 m
800 kBit/s	50 m
500 kBit/s	100 m
250 kBit/s	250 m
125 kBit/s	500 m
100 kBit/s	650 m
50 kBit/s	1000 m
20 kBit/s	2500 m

Table 1: Dependence of bitrate from cable length

The CiA recommends not to use 100 kBit/s baudrate in new systems.

1. CAN in Automation Internationial Users and Manufacturers Group e.V. MicroControl is a member of CiA and joins the working groups for development of new protocols and standards.

3.4 Case Dimensions

The case dimensions of the module are given in the drawing below. The high protection class IP65 of the µCAN module allows an assembly at places with a harsh environment. It is possible to mount the µCAN module inside a switching cabinet as well as direct on a machine. Please check the technical data section for detailled information about maximum environment conditions.

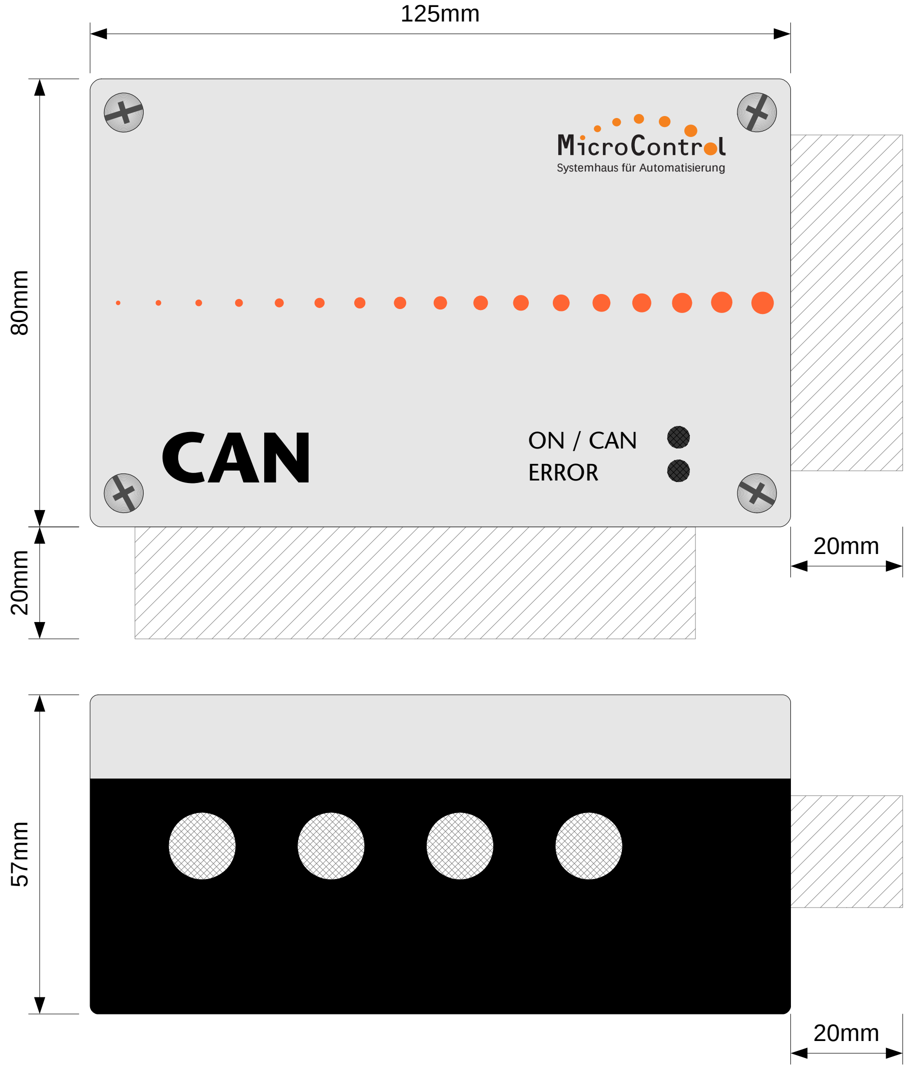
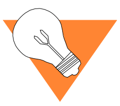


Fig. 4: Case Dimensions

The cross hatchures show the dimensions of the module when using cable glands. Make sure to add some additional space around the bare housing when using cable and cable glands. The absolute values for the additional space may vary due to different cable diameters and different size of cable glands.



Further details to the case are given in chapter "Technical Data" on page 87.

3

4. Assembly and Disassembly

4.1 General Information

Assembly

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

Power Supply

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

Earthed Conductor

The non-fused earthed conductor is connected at the terminal outside the case (refer to Fig. 5, "Connection of earthed conductor").

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

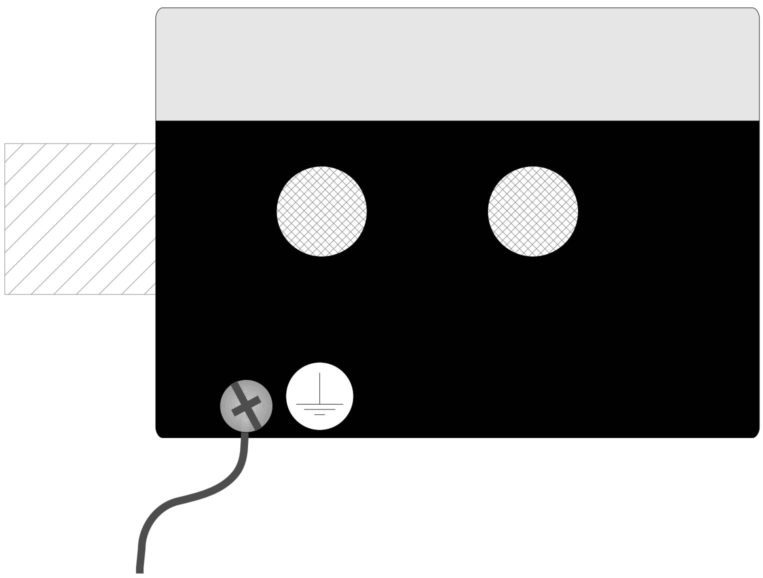


Fig. 5: Connection of earthed conductor

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

4.2 Assembly

The modules can be directly fixed to a metal plate. For this purpose you will find two holes in the body of the housing. Please use these two holes with M4 (4mm diameter) mounting screws which can be used for mounting the housing to a metal plate.

For ease of use you will find a drill template in the following figure.

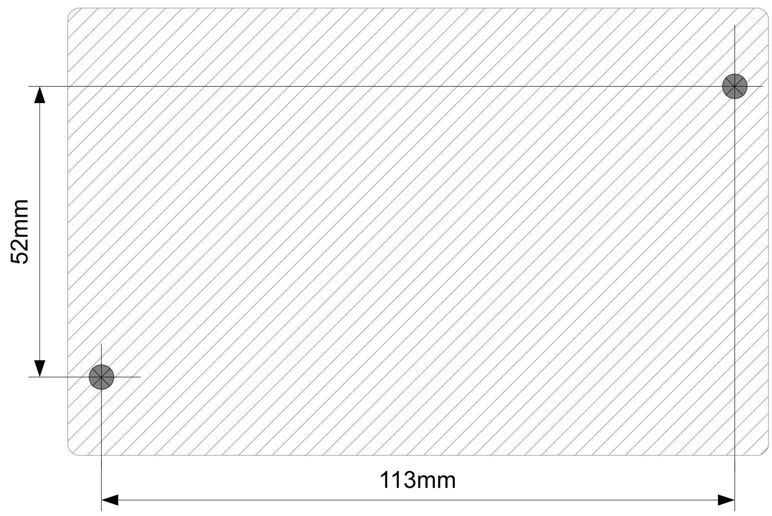


Fig. 6: Case bore-holes for screws

When fixing several µCAN modules at the same place please make sure to leave some additiona space for the cable glands.

For a quick identification of the modules during operation you may use paper sticker. Please write down the ID that is set for the module.

Please make sure that the last node that is installed to the CAN bus is terminated with a resistor (refer to “Termination” on page 27).

4.3 Disassembly

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

Open the cover from the module and remove the temperature sensors first. Now you can remove the cables for CAN bus and power supply from the terminals.

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

4

5. Installation

5.1 Potential Basics

The potential environment of a system that is realized with µCAN modules is characterized by following features:

- The CAN bus potential is isolated from the power supply.
- The electronic of the µCAN modules is isolated from the power supply.
- All µCAN modules have a separate power supply.
- All I/O signals are optically isolated from the CAN bus potential.

5.2 EMC Considerations

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

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

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

DC Coupling

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

Inductance Coupling

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

Capacitive Coupling

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

RF Coupling

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

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

5.2.1 Grounding

All inactive metal plates must be grounded with low impedance. By this step all elements of the system will have the same potential.

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

The µCAN modules are grounded by the contact which is mounted under one of the PG screws (refer to Fig. 5, on page 15). Additional contacts can be mounted under the PG screws for shielding purposes on demand. The ground potential may not be connected to a terminal inside the case.

If µCAN modules are shipped in a plastic case they have to be grounded with a metal tape.

5.2.2 Shielding of Cables

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

Cable Types

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

Cable 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. The cable shield can not be grounded on both ends for temperature sensors. The grounding on one end of the cable is necessary if

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

The shield of the CAN bus cable may never lead inside the µCAN case. Never connect the shield to one of the terminals inside the case.

For a fixed operation the shield of the CAN bus cable should be connected to safety earth.

5.2.3 CAN Cable

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

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

Table 2: Specifications of CAN bus cable

The CAN bus cable is connected to the µCAN.8.dio-BOX module via terminals inside the case. For the pinning of the terminal refer to “CAN Bus” on page 24 of this manual.

Do not confuse the signal lines of the CAN bus, otherwise communication between the modules is impossible. The shield of the CAN bus cable may never lead inside the µCAN case. Never connect the shield to one of the terminals inside the case.

5.3 Power Supply

The µCAN.4.ti-BOX modules are designed for industrial applications. By means of a DC/DC converter the CANbus of the module is isolated from the supply voltage. The supply voltage must be within the range from 9 V DC to 36 V DC. The input is protected against confusing the poles.

Please make sure not to confuse the poles when connecting the power supply.

The positive supply is connected to the terminal **V+**. The two **V+** terminals are internally linked to feed the supply through the module.

The negative supply is connected to the terminal **GND**. The two **GND** terminals are also internally linked.

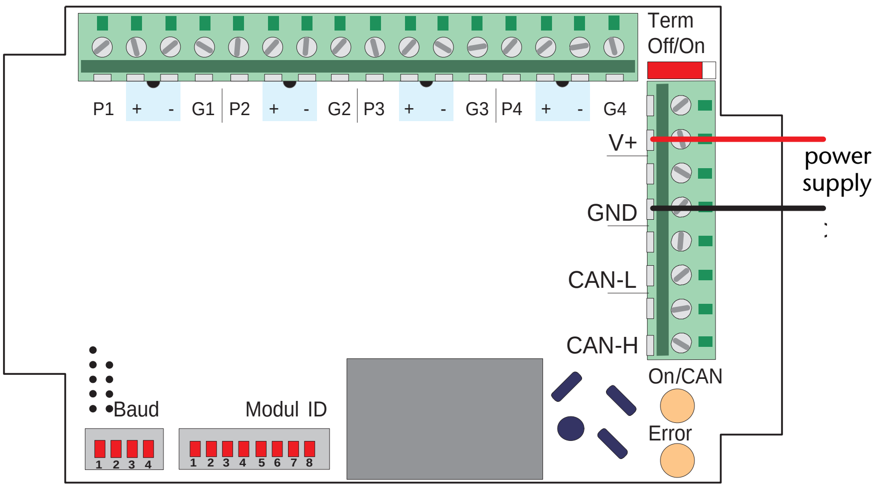
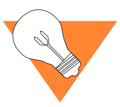


Fig. 7: Connection of power supply

The maximum supply voltage is 36V DC. Higher voltages will destroy the electronic.



A cable shield may not lead into the housing or may not be connected to a terminal inside the housing. Cable shields have to be connected to the terminals outside the housing.

5.4 CAN Bus

The two wires of the CAN bus are connected to the appropriate terminals. Please make sure that the CAN bus is fed from the right side into the module and keep the wires as short as possible. The terminals for CAN-H respective CAN-L are internally linked. By this the CAN bus can be connected through the module.

To reduce the influence of EMI please take care that the CAN bus cable does not cross the wires of the sensor.
Terminals for CAN

The CAN bus line with positive potential must be connected to the terminal **CAN-H**. The CAN bus line with negative potential must be connected to the terminal **CAN-L**.

5

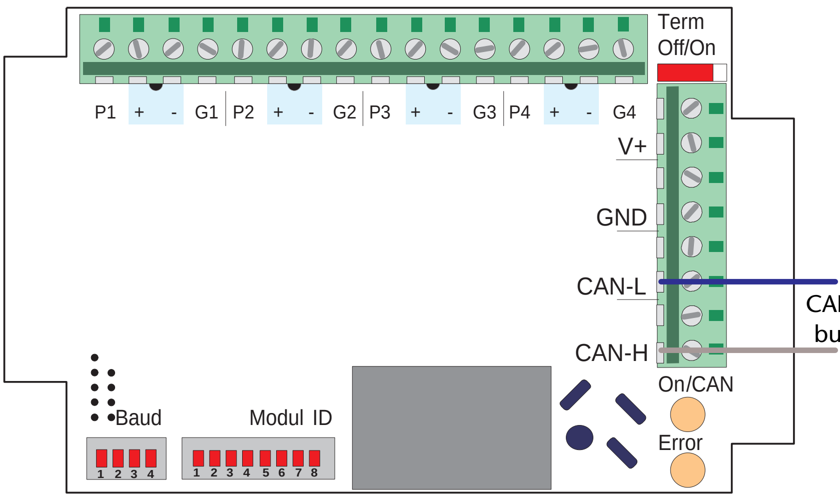
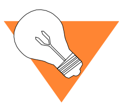


Fig. 8: Connection of CAN line

If you confuse the poles the communication on the bus will not be possible. The shield of the CAN bus may not lead into the housing and may not be connected to a terminal inside the housing. Cable shields have to be connected to the terminals outside the housing.



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

5.5 Module Address

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

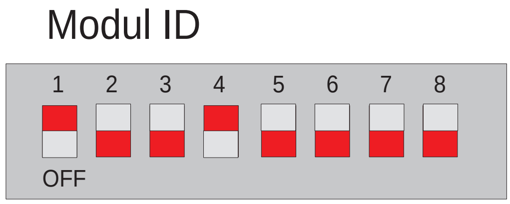


Fig. 9: Setup of µCAN module address (here address 9 is shown)

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

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

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

If module address and baudrate switches are set to position OFF than µCAN.4.ti-BOX starts in LSS mode.

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

5.6 Baudrate

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

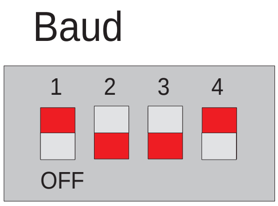


Fig. 10: Setup of baudrate (here 1 MBit/s is shown)

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

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

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

Table 3: Configuration of baudrate

a. LSS will be used when all module addresses switches are set to OFF

The baudrate 10 kBit/s is not supported by µCAN.4.ti-BOX. In the configuration **Autobaud** the µCAN.4.ti-BOX detects valid baudrate itself, automatically. In configuration **LSS** the stored baudrate and module address will be used.

5.7 Termination

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

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

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

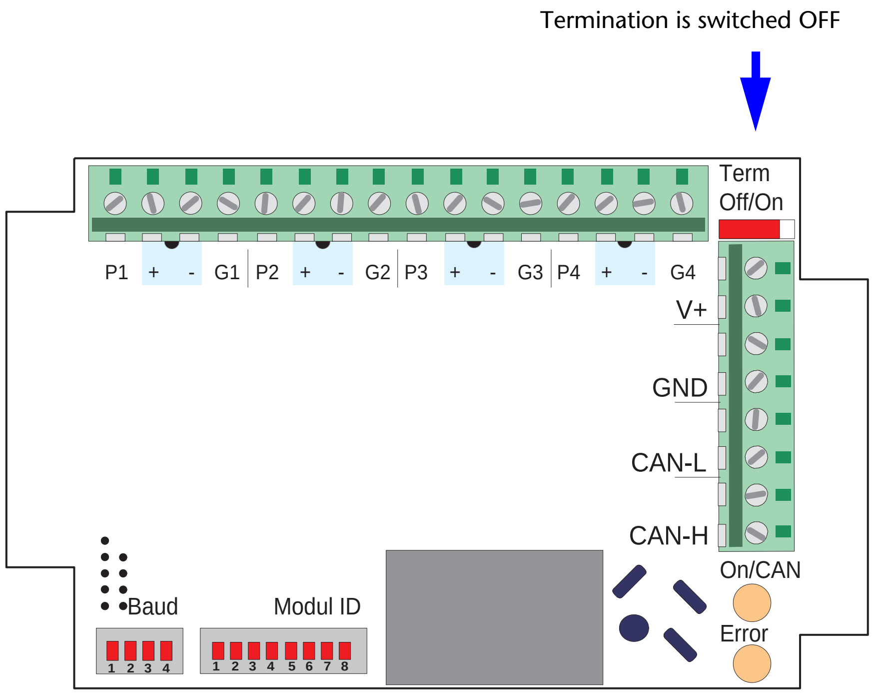


Fig. 11: Termination of CANbus

In the shown figure the terminations of µCAN.4.ti-BOX is switched off. So this µCAN module is used as a T-piece in a CAN network and an other µCAN module have to terminate the CAN line with a 120 Ohm resistor.

5

6. Signal Inputs

This chapter of the manual will show you how different kinds of temperature sensors and analogue standard signals are connected to the µCAN.4.ti-BOX modules. Please keep the basics of EMI rules in mind when planning the wiring. Only proper wiring and EMI precautions make sure that the module runs without trouble.

The µCAN.4.ti-BOX has four inputs, which are numbered from 1 to 4. The terminal with marking P1 belongs to channel 1. Also the following terminals, marked with +, - and G1, belong to channel 1. The last input (channel 4) has terminals marked with P4, +, - and G4.

All sensor types or analogue signals may only be connected in power off state in order to prevent a damage of the electronic.

6.1 Connection of Temperature Resistors

The module can handle different kind of temperature resistors. The supported resistors are Pt100, Pt200, Pt500 and Pt1000. The measuring range is defined from -200,0°C to +850,0°C for Pt100. The resistor value within this range is 18,520Ω to 390,481Ω. For measurement a constant current source is used and the current running through the resistor is 645µA.

In case of an invalid measuring signal (sensor break / sensor shortening) there will be displayed a measuring value of -437,0°C = -4370d (signed) = 61166d (unsigned) = EEEEh. An additional Emergency message will be send on the bus. For details please refer to refer to “Emergency Message” on page 85.

As mentioned before, the µCAN.4.ti-BOX works with Pt100 sensors as well as with thermocouples. Sensors of type Pt100 can be connected in three different ways.

6.1.1 Two-wire Connection

Connection between the Pt100 resistor and the electronic is done with 2 wires (refer to Fig. 12, “Connecting a Pt100 in 2-wire technique”). As every conductor these wires have an resistance, which is switched in series to the Pt100 resistor.

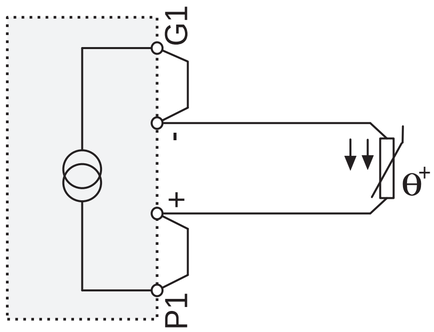


Fig. 12: Connecting a Pt100 in 2-wire technique

As a result the Pt100 resistor and the resistance of the wires are added. That means a higher temperature than the really present temperature is measured. To reduce this effect, the resistance of the wires must be compensated manually.

When using a 2-wire connection short circuits have to be added between the terminal block "G1" and "-" as well as between terminal "P1" and "+".

6.1.2 Three-wire Connection

In industrial applications quite often the Pt100 resistor is used in a 3-wire version. For this type of sensor an additional wire is connected to the Pt100 resistor. This additional wire generates a second measuring circuit. The second measuring circuit is used as reference. For a 3-wire Pt100 sensor the offset by the conductor resistance and the influence of the ambient temperature can be compensated. However the ambient temperature must influence all three wires. The µCAN.4.ti-BOX does not support the compensation of the second circuit. You may use 3-wire technology but the measurement is the the same as with 2-wire connectors.

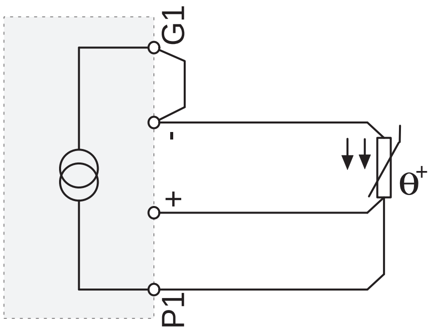
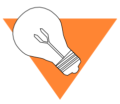


Fig. 13: Connecting a Pt100 in 3-wire technique



When using 3-wire technology there has to be added a short circuit between terminal block "G1" and "-".

6

6.1.3 Four-wire Connection

For a 4-wire Pt100 the current is fed into the resistance via two additional conductors. The voltage drop over the resistor is measured with the parallel conductors. A compensation is not necessary. For a high-impedance input the resistance of the conductor material can be neglected. The voltage drop over the Pt100 resistor is independent from the conductor resistance.
So this is the best technique for measuring Pt100 (or in general temperature resistors).

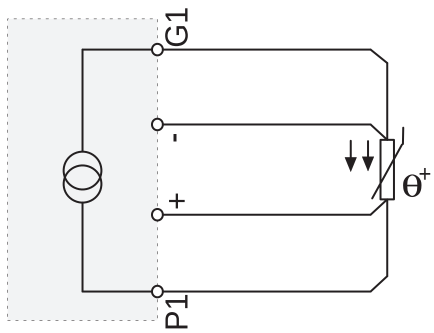
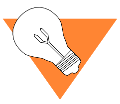
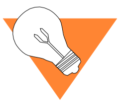


Fig. 14: Connecting a Pt100 in 4-wire technique

6



Measurement inputs which are not used by the application must be shorted through a link between the terminals „+“ and „-“ of the unused input. By this step the influence of EMI is reduced.



The shield of the temperature sensor may not lead inside the case in order to avoid EMI. The shield has to be connected outside the case to the appropriate terminal. All modules are shipped with one earthing connector, additional earthing connectors can be ordered.

6.2 Connection of Thermocouples

The µCAN.4.ti-BOX is featured for measuring different kinds of thermocouple sensors. The following table shows the actual supported thermocouple types:

thermoouple	min. temperature [°C]	max. temperature [°C]
Type J	-200,0	+1200,0
Type K	-200,0	+1200,0
Type R	-200,0	+1200,0
Type T	-200,0	+1200,0

Table 4: Measurement range of thermocouples

In case of an invalid measuring signal (sensor break) there will be displayed a measuring value of -437,0°C = -4370d (signed) = 61166d (unsigned) = EEEEh. An additional Emergency message will be send on the bus. For details please refer to refer to “Emergency Message” on page 85.

Please take care not to confuse the poles when connecting the thermocouple. This will lead to decreasing temperatures shown on the bus when heating up the thermocouple.
The following figure shows the connection of a thermocouple to measurement input 1.

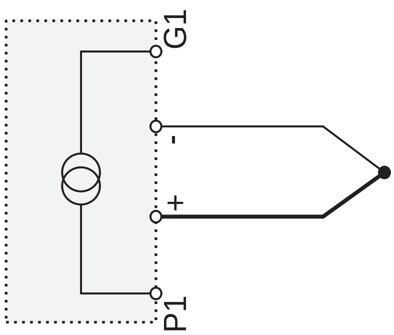


Fig. 15: Connection of thermocouple

Measurement inputs which are not used by the application must be shorted through a link between the terminals „+“ and „-“ of the unused input. By this step the influence of EMI is reduced.

The shield of the temperature sensor may not lead inside the case in order to avoid EMI. The shield has to be connected outside the case to the appropriate terminal. All modules are shipped with one earthing connector, additional earthing connectors can be ordered.

6

7. Diagnosis

All modules of the µCAN family have LEDs to display the operating state and to signalize an error state. The light of the LEDs can be seen through beam waveguides on top of the housing.

The µCAN.4.ti-BOX has two Duo-LEDs (green/red) labeled with "On/CAN" (state of CAN network) and "Error" (state of µCAN module).

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

Figure 16 shows the position of LEDs marked by (1) and (2).

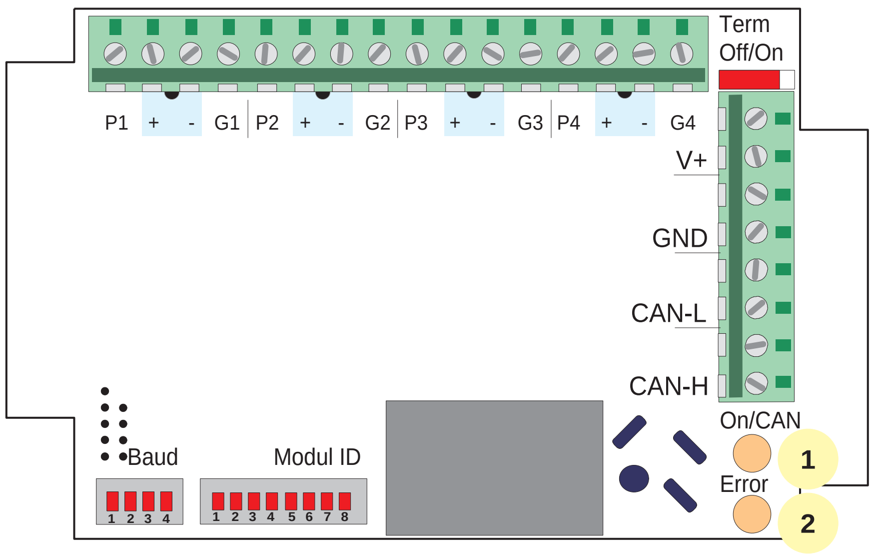


Fig. 16: Position of LEDs on µCAN.4.ti-BOX

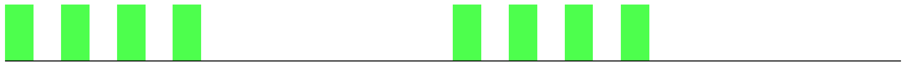
In normal operation all LEDs should have a green color. A red steady light or a red blinking of a LED indicates an error condition.

7.1 State of CAN network

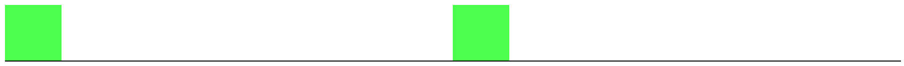
The LED labeled with "On/CAN" (state of CAN network) displays the state of CANopen NMT state machine and error conditions of CAN controller.

7.1.1 Signalling of CANopen NMT state

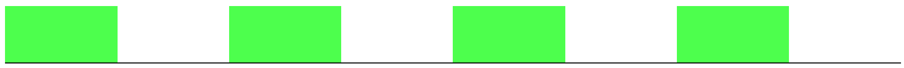
The green LED displays the state of CANopen Network Management (NMT) .



Initialisation (Autobaud Detection)



NMT Status: Device in "Stopped" state

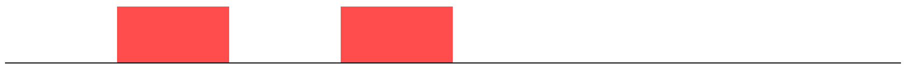


NMT Status: Device in "Pre-operational" state

NMT Status: Device in "Operational" state

7.1.2 Signalling of CAN controller state

The red LED is signalling the status of the CAN controller. Only in fault condition the red LED will show the status.



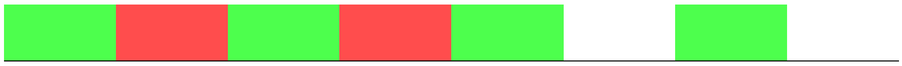
CAN Status: Controller in "Warning" state

CAN Status: Controller in "Error Passive" state

CAN Status: Controller in "Bus-Off" state

7.1.3 Combined signalling of NMT and CAN State

In combination there will be shown the network status and the controler status.



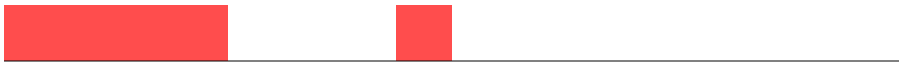
Device in "Pre-operational" state, CAN Controller in "Warning" state

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

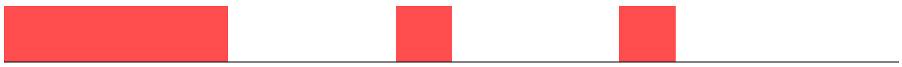
7.2 State of µCAN module

The LED marked with Module Status" (on the case cover denoted as Error) displays the status of the device hardware.

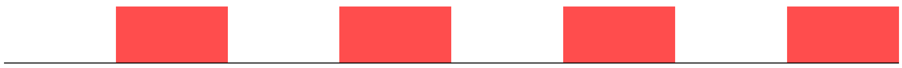
Modul Status: Function/Power OK (No short circuit / overload)



Modulstatus: Wrong setting of Baudrate DIP switches



Modulstatus: Wrong setting of Address DIP switches



Modulstatus: Sensor break

Please note that default sensor type is set to thermocouple J.

8. CANopen Protocol

This chapter provides detailed information on how to connect the modules of the µCAN-series to a CANopen-Manager and set into operation. A CANopen-Manager can be a PLC, a PC with a CAN interface or any other CAN-Device with CANopen network managment functionality.

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

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

8.1 General Information

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

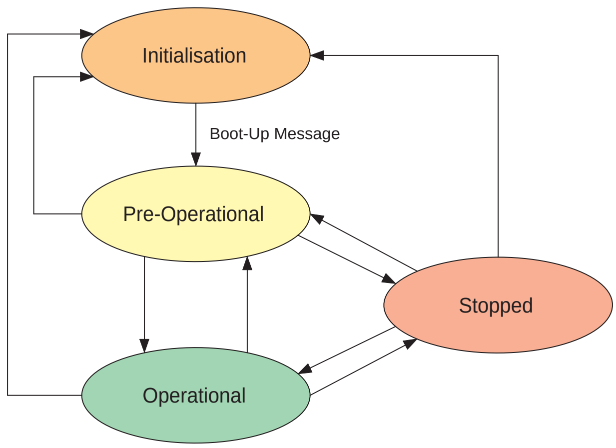
Object	COB-ID (dec.)	COB-ID (hex)
Network Management	0	0x000
SYNC	128	0x080
EMERGENCY	129 - 255	0x081 - 0x0FF
PDO 1 (transmit)	385 - 511	0x181 - 0x1FF
PDO 2 (transmit)	641 - 767	0x281 - 0x2FF
SDO (transmit)	1409 - 1535	0x581 - 0x5FF
SDO (receive)	1537 - 1663	0x601 - 0x67F
Heartbeat / Boot-up	1793 - 1919	0x701 - 0x77F

Table 5: Identifier values according to the Pre-defined Connection Set

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

8.2 Network Management

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



Start Node

Start Node

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

Node = module address, 0 = all modules

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

Stop Node

Stop Node

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

Node = module address, 0 = all modules

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

Pre-Operational

Enter Pre-Operational

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

Node = module address, 0 = all modules

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

Reset Node

Reset Node

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

Node = module address, 0 = all modules

By transmitting the „Reset Node“ command the CAN-node will issue a reset operation. After reset the node will send a "Boot-up Message" (refer to “Heartbeat Protocol” on page 73) and enter the Pre-operational state automatically.

8.3 SDO-Communication

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

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

The Command Byte (**CMD**) has the following meaning:

SDO-Client (Master)	**SDO-Server (Slave)**	**Funktions**
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 message

8

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

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

8.3.1 SDO Abort Protocol

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

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
	8	80h	Index		Sub-Index	Abort code			

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

The abort code may have the following values:

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

Table 7: SDO abort codes

8

8.4 Object Dictionary

This chapter describes the implemented objects for the module µCAN.4.ti-BOX. For further information on the objects please refer to the CANopen communication profile CiA 301 and the device profile CiA 404.

EDS

The implemented objects of the module µCAN.4.ti-BOX are described in an "Electronic Data Sheet" (EDS). The EDS file is available on the MicroControl Homepage.

8.4.1 Communication Profile

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

Index	Name
1000h	Device Profile
1001h	Error Register
1002h	Manufacturer Status Register
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 communication profile

Index	Name
1800h	1st Transmit PDO Parameters
1801h	2nd Transmit PDO Parameters
1A00h	1st Transmit PDO Mapping
1A01h	2nd Transmit PDO Mapping
1F80h	NMT Startup

Table 8: Supported objects of the communication profile

8

Device Profile

Index 1000h

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

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

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

Example: read parameter, node-ID = 2, index = 1000h

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

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

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

Byte 5 + Byte 6 = 0194h = 404d (Device Profile Number)
Byte 7 + Byte 8 = 0002h = 2 (Additional Information)

Error Register

Index 1001h

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

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

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

Example: read parameter, node-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 response the µCAN.4.ti-BOX will return its error register value.

The following error types are supported:

B4	**Description**
08h	Temperature Error: active when an error occurs while temperature measurement.
10h	Communication Error: active when error occurs on CAN network. More detailed information are given in chapter "Emergency Message" on page 85.

Table 9: Supported error types

Note: Bytes 5 to 7 always have the value 00h.

Manufacturer Status Register

Index 1002h

Via index 1002h it is possible to read the manufacturer status register of the device.

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned32	ro	Manufacturer Status Register	00h

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

Example: read parameter, node-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 response the µCAN.4.ti-BOX will return the manufacturer status register value.

Manufacturer status register gives informations about ADC (Analog to Digital Converter) and EEPROM (Electrically Erasable Programmable Read-Only Memory).

Possible register values are described in the following table.

B4	B5	B6	B7	Description
01h	00h	00h	00h	EEPROM error: communication with EEPROM
02h	00h	00h	00h	EEPROM error: write access to EEPROM failed
10h	00h	00h	00h	ADC1 error: no communication to ADC 1
20h	00h	00h	00h	ADC1 stopped: first ADC (channel 1 and 2) is stopped.
00h	01h	00h	00h	ADC2 error: no communication to ADC 2
00h	02h	00h	00h	ADC 2 stopped: second ADC (channel 3 and 4) is stopped.

Table 10: Manufacturer Status Register values

Predefined Error Register

Index 1003

The object at index 1003h holds the errors that have occured on the device. The object stores a maximum of 4 error conditions.

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

The object supports the sub-indices 0 to 4. An access to other sub-indices will lead to an error message. Writing to sub-index 0 will clear the error history.

Example: read parameter, node-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 response the µCAN.4.ti-BOX will return the error value at position 3 in the history.

Manufacturer Device Name

Index 1008

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

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

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

Hardware Version

Index 1009h

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

Sub-Index	Data Type	Acc.	Name	Default Value
0	Visible String	ro	Hardware version	4.02

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

Software Version

Index 100Ah

The object at index 100Ah contains the manufacturer software version

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

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

Store Parameters

Index 1010h

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

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

In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-index. The signature is "*save*" (in ASCII).

Example: save all parameters, node-ID = 2, index = 1010h

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

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

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

Restore Default Parameters

Index 1011h

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

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

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

Beispiel: restore all parameters, node-ID = 2, Index = 1011h

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

As response the µCAN.4.ti-BOX 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

The object at index 1014h defines the COB-ID of the emergency message.

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

Only sub-index 0 is supported. An access to other sub-indices will lead to an error message. The default value for COB-ID EMCY is 80h + node-ID (1 to 127).

Identity Object

Index 1018h

The object at index 1018h provides general identification information of the device.

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	04h
1	Unsigned32	ro	Vendor ID	0000 000Eh
2	Unsigned32	ro	Product Code	--
3	Unsigned32	ro	Revision Number	--
4	Unsigned32	ro	Serial Number	--

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

Vendor ID

The Vendor ID contains a unique value allocated to each manufacturer. The numbers are managed by the CAN in Automation (CiA). Vendor ID 0x0000000E is allocated to *MicroControl GmbH & Co. KG*.

Product Code

The Product Code identifies a specific device version.

Revision Number

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

Serial Number

The Serial Number identifies a specific device.

Error Behaviour

Index 1029h

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

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

The following codes are possible:

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

Table 11: Codes for error behaviour setup

The device detects the following communication errors:

- Bus-off conditions of the CAN interface
- Life guarding event with the state "occurred" and the reason "time out"
- Heartbeat event with state "occurred" and the reason "time out"

NMT Startup

Index 1F80h

The object at index 1F80h defines the NMT Startup behaviour of the µCAN module.

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

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

The NMT startup behaviour after Power-Up or Reset-Node can be changed by this index. Following values are supported:

Value	Behaviour description
00h	Default behaviour, change to Pre-Operational
02h	Send NMT "Start All Nodes"
08h	Change to NMT state Operational

8.4.2 Device Profile CiA 404

In this section you will find all device profile specific indices for the µCAN.4.ti-BOX. These indices are implemented according to the CiA 404 device profile.

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
7130h	AI Process Value

Table 12: Supported objects of device profile CiA 404

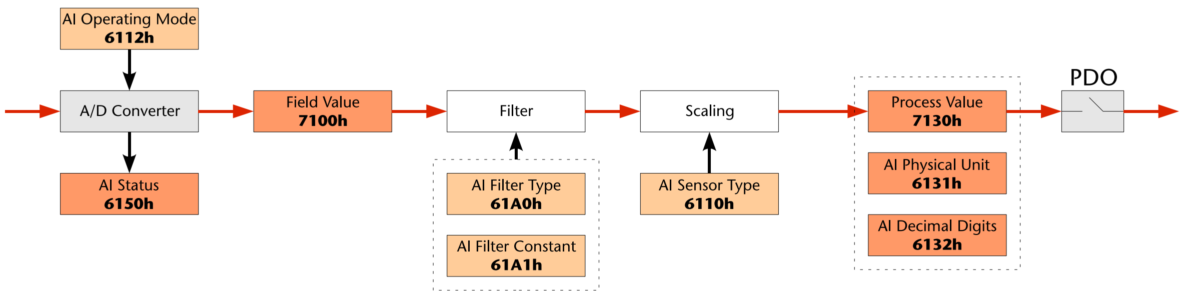


Fig. 17: Block diagram of an input channel

8

AI Sensor Type

Index 6110h

Index 6110h specifies the type of sensor which is connected to the analogue input of the µCAN.4.ti-BOX.

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	number of entries	04h
1	Unsigned16	rw	AI Sensor type of Channel 1	0001h
2	Unsigned16	rw	AI Sensor type of Channel 2	0001h
3	Unsigned16	rw	AI Sensor type of Channel 3	0001h
4	Unsigned16	rw	AI Sensor type of Channel 4	0001h

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

Following table list supported values for sensor type:

Value	Sensor type
01h	Thermocouple J
02h	Thermocouple K
05h	Thermocouple R
07h	Thermocouple T
1Eh	PT100
1Fh	PT200
20h	PT500
21h	PT1000

Table 13: Supported sensor types

Other temperature sensors are available on request.

Example: Read channel 1 sensor (sub-index 1), node-ID = 2

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

As response the µCAN.4.ti-BOX will send

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

The response message in this example shows the sensor type value 01h (byte 4). This means a thermocouple type J is configured as input signal.

Setting a new sensor type always affects two input channels, i.e. channel 1 / 2 and channel 3 / 4 always have the same sensor type.

Example: Configure channel 1 and 2 for Pt100, node-ID = 2

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
602h	8	2Bh	10h	61h	01h	1Eh	00h	00h	00h

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

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

Other values than listed in table 13 on page 58 will lead to an SDO abort message (refer to “SDO Abort Protocol” on page 44).

Storing of sensor type will not be done automatically. Please issue the "Save all" or "Save application" command to store the sensor type in non-volatile memory (refer to “Store Parameters” on page 52).

AI Operating Mode

Index 6112h

The operating mode of each channel can be configured via index 6112h.

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	number of entries	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

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

Writing the value "0" switches the channel off, writing the value "1" switches the channel on (factory default).

Example: Switch channel 3 off, node-ID =2

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

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

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

Channel 3 is now disabled, the process value is set to 0. In case there was an error active on this channel, the error status is cleared.

Storing of operating mode will not be done automatically. Please issue the "Save all" or "Save application" command to store the operating mode in non-volatile memory (refer to "Store Parameters" on page 52).

AI Physical Unit Process Value

Index 6131h

By a read-access on index 6131h the physical unit of the process value (PV) can be requested. This object is read-only and has the following structure:

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	number of entries	04h
1	Unsigned32	ro	AI Physical Unit PV of Channel 1	002D0000h
2	Unsigned32	ro	AI Physical Unit PV of Channel 2	002D0000h
3	Unsigned32	ro	AI Physical Unit PV of Channel 3	002D0000h
4	Unsigned32	ro	AI Physical Unit PV of Channel 4	002D0000h

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

Example: Read physical unit for channel 3, node-ID=2

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

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

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	31h	61h	03h	00h	00h	2Dh	00h

8

The returned value is 002D0000h, which corresponds to the unit degree celsius (°C). A complete list of possible physical units is available in the CiA 302-2 document.

AI Decimal Digits Process Value

Index 6132h

By a read access on index 6132h the number of decimal digits of the process value (PV) can be requested. This object is read-only and has the following structure:

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	04h
1	Unsigned8	ro	AI Decimal Digits PV of Channel 1	01h
2	Unsigned8	ro	AI Decimal Digits PV of Channel 2	01h
3	Unsigned8	ro	AI Decimal Digits PV of Channel 3	01h
4	Unsigned8	ro	AI Decimal Digits PV of Channel 4	01h

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

Example: Read decimal digits for channel 3, node-ID=2

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

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

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	40h	32h	61h	03h	01h	00h	00h	00h

8

The module returns the value 01h, i.e. process values are communicated with 1 decimal digit (refer to “AI Process Value” on page 67).

AI Status

Index 6150h

By a read access on index 6150h the status of each channel can be requested. This object is read-only and has the following structure:

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	04h
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

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

The follwing table lists possible bit-coded status values:

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

Table 14: Possible status values for each channel

Example: Read status for channel 1, node-ID=2

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	40h	50h	61h	01h	00h	00h	00h	00h

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

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	4Bh	50h	61h	01h	03h	00h	00h	00h

In this example the module returns the status value 03h, i.e. a positive overload has occured (bit 2 set to ’1’) and the process value is not valid (bit 1 set to ’1’).

8

AI Filter Type

Index 61A0h

The filter type on each channel can be configured via index 61A0h.

Sub-Index	Data Type	Acc.	Name	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 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

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

Following values are possible for the filter type:

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

Table 15: Filter types

Other filter types are available on request.

Storing of filter type will not be done automatically. Please issue the "Save all" or "Save application" command to store the filter type in non-volatile memory (refer to “Store Parameters” on page 52).

AI Filter Constant

Index 61A1h

The filter constant on each channel can be configured via index 61A1h.

Sub-Index	Data Type	Acc.	Name	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

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

The filter constant value has a range from 1 to 50. Writing other values will lead to an error message.

Example: Write filter constant 5 on channel 3, node-ID=2

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

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

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 correct filter type is selected via index 61A0.

Storing of filter constant will not be done automatically. Please issue the "Save all" or "Save application" command to store the filter constant in non-volatile memory (refer to “Store Parameters” on page 52).

AI Field Value

Index 7100h

Index 7100h holds the field value of each channel. The field value is the converted value of the internal A/D converter. The value can be already filtered (objects 61A0h and 61A1h), but there has been no linearisation for the selected sensor type..

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	04h
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

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

Example: read A/D value for channel 3, node-ID=2

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

A possible response of the µCAN.4.ti-BOX might be:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	4Bh	00h	71h	03h	11h	0Ah	00h	00h

The actual converted analogue value of the A/D converter is 0A11h.

Reading the values of all 4 channels at the same time is possible via PDO 2 (refer to “Transmit PDO 2” on page 80).

AI Process Value

Index 7130h

Index 7130h holds the linearised process value for each channel. The linearisation depends on the selected sensor type (refer to "AI Sensor Type" on page 58). The index has the following structure:

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	04h
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

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

Example: Read process value on channel 3, node-ID=2

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

A possible response of the µCAN.4.ti-BOX might be:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	4Bh	30h	71h	03h	45h	03h	00h	00h

The actual converted process value is 0345h = 837d = 83,7°C.

Reading the values of all 4 channels at the same time is possible via PDO 1 (refer to "Transmit PDO 1" on page 79).

In case of a sensor failure the process value is set to EEEEh = -4370d = -437,0°C. The AI Status of the channel is set to the appropriate value (refer to "AI Status" on page 63). In addition an Emergency message is transmitted (refer to "Emergency Message" on page 85).

8.4.3 Manufacturer Specific Objects

Within this chapter the manufacturer specific objects of the µCAN.4.ti-BOX can be found.

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

Table 16: Manufacturer specific objects

Customer Data

Index 2010h

By means of the index 2010h the customer can store up to 8 words (32 bit) of data to the non-volatile memory of the device.

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	08h
1	Unsigned32	rw	Customer Data 1	-
2	Unsigned32	rw	Customer Data 2	-
..	..	..	..	..
8	Unsigned32	rw	Customer Data 8	-

Sub-Indices from 0 to 8 are supported. An access to other sub-indices will lead to an error message.

On writing to the sub-indices 1 to 8 the customer data will automatically be stored on EEPROM. It is not required to issue the Store Parameters command (refer to “Store Parameters” on page 52).

COB-ID Storage

Index 201Ah

The contents of this object controls the behaviour of the identifiers from the "Predefined Connection Set" when changing the node-ID. This effects the bahviour of identifiers such as PDO-ID or EMCY-ID.

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

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

The following values are supported:

Value	Meaning
00h	Stored COB-IDs from PDO/EMCY will remain unchanged after change of module address
01h	Stored COB-IDs from PDO/EMCY will fall back to default Pre-defined Connection Set when changing module address
02h	Stored COB-IDs from PDO/EMCY will be calculated as "Storded COB-ID" + module address

The object 201Ah will have an direct effect on the use of the objects 1014h, 1800h, 1801h, and 1010h.

PDO Data Format

Index 2E00h

By means of this object the byte order in a PDO can be changed. Supported are the Intel (Little-Endian) oder Motorola (Big-Endian) formats.

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

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

The following values are supported:

Value	Meaning
00h	PDO data will be send in Intel-Format (default)
01h	PDO data will be send in Motorola-Format

Disable Boot-Up Message

Index 2E10h

In some applications it might be useful to disable the transmission of the "Boot-Up Message". This can be done by means of the object 2E10h.

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

8

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

The following values are supported:

Value	Meaning
00h	Boot-Up message will be send after power up or reset of node (default)
01h	Transmission of Boot-Up message is suppressed

Bus Statistic

Index 2E22h

By means of the object 2E22h the CAN bus statistics of the module can be read.

Sub-Index	Data Type	Acc.	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	-

Sub-Indices from 0 to 3 are supported. An access to other sub-indices will lead to an error message.

All sub-indices are read-only. The values relect the number of transmitted and received messages as well as number of CAN errors. All values have an overflow to zero.

8.5 Device Monitoring

For device monitoring CANopen provides two mechanisms (protocols):

- heartbeat
- node guarding

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

8.5.1 Heartbeat Protocol

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

Heartbeat ID

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

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

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

Table 17: Status Information for Heartbeat

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

Example: Power-on of module with address 2

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

Consumer heartbeat time

Index 1016h

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

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

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

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

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

Producer heartbeat time

Index 1017h

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

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

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

Example: Producer time 1000 ms, module address 1

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

The answer you will receive from the module is:

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

The heartbeat producer time is not saved inside the non-volatile memory autonomously. It is necessary to store this parameter via object 1010h (refer to "Store Parameters" on page 52).

8.5.2 Node Guarding

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

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

Guard time

Index 100Ch

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

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

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

Life time factor

Index 100Dh

8

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

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

The value 00h disables the life guarding.

8.6 PDO-Communication

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

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

8.6.1 Transmission Modes

Event Driven

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

Timer Driven

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

8.6.2 Transmit PDO 1

Index 1800h

The object at index I1800h defines communication parameters for the Transmit PDO 1.

Sub-Index	Data Type	Acc.	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

Only sub-indices 0 to 2 and 5 are supported. An access to other sub-indices will lead to an error message.

COB-ID for PDO

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

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

Table 18: Definition of COB-ID for PDO

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

8

Transmission Type

The transmission type defines the transmission character of the PDO.

Transmission Type	Description
00h	acyclic synchronous, µCAN module considers each SYNC message
01h - F0h (1 - 240 dec.)	cyclic synchronous, µCAN module considers only nth SYNC message
FFh (255 dec.)	event driven,, PDO is sent when Event Timer elapses

Table 19: Setup of Transmission Type

The Transmit PDO has 8 byte of process data. The contents is copied from object 7130h, sub-index 1 to 4 (refer to “AI Process Value” on page 67) into the PDO.

8.6.3 Transmit PDO 2

Index 1801h

The object at index I1801h defines communication parameters for the Transmit PDO 2.

Sub-Index	Data Type	Acc.	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

Only sub-indices 0 to 2 and 5 are supported. An access to other sub-indices will lead to an error message.

COB-ID for PDO

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

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

Table 20: Definition of COB-ID for PDO

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

8

Transmission Type

The transmission type defines the transmission character of the PDO.

Transmission Type	Description
00h	acyclic synchronous, µCAN module considers each SYNC message
01h - F0h (1 - 240 dec.)	cyclic synchronous, µCAN module considers only nth SYNC message
FFh (255 dec.)	event driven, PDO is sent when Event Timer elapses

Table 21: Setup of Transmission Type

The transmit PDO has 8 byte of field value data. The contents is copied from object 7100h, sub-index 1 to 4 (refer to “AI Field Value” on page 66) into the PDO.

8.6.4 Transmit PDO 1 Mapping

Index 1A00

The object at index 1A00h defines the mapping parameters for PDO 1.

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

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

Each entry defines an oobject which is transmitted with PDO 1. The entry has the following structure:

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

Table 22: Structure of mapping entry

8

8.6.5 Transmit PDO 2 Mapping

Index 1A01h

The object at index 1A01h defines the mapping parameters for PDO 2.

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

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

Each entry defines an oobject which is transmitted with PDO 2. The entry has the following structure:

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

Table 23: Structure of mapping entry

8

8.6.6 Transmit PDO Example

Both transmit PDOs are configured to Transmission Type 1 (cyclic, SYNC message) by default. Hence transmission of the PDOs is triggered by a SYNC message (index 1005h).

Example: node-ID=1, send SYNC

ID	*DLC*
80h	0

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

<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="2">Index 7130h, Sub 01h</td><td colspan="2">Index 7130h, Sub 02h</td><td colspan="2">Index 7130h, Sub 03h</td><td colspan="2">Index 7130h, Sub 04h</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>182h</td><td>8</td><td colspan="2">Index 7100h, Sub 01h</td><td colspan="2">Index 7100h, Sub 02h</td><td colspan="2">Index 7100h, Sub 03h</td><td colspan="2">Index 7100h, Sub 04h</td></tr>
</table>

Transmission of PDOs is only possible in Operational mode of the device (refer to “Start Node” on page 41).

8.7 Synchronisation Message

Index 1005h

The object at index 1005h defines the identifier for the SYNC-message. On reception of a message with this identifier the transmission of PDOs is triggered (refer to "Transmit PDO 1" on page 79).

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

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

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

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

As answer you will get the following message:

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

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

The SYNC-identifier is not saved inside the non-volatile memory autonomously. It is necessary to store this parameter via object 1010h (refer to "Store Parameters" on page 52)

8.8 Emergency Message

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

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

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

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

The following emergency error codes are supported:

Error Code	Description
0000h	Error reset or no error
5030h	sensor fault
8100h	CAN controller entered "warning" state
8110h	CAN controller overrun
8120h	CAN controller entered "error passive" state
8130h	heartbeat event / node guarding event
8140h	device recovered bus-off
8150h	identifier collision (Tx-ID reception)

Table 24: Emergency error codes

The field „ER" (error register) of the Emergency message is a copy of the CANopen object 1001h.

8

9. Technical Data

Power Supply	
Supply Voltage	9..36V DC, reverse current 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	Fieldbus/Supply: 500 Veff
Physical Interface	Terminal Block (2,5 mm^2)

CAN Bus	
Baudrates	20 kBit/s .. 1 MBit/s
Status on the bus	active node
Protocol	CANopen CiA 301 V4.02, CiA 404 V1.02
Physical Interface	Terminal Block (2,5 mm^2)

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	
Operating temperature	-40°C bis +85°C
Signal type	Resistance thermometers Pt100, Pt200, Pt500, Pt1000 Thermocouples Type J, Type K, Type R, Type T
Resolution	16 Bit
Sample rate	100 Hz on each channel

Housing	
Aluminium die cast	EN AC-44300 DIN EN 1706 (GD Al Si 12 / DIN 1725)
Protection class	IP 66 / EN 60529
Finishing	standard coating powder color RAL 7032, RAL 7001 stoved enamel coating
Dimensions	125 * 80 * 57 mm (l * w * d) without cable glands / connectors
Weight	540 g
Weight incl. cable glands	640 g

Index

P

S

T

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

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

MicroControl
Systemhaus für Automatisierung

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