MicroControl
Systemhaus für Automatisierung GmbH & Co. KG

µCAN.sensor

Manual Analogue Data Acquisition for OEM-customers
Version 1.02

MicroControl GmbH & Co. KG
Lindlaustraße 2 c
D-53842 Troisdorf
Fon: 02241 / 25 65 9 - 0
Fax: 02241 / 25 65 9 - 11
http://www.MicroControl.net

Index

µCAN.sensor

Index

µCAN.sensor

Warranty Limitations

µCAN.sensor

Remarks on CE-conformance of µCAN-modules

µCAN-modules which have CE-conformance label, have passed test specifications of EU-criteria 89/336/EWG "Electromagnetic Emission and Immunitiy" and standardized European norms (EN).
Papers of declaration for EU-conformance, according to Art.10 of EN, are available at:

MicroControl GmbH & Co. KG
Lindlaustr. 2 c
D-53842 Troisdorf
Germany

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.

1. Operation of µCAN.sensor

1.1 Overview

µCAN.sensor

The µCAN.sensor empowers the OEM customer to inerface a wide range of analogue signals directly to the CAN bus. The µCAN.sensor is a cost sensitive and fast solution for your needs of analogue to CAN interfaces.

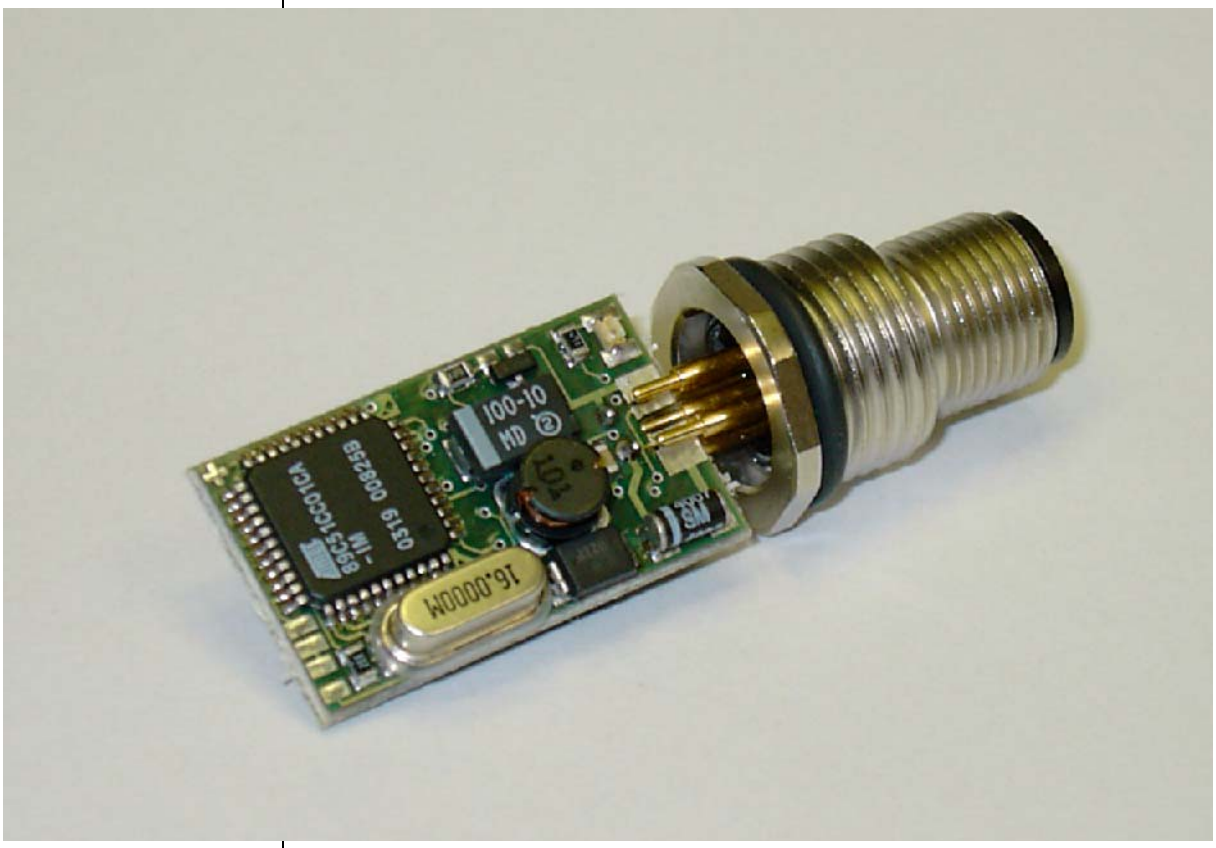


Fig. 1: µCAN.sensor with M12 connector

Typical signals to be interfaces by the µCAN.sensor are temperature sensors directly from thermosouples, Pt100-temperature resistors, +/-10VDC, 0(4)..20mA and strain gauge signals. The strain gauge is powered direct out of the module and cold junction compensation for thermocouple is also integrated.

The main application fields are automation systems, mobile applications, food industry and environmental analysis systems.

Operation of µCAN.sensor

Overview

The µCAN.sensor is equipped with the higher layer protocol

CANopen

according to DS-404. Other protocol stacks are available on request.

space saving and compact

The µCAN.sensor is designed for heavy duty applications. Due to small size of pcb the hardware can be integrated in many different applications. Also available in steel-housing ensures high protection class IP67. 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.sensor in your application reduces the development effort. Costs for material and personnel are reduced. The easy installation makes maintenance and replacement quite simple.

2. Address- and Bitrate-Setting, Termination

2.1 Address- and Bitrate-Setting

The module address as well as the bitrate can only be changed by CAN bus interface. The protocol to change is the so called Layer-Setting services ("LSS"). Protocol is standarized and can be found in DS-

2.2 Termination

When using several modules on the bus make sure to terminate the CAN bus at the starting point and at the end with a resistor of 120Ohm. This ensures to run the CAN bus without physical disturbances like reflections.

To terminate the bus simply connect a resisitor of 120Ohm between the CAN-H and CAN-L bus line. When using the M12 connectors this can easiliy be done by using MicroControl´s M12-termination-connectors.

Note

Modules used as T-modules must not be terminated. Please make sure not to multiple terminate the bus. When measuring the resistance between CAN-H and CAN-L (in powered down network) you will get a total value of about 60Ohm.

3. Analogue Input

3.1 General information

This chapter shows the different signal types and connection of signal lines to the µCAN.sensor. Ensure to use proper wiring and keep in mind EMI-rules. Other devices used in conjuction have to meet the safety regulations and EMI requirements. Only if these basic rules are fullfilled, the µCAN.sensor will deliver stable and high quality measuring signals.

Naming of input channel

The µCAN.sensor incorporates one analogue input channel. The inpur channel consists of 4 solder pads. In the following table and schematic the 4 pads are defined.

Name	Signal type
P1	Sensor supply, Pos. chain
G1	Sensor supply, Neg. chain
+	Signal +
-	Signal -

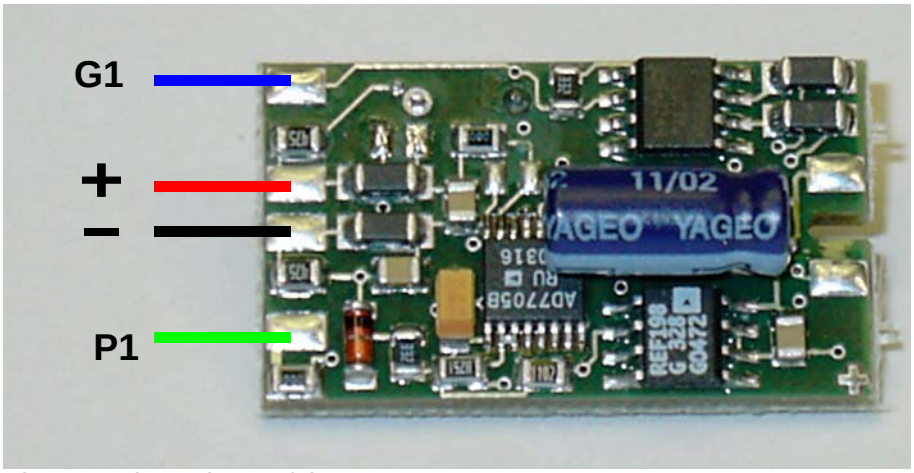


Fig. 2: Analogue input of the µCAN.sensor

On this picture the rear side of the pcb is seen.

Attention !

Please ensure to connect signal lines only if powered down the module. In any other case destruction of hardware may occur.

3.2 Temperature signals

3.2.1 Connection of temperature resisitors

As mentioned before the µCAN.sensor can diretcly handle thermocouple signals as well as temperature resistors.

Connection of Pt100/Pt1000 or any other temperature resisitor is shown in the following schematic.

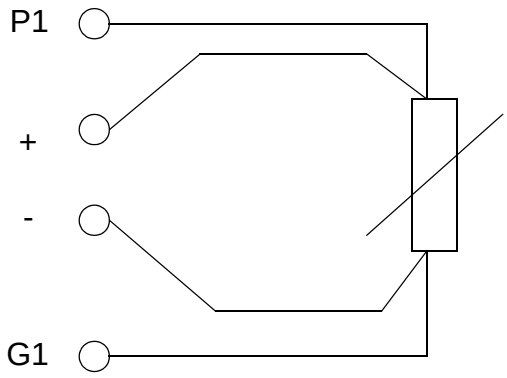


Fig. 3: Connecting temperature resisitors

3.2.2 Connection of thermocouples

Using thermocouple signals is as easy as using the temperature resisitors. Simply connect the signal chains according to the polarity to the "+" and "-" input pad.

In any case of polarity mismatch you will get negative measuring values at room temperature.

The following schematic shows the thermocouple connection of signal chains.

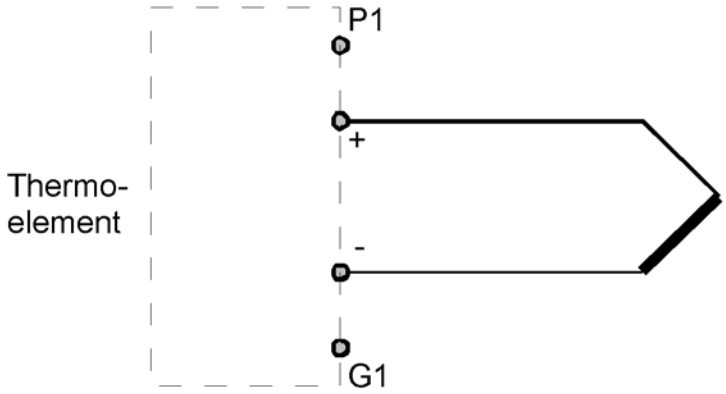


Fig. 4: Connecting thermocouples

Note

An automated cold junction compensation takes place on the pcb. The temperatures seen on the CAN bus are completely linearized and compensated.

3.3 Connection of strain gauge

When using strain gauges the bridge can be powered out of the µCAN.sensor. There is an constant voltage supply of 2.5V DC.

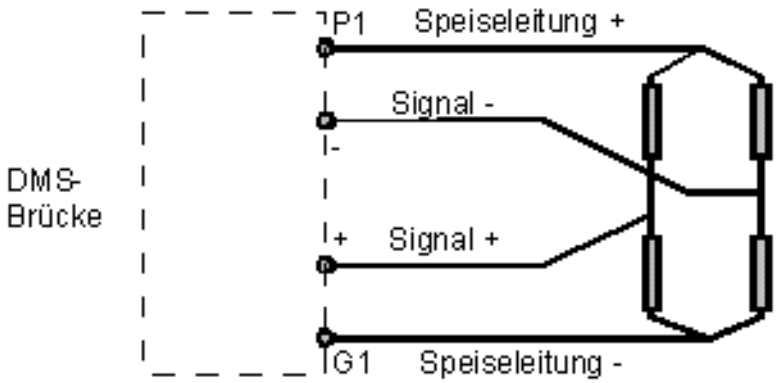


Fig. 5: Connecting strain gauges

3.4 Standard signals

Standard signals like +/-10V DC or 0(4)..20mA can be directly connected to the "+" and "-" solder pads. Please make sure not to confues the poles.

4. Connection of supply voltage and CAN

4.1 Supply voltage

The µCAN.sensor is designed for standard industry applications. The power supply is optimized for supply voltage of 24V DC.

The wide supply input can handle voltages in the range of 9V DC up to 36V DC. For more detaild information please refer to the technical specification.

The supply input is protected against reverse voltage.

4.1.1 Pinout for power supply and CAN bus

Power supply has to be connected to the pins V+ amd GND. Where V+ is the positive chain of the power supply and GND the negative chain.

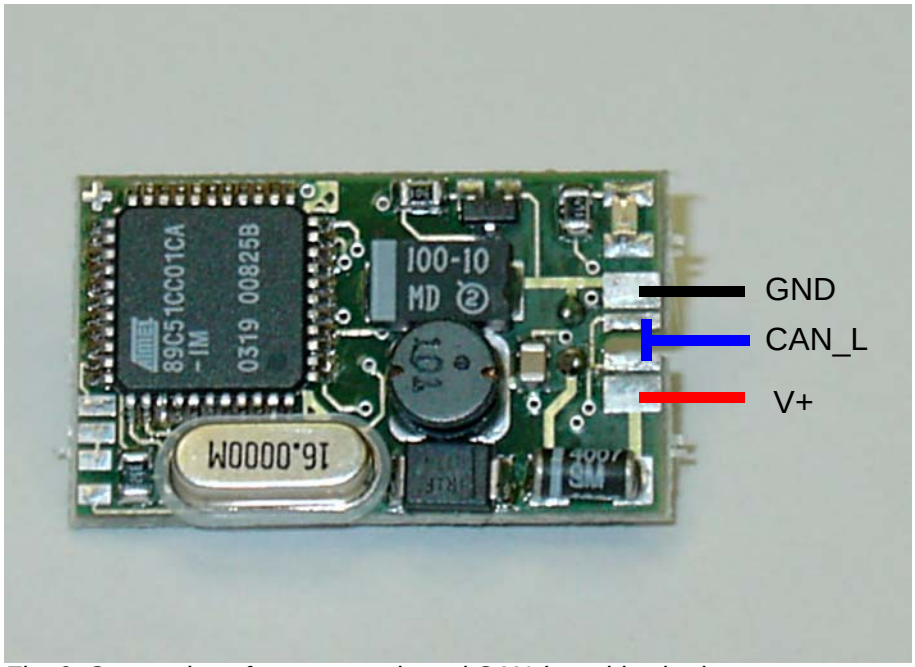


Fig. 6: Connecting of power supply and CAN (top side view)

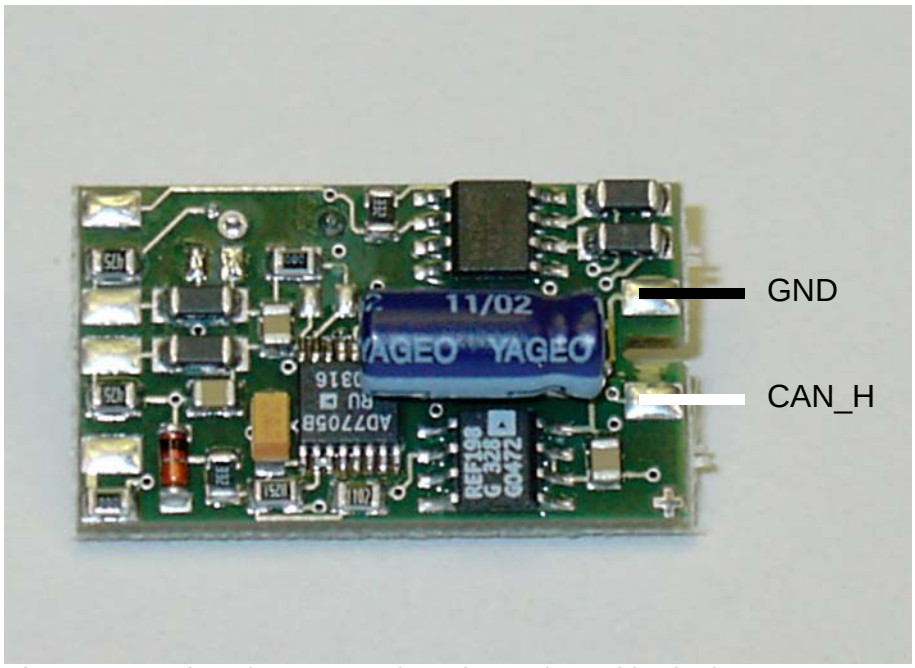


Fig. 7: Connecting of power supply and CAN (rear side view)

Attention !

Possible shielding may not be led into the housing. Also the shield may not be connected to any of the solder pads of the pcb. Please make sure to use proper shielding outside the housing.

4.2 CAN bus line

The CAN bus will be connected with 2 lines direct to the solder pads. Please refer to figure 6 and figure 7.
To avoid disturbances of the analogue input make sure not to lead the bus lines over or close to the analogue input.

4.3 Pinout for M12-connectors

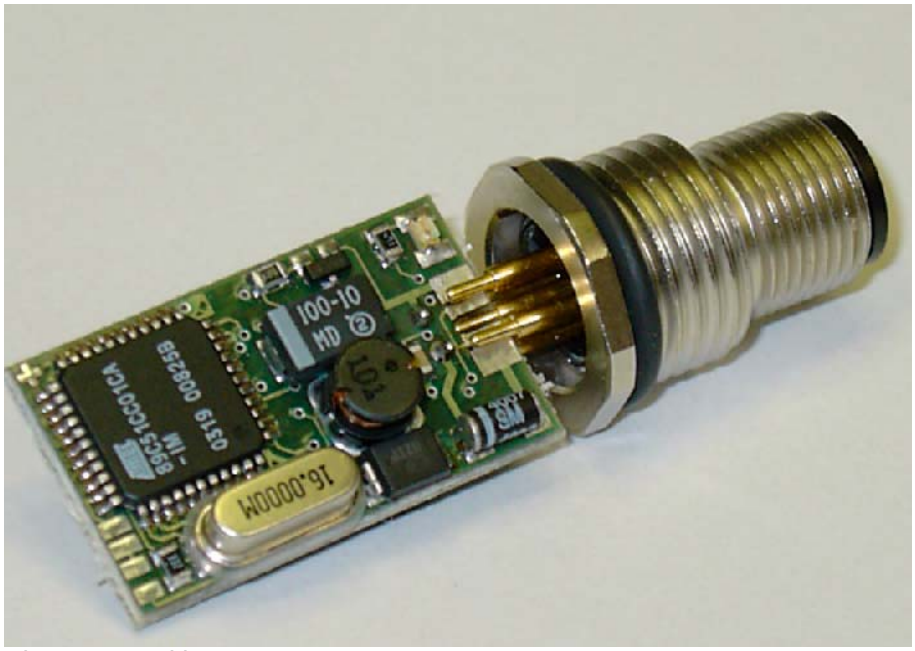


Fig. 8: PCB with M12-connector

In case the module is equipped with a M12-connector (male) the connector has the standarized (CANopen) pinout. The pinout is defined in DS-303 specification sheet and can be seen in the following figure.

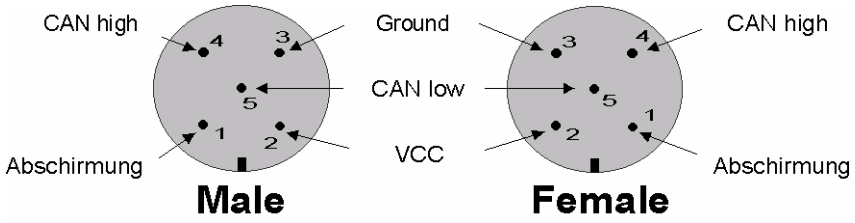


Fig. 9: M12-connector

5. Diagnosis

5.1 Diagnosis LED

The µCAN-modules are equipped with a status information LED. This status information provides the service personnel with fault-condition and CAN bus information.
LED can be found on the pcb like seen in the following figure.

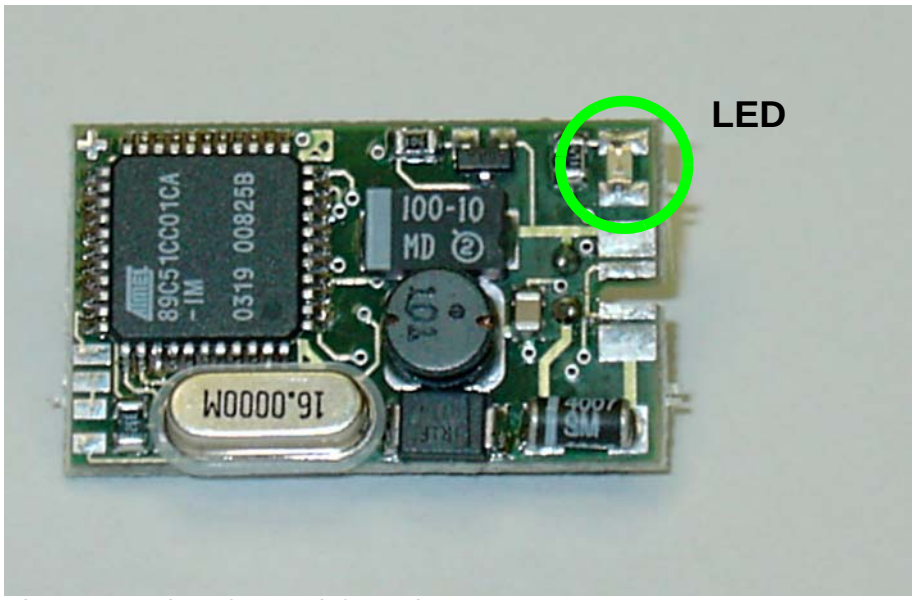


Fig. 10: Location of status information LED

5.2 µCAN.sensor - Diagnosis

After power up of the device the LED will flash for two times and then show one of the following status information:

Status	Meaning
blinking	Device is Pre-Operational (CANopen)
constant OFF	No power supply / hardware is defective
constant ON	Device is Operational (CANopen)
quick flash-light	Device is Stopped (CANopen)

6. CANopen Protocol

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

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

This documentation provides the actual implemented functions and services of the µCAN.sensor.

6.1 Introduction

The identifiers of the µCAN.sensor are set up according to the **Pre-defined Connection Set**, which is described in the CANopen communication profile DS-301 in detail. 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 1 (Receive)	513 - 639	0x201 - 0x27F
SDO (Transmit)	1409 - 1535	0x581 - 0x5FF
SDO (Receive)	1537 - 1663	0x601 - 0x67F
Heartbeat / Boot-Message	1793 - 1919	0x701 - 0x77F

Table 1: Identifier values according to the Pre-defined Connection Set

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

6.2 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" (siehe “Heartbeat Protocol” auf Seite 27) and enter the Pre-operational state automatically.

6.3 SDO Communication

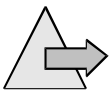
All parameters of the devices (organized in an object dictionary) are accessed via the SDO service (Service Data Object). A SDO message has the following contents:

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

For calculation of the SDO message identifier please siehe “Introduction” auf Seite 14.

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

Master wants to read from Slave	40h
Slave answers on the read-request	42h
Master wants to write to Slave	22h
Slave answers on the write-request	60h



Note

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

Attention !

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

6.3.1 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:

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

Table 2: SDO abort codes

6.4 Object Dictionary

This chapter describes the implemented objects for the digital module µCAN.sensor. For further information please refer to the CANopen communication profile DS-301 and the device profile DS-401.

6.4.1 Communication Profile

The module µCAN.sensor supports the following objects from the communication profile DS-301:

Index	Name
1000h	Device Profile
1001h	Error 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
1017h	Heartbeat Producer Time
1018h	Identity Object
1800h	1st Transmit PDO Parameters
1801h	2nd Transmit PDO Parameters
1A00h	1st Transmit PDO Mapping
1A01h	2nd Transmit PDO Mapping

Table 3: Supported objects of the communication profile

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	0003 0191h

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

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

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

As response the µCAN.sensor will send:

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
582h	8	42	00	01h	00	94h	01h	02h	00

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, module ID = 2, Index = 1001h

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

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

Generic Error

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

Communication Error

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

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

Pre-defined Error Field

Index 1003

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

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

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

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

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

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

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	µCAN.8.dio

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

Manufacturer Hardware Version

Index 1009h

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

Sub-Index	Data Type	Acc.	Name	Default Value
0	Visible String	ro	Hardware version	HW-1.1

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

Manufaturer Software Version

Index 100Ah

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

Sub-Index	Data Type	Acc.	Name	Default Value
0	Visible String	ro	Software version	SW-1.0

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	3
1	Unsigned32	rw	Save all parameters	1
2	Unsigned32	rw	Save communication	1
3	Unsigned32	rw	Save application	1

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".

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

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	22h	10h	10h	01h	73h	61h	76h	65h

As response the µCAN.sensor 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	3
1	Unsigned32	rw	Restore all param.	1
2	Unsigned32	rw	Restore commun.	1
3	Unsigned32	rw	Restore application	1

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

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

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
602h	8	22h	11h	10h	01h	6Ch	6Fh	61h	64h

As response the µCAN.sensor will send:

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

Identity Object

Index 1018h

The object at index 1018h contains general information about the device.

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	4
1	Unsigned32	ro	Vendor ID	0000 000Eh
2	Unsigned32	ro	Product Code	0013 9F70h
3	Unsigned32	ro	Revision Number	0001 0000h
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 (http://www.can-cia.org).

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.

Guard Time

Index 100Ch

Über den Index 100Ch kann die Guard Time für das Node-Guarding Protokoll eingestellt werden. Die Multiplikation der Werte auf den Indicies 100Ch und 100Dh ergibt die "Life Time" eines Knotens.

Sub-Index	Datentyp	Zugriff	Bedeutung	Defaultwert
0	Unsigned16	rw	Guard Time	0

Das Objekt kann gelesen und geschrieben werden. Es wird nur Sub-Index 0 unterstützt. Ein Zugriff auf andere Sub-Indices wird mit einer Fehlermeldung quittiert.

Life Time Factor

Index 100Dh

Über den Index 100Dh kann der Life Time Factor für das Node-Guarding Protokoll eingestellt werden. Die Multiplikation der Werte auf den Indicies 100Ch und 100Dh ergibt die "Life Time" eines Knotens.

Sub-Index	Datentyp	Zugriff	Bedeutung	Defaultwert
0	Unsigned8	rw	Life Time Factor	0

Das Objekt kann gelesen und geschrieben werden. Es wird nur Sub-Index 0 unterstützt. Ein Zugriff auf andere Sub-Indices wird mit einer Fehlermeldung quittiert.

6.5 Heartbeat Protocol

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

Heart Beat 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 4: Status Information for Heartbeat

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

Example: Power-on of module with address 2

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

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

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:

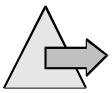
ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
581h	8	60h	17h	10h	00h	00h	00h	00h	00h

Note

The Heartbeat Producer Time is not saved inside the non-volatile memory autonomously. It is necessary to store this parameter via object 1010h (siehe “Store Parameters” auf Seite 23).

6.6 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.



Note

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

6.6.1 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.

6.6.2 Transmit PDO 1

Index 1800h

The object at index 1800h defines communication parameters for the Transmit-PDO.

Sub-Index	Data Type	Acc.	Name	Default Value
0	Unsigned8	ro	Largest Sub-Index	5
1	Unsigned32	rw	COB-ID for PDO	180h + Node
2	Unsigned8	rw	Transmission Type	FFh
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 5: Definition of COB-ID for PDO

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

Transmission Type

The transmission type defines the transmission character of the PDO.

Transmission Type	Description
00h	acyclic synchronous,
01h - F0h (1 - 240 dez)	cyclic synchronous,
FFh (255 dez)	event driven, PDO is sent when Event Timer elapses

Table 6: Setup of Transmission Type

The Transmit-PDO has 2 bytes of process data. The contents is copied from object 7130h into the PDO.

Example: Temperature input at 200,8°C = 07 D8 hex, module address = 1

ID	DLC	B0	B1
181h	1	D8h	07h

Note

Transmission of the PDO is only possible in Operational Mode.

6.6.3 Transmit PDO 2

Index 1801h

Index 1801h has same settings like Index 1800h except for the fact that Transmit PDO 2 contains the so called field values.

The field value consist of 2 bytes data and contains the non-linearized values from the analog-to-digital converter. These data is normally not for use but in case of a linear ratio between input and output value (like 0..20mA) these values can be used for customer specific linearisation on the master module.

Example: Analogue input at 0V DC = 8000 hex, module address = 1

ID	DLC	B0	B1
281h	1	00h	80h

6.6.4 Synchronisation Message

Index 1005h

The object at index 1005h defines the identifier for the SYNC-message. On reception of a message with this identifier the transmission of PDOs is triggered (siehe “Transmit PDO 1” auf Seite 30)..

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

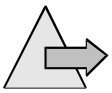
The default identifier is 80h in order to ensure a high priority of the SYNC-message.

Note

The SYNC-identifier is not saved inside the non-volatile memory autonomously. It is necessary to store this parameter via object 1010h (siehe “Store Parameters” auf Seite 23)

6.7 Emergency Message

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



Note

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

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

ID	*DLC*	*B0*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
	8	Error Code		00h	Manufacturer Specific Error Field				

The following emergency error codes are supported:

Error Code (hex)	Description
0000	Error Reset or No Error
1000	Generic Error
50xx	Device-Hardware
6000	Device-Software
80xx	Communication Error / Physical

Table 7: Emergency Error Codes

6.7.1 Device Specific Parameter

The µCAN.sensor contains the following device specific objects. These objects can be found in the DS-404 profile.

Object	Description	Access
6110	Sensor Type	ro
6112	Operating Mode	rw
6131	Process Value Physical Unit	ro
6132	Process Value Dec. Digits	ro
6150	Input Status	ro
6508	Alarm Type	rw
6509	Alarm Action	rw
650D	Alarm State	ro
7100	Field Value	ro
7130	Process Value	ro
7500	Alarm Input Value	ro
750A	Alarm Level	rw
750B	Alarm Hysteresis	rw

7. Technical Specification

7.1 µCAN.sensor

Technichal Specification

Supply Voltage	
Supply Voltage	8...40 V DC, polarity protected
Power consumption	typ. 350mW, max. 600mW
Connection	Solder pads or M12-connector
CAN-Interface	
Supported Bitrates	10kBit/s up to 1MBit/s (according to CiA recommandation)
max. number of nodes	127
Status on the bus	active Node
Protocol	CANopen, DS-404, customer specific protocols on request
CAN specification	2.0A and 2.0B
Connection	Solder pads or M12-connector
Environmental conditions	
Working temp.	-40°C ... +85°C
Storage temp.	-50°C ... +100°C
rel. humidity	15-95%
Diagnosis	
CAN / Hardware	LED constant ON / 50% cycle flash
Error conditions	LED with short flashes
General	
Size (WxLxH)	18 x 30 mm (without M12)
Weight	ca. 4g
Sensor-Input	

Sensortypes	Thermocouple Typ J,K,L (different types on request) Pt100 / Pt1000
Signaltypes	+/- 10 V DC 0-20mA, 4-20mA Strain Gauge, 350 Ohm, 4-wire Pressure Transducers
Connection	Solder pads
Resolution	All signals are converted with 16Bit resolution at sample rates higher 200Hz resolution will decrease to 12Bit
EMC	
EMC	EN 50082-2
Electromagnetic fields	10 V/m, according to ENV 50204
Burst	5 kHz, 2 kV according to EN 6100-4-4
HF unsymmetrical	10 V, according EN 61000-4-6
	Emmision according to EN 50081-2, EN 55022, Class A