

CANoPEN MANUAL

CANopen®

Draw Wire Sensor Series SX
Draw Wire Sensor Series MH
Encoder Series WP

COPYRIGHT

This document is copyrighted by WayCon Positionsmesstechnik GmbH. Changes to, additions to and duplications of this manual are prohibited without the prior written approval of WayCon Positionsmesstechnik GmbH.

SUBJECT TO CHANGE

WayCon Positionsmesstechnik GmbH reserves the right to change this document without notice.

VERSION NUMBER

3.0

DATE

18.01.2019

CONTACT

WayCon Positionsmesstechnik GmbH
Mehlbeerenstraße 4
82024 Taufkirchen
Phone.: +49 89 679713 0
Fax: +49 89 679713 250
Email: sales@waycon.de
Internet: www.waycon.biz

TABLE OF CONTENTS

1. General – Properties	4
2. Data transfer	4
3. NMT - Network Management.....	5
4. PDO - The process data object.....	6
5. SDO Communication.....	7
6. Emergency Message.....	11
7. Offline configuration via Squeezer.....	12
7.1 LED status indicator mode	12
7.2 Configuration mode	13
7.3 Menu items	14
7.4 Factory settings	16
8. Electrical Connection	16

1. GENERAL – PROPERTIES

CANopen is a standardised user interface allowing a simplified system set-up with a variety of devices. CANopen is optimised for fast data exchange in real-time systems and features different standardised equipment profiles. The respective profiles are created and standardised by the users' and manufacturers' group for CAN in Automation (CiA).

WCAN devices support the CANopen communication profile to DS 301 V4.2.0.

CANopen features:

- Easy access to all device parameters
- Network and device configuration
- Synchronising devices on the network
- Cyclic and event-driven process data communication
- Simultaneous data input and output

CANopen uses three communication objects (COB) with different properties:

- Process data objects (PDO) for real-time data
- Service data objects (SDO) for parameter and program transmission
- Network Management (NMT, Life-Guarding, Heartbeat)

All device parameters are saved to an object directory. This object directory contains the description, data type and structure of parameters as well as the address (index). The directory is divided into a communication profile part, a device profile part, and a manufacturer-specific part.

2. DATA TRANSFER

CANopen uses two different types of communication (COB=Communication Object) with different properties to transfer data:

- Process data objects (PDO - real-time capable)
- Service data objects (SDO)

Process data objects (PDO) are used for a highly dynamic exchange of real-time data (e.g. transducer position, speed, status of reference positions) with a maximum length of 8 byte. This data is transmitted with high priority (low COB identifier). PDOs are broadcast messages and supply all designated receivers with real-time data simultaneously.

Service data objects (SDO) form the communication channel for transmitting device parameters (e.g. programming the transducer resolution). Since these parameters are transmitted acyclically (e.g. only once when starting the network), SDO objects are low priority (high COB identifier).

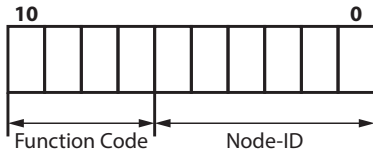
CANopen uses the 'Predefined Master/Slave Connection Set' for easy identifier management. All identifiers are defined by standard values in the object directory. Some identifiers, however, can only be customised through SDO access.

2. DATA TRANSFER

The 11-bit identifier consists of a 4-bit function code and a 7-bit node number.

Bit-No:

COB-Identifier



Note: The higher the value of the COB identifier, the lower priority it is!

3. NMT - NETWORK MANAGEMENT

All NMT commands are transmitted as an unconfirmed NMT object. The broadcast communication model ensures all devices recognise the NMT commands. An NMT object consists of:

COB-ID of the NMT	Byte 0	Byte 1
000h	Command byte	Node ID (module address)

The COB-ID of the NMT object is always 000h. Byte 0 is the command byte.

The following table lists the possible commands:

Command byte (hex)	Description
01h	Start_Remote_Node: Switch to operational
02h	Stop_Remote_Node: Switch to prepared
80h	Enter_Pre-Operational_State: Switch to pre-operational
81h	Reset_Node: Reset node
82h	Reset_Communication: Reset communication

The module addressed by byte 1 will now respond to the command in byte 0. CANopen allows up to 127 devices (1 to 127).

Setting byte 1 to 0 will simultaneously communicate with all devices, i.e. all modules execute the command in byte 0.

There are three modes: pre-operational, operational and stopped. Commands 81h and 82 will reset the module/modules. A reset is the same as restarting the module, i.e. the module will temporarily enter initialisation mode and send a boot-up sequence (701h to 7FFh). WCAN will then enter the mode defined as the starting condition.

3. NMT - NETWORK MANAGEMENT

Not every mode will allow an object to be accessed. The table below shows which allows access:

Object	Operating Modus			
	Initialisation	Pre-Operational	Operational	Stopped
PDO			X	
SDO		X	X	
SYNC		X	X	
Emergency		X	X	X
NMT		X	X	X
Node Guard (Heartbeat)		X	X	X
Boot-Up	X			

4. PDO - THE PROCESS DATA OBJECT

Process data exchange with CANopen in turn is purely CAN bus, so without protocol overhead. The broadcast properties of the CAN bus remain completely intact. A message can therefore be received and analysed by all nodes (Producer/Consumer model). Data exchange with PDOs therefore cancels the rigid master/slave model.

Since the message is lacking protocol structure, it is essential that the users of the bus, for which this data is intended, know how the information is embedded in the data part of the PDO (which bit/byte corresponds to which value). This is therefore declared beforehand during network initialisation through so-called PDO mapping, which allows the desired information to be placed in a specific location in the data area of a PDO.

The WayCon CANopen interface (WCAN) only supports TPDO1 and has a data length of 2 bytes. Depending on the selected node ID (node address) the COB-ID will range from 181h (node 1) and 1FFh (node 127). The TPDO1 range is 16 bit, so can theoretically be any value from zero to 65535. However, our WCAN devices are taught in to the standard start value of 5000 and end value of 55000. This makes it easier for the user to assign the position. Be sure not to define the start value (5000) of a draw-wire sensor to 'impact damper at stop', but with a little pulled out. This therefore allows TPDO1 data values < 5000. The same applies to the final point. Values can be > 55000. Overflow or underflow of the 16-bit value range, however, is impossible.

TPDO1	Data0	Data1
181h...1FFh	Low-Byte	High-Byte

5. SDO COMMUNICATION

Service data objects (SDO) form the communication channel for transmitting device parameters. Since parameter transmission is acyclic (e.g. only once when starting the network), the SDO objects are low priority (high COB identifier).

SDO message structure

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
SDO identifier	Data length	Command	Index L	Index H	Sub-index	Data 0	Data 1	Data 2	Data 3

The SDO identifier is defined as:

- Client (controller) -> Server (absolute-value transducer): 600h + node ID
- Server (absolute-value transducer) -> Client (controller): 580h + node ID

The data length (DLC) is always 8:

1 command byte + 2 index bytes (object) + 1 sub-index byte + 4 data bytes

The command determines whether to write (download) or read (upload) data and how many bytes of user data it contains:

Command	Description	User data	Function
22h	SDO(rx), Download Request	undefined	Send parameters to sensor
23h		4 bytes	
2Bh		2 bytes	
2Fh		1 bytes	
60h	SDO(tx), Download Response	-	Confirm parameter transfer to client
40h	SDO(rx), Upload Request	-	Request parameters from sensor
42h	SDO(rx), Upload Response	undefined	Send parameters to client
43h		4 bytes	
4Bh		2 bytes	
4Fh		1 bytes	
80h	SDO(tx), Abort Domain Transfer	4 bytes	Sensor sends error code to client

If an error occurs, an error message with command 80h (SDO Abort Message) will be sent in place of a normal confirmation (Response). Index and sub-index are part of the above object. Bytes 5 to 8 contain the error code (abort code).

5. SDO COMMUNICATION

The WCAN electronics contain the following objects from the CiA 301 communication profile:

Object	Name	Attribute
1000h	Device Type	ro
1001h	Error Register	ro
1005h	COB-ID SYNC Message	rw
1010h	Store Parameters	rw
1011h	Restore Default Parameters	rw
1014h	Emergency COB-ID Message	ro
1017h	Heartbeat Producer Time	rw
1018h	Identity Object	ro
1029h	Error Behaviour Object	ro
1800h	1st Transmit PDO Parameters	rw
1A00h	1st Transmit PDO Mapping	ro
2100h	Manufacturer specific profile area (baud rate)	rw
2101h	Manufacturer specific profile area (node address)	rw
2102h	Manufacturer specific profile area (termination)	rw
2104h	Manufacturer specific profile area (Autostart)	rw
2105h	Manufacturer specific profile area (save)	rw
6004h	Position Value	ro

Object 1000h: Device Type

This object transmits the device type per CiA. The data part of this object contains the policy and the device profile.

Object 1001h: Error Register

This object states the error per CiA.

Object 1005h: COB-ID SYNC Message

The object defines the COB-ID for the SYNC message. It further defines whether the device produces or receives SYNC objects. WCAN only serves as receiver. The default of the COB-ID is 80h, but can be configured from 001h to 7FFh by the user.

(Note: WCAN will only respond to SYNC commands if the transmission type in object 1800h was configured accordingly.)

Object 1010h: Store parameter

This object saves the current values of the COB-ID SYNC, heartbeat timer, the transmission type and the event timer to the non-volatile WCAN memory.

This object has 1 sub-index. Sub-index 0 shows the number of sub-indices supported (in this case 1). Sub-index 1 shows that only the option 'Store all parameter' is supported. In the event of a write error, data area D0 to D3 in this sub-index must contain the values 73h, 61h, 76h and 65h to allow saving. (These values correspond to the letters **s a v e** in ASCII code.)

5. SDO COMMUNICATION

Object 1011h: Restore Default Parameters

This object will reset the parameters COB-ID SYNC, heartbeat timer, transmission type and the event timer to the factory settings. For heartbeat this is 0h, for COB-ID SYNC it is 80h, for the event timer it is 0h and for the transmission type it is 01h. This object has 1 sub-index. Sub-index 0 shows the number of sub-indices supported (in this case 1).

Sub-index 1 shows that only the option 'Restore all parameters' is supported. In the event of a write error, data area D0 to D3 in this sub-index must contain the values 6Ch, 6Fh, 61h and 64h to allow saving.

(These values correspond to the letters *l o a d* in ASCII code.)

Object 1014h: COB-ID Emergency

The object shows the COB-ID for emergency messages. The COB-ID is determined by the selected node ID and cannot be changed (read only). If an error occurs, Error Behaviour will be written to object 1029h.

Object 1017h: Producer heartbeat time

The producer heartbeat time defines the cycle time of the heartbeat. The range is 2 bytes in the unit ms. The cycle time can be set between 1 ms and 65535 ms. If the entry is null, a heartbeat will not be sent, thus disabling this function.

Object 1018h: Identity object

This object contains manufacturer-specific information. It has 2 sub-indices. Sub-index 0 shows the number of sub-indices supported (in this case 2). Sub-index 1 contains the Vendor-ID (registered with the CiA Organisation). Sub-index 2 contains a product code.

Object 1029h: Error behaviour

This object has 1 sub-index. Sub-index 0 shows the number of sub-indices supported (in this case 1). Sub-index 1 shows that if an error occurs, the WCAN will follow the CiA specification, 01h = No change of the NMT state'.

Object 1800h: TPDO1- communication parameter

The object contains the parameters for the process data object PDO1 and supports 5 sub-indices.

- Sub-index 0 shows the number of sub-indices supported (in this case 5).
- Sub-index 1 is used to read the COB-ID (Data0 and Data1) for TPDO1. (Example: if node ID = 7, TPDO1 will be COB-ID 187h). There is no write access to the COB-ID, i.e. TPDO1 is always node ID + 180h. Data3 in this sub-index contains the value 40h. This means that RTR frames (Remote Transmission Frames) are not allowed for this PDO.
- Sub-index 2 defines the transmission type. Possible values here are 1 to 240 and 255 (decimal). Values 1 to 240 will send a TPDO1 if a sync message is received (see object 1005h COB-ID SYNC Message). With value 1 a TPDO1 will be sent on every sync, with value 2 only every other sync, etc., all the way to 240. With value 255, sync is disabled. TPDO1 process data will then be sent cyclically in operational mode without prompt from a sync message. The description for sub-index 5 explains how the cycle time (repeat rate) is determined.

5. SDO COMMUNICATION

- Sub-index 3 & 4 have no assigned function and are not used.
- Sub-index 5 determines the cycle time of TPDO1 messages if the function 255 was selected in sub-index 2. The range is 2 bytes in the unit ms. The repeat rate can therefore be set to between 1 ms and 65535 ms. If the entry is null, no TPDO1 will be sent. (This status corresponds to disabling TPDO1, as the system also will not respond to sync messages if sub-index 2 is set to 255).

Object 1A00h: TPDO1- mapping parameter

The object contains the mapping parameters for process data object TPDO1 and has 1 sub-index. Sub-index 0 shows the number of sub-indices supported (in this case 1). Sub-index 1 references object 6004h and shows the process data range is 16 bit.

Objects 21xxh: Manufacturer specific profile area

Object 2100h: Baud Rate

This object is used to set the WCAN baud rate. The range is 1 to 7.

The values are assigned the following baud rates:

Value	1	2	3	4	5	6	7
Baud rate	1 Mbps	800 kbps	500 kbps	250 kbps	125 kbps	50 kbps	20 kbps

Changes require restarting or resetting the WCAN. This requires saving with object 2105h (see below).

Object 2101h: Node Number

This object is used to change the node address. The default is 0x07h. The range is 1 to 127. CANopen therefore allows up to 127 devices to be operated with one bus. (The node number 0 is reserved by NMT and must not be used!)

Changes require restarting or resetting the WCAN. This requires saving with object 2105h.

Object 2102h: CANbus termination (enable/disable terminating resistor)

This object is used to enable/disable bus terminating. It controls an internal 120 Ohm terminating resistor between the CAN-High and CAN-Low lines. The range of this object is 0 and 1. If set to 0 the terminating resistor is disabled, if set to 1 the terminating resistor is enabled.

Changes require restarting or resetting the WCAN. This requires saving with object 2105h.

Object 2104h: NMT Autostart

This object defines the WCAN start-up mode when switched on. The range is 0 and 1. If set to 0, WCAN will enter pre-operational mode after being switched on or reset, if the value is 1, WCAN will straight away enter operational mode without command from NMT.

Changes require restarting or resetting the WCAN. This requires saving with object 2105h.

Object 2105h: Store parameter of manufacturer specific profile area

This object saves all current parameters of objects 2100h, 2101h, 2102h and 2104h in WCAN. In the event of a write error, data area D0 to D3 in this sub-index must contain the values 73h, 61h, 76h and 65h to allow saving. (These values correspond to the letters **s a v e** in ASCII code.)

Object 6004h: Position value

This is the sensor's current position value. The range is 16 bit. The LowByte of this 16 bit value is data 0, the HighByte is data 1.

5. SDO COMMUNICATION

Error messages (SDO abort codes)

In the event of a communication error, an error message will be output to the SDO.

The command byte is always 80h.

The table below shows possible error messages:

0504 0001h	Client/server command specifier not valid or unknown.
0601 0002h	Attempt to write a read only object.
0602 0000h	Object does not exist in the object dictionary.
0607 0012h	Data type does not match, length of service parameter too high
0607 0013h	Data type does not match, length of service parameter too low
0609 0011h	Sub-index does not exist.
0609 0030h	Invalid value for parameter
0800 0021h	Data cannot be transferred or stored to the application because of local control.

6. EMERGENCY MESSAGE

WCAN will automatically send emergency messages if an error occurs. Here we need to distinguish between SDO error messages due to an access error to an SDO object and 'true' error messages as an Emergency Message. The first type of error will trigger an error message. An error message with error code 0000h will also be sent if the error has been corrected and the cause of the error no longer exists.

The COB-ID for the emergency message is computed from the node ID (module address) + 80h.

In WCAN the error codes can be:

Byte_1	Byte_0	
00h	00h	No error
81h	20h	CAN in error passive mode
81h	50h	CAN-ID collision

7. OFFLINE CONFIGURATION VIA SQUEEZER

WCAN devices do not support Layer Setting Service (LSS), but do have a convenient function to view and change important parameters such as baud rate, node address, auto-start, termination, etc. without CANbus connection. This function requires an inexpensive accessory called Squeezer.



The following describes this function.

7.1 LED status indicator mode

The Squeezer on principle has two functions. On the one hand it serves as an LED status display for CANopen, on the other hand it allows the user to view or change some CANopen parameters.

The LED status indicator mode starts automatically approx. 3 seconds after powering on the sensor. However, the LED sensor must not be activated during this time.

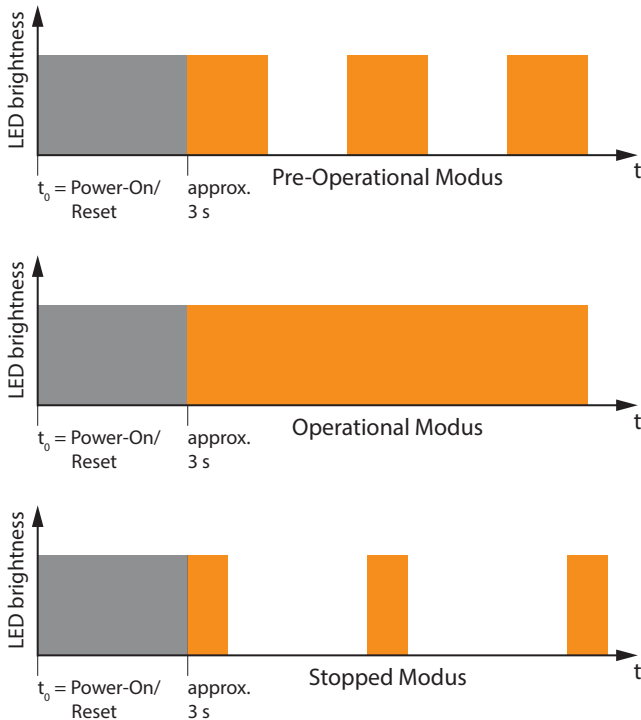


Figure 1

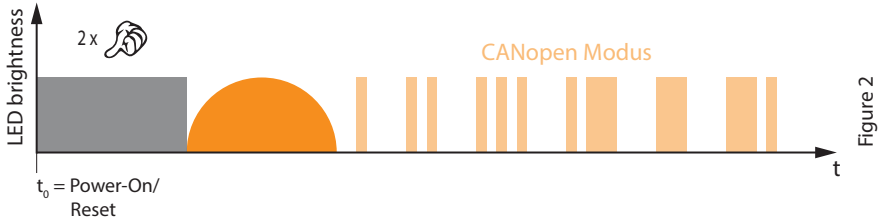
As shown in figure 1, the ON/OFF ratio of the LED in pre-operational is 1:1, in operational mode the LED stays on and in stopped mode the ratio is approx. 1:4.

7. OFFLINE CONFIGURATION VIA SQUEEZER

7.2 Configuration mode

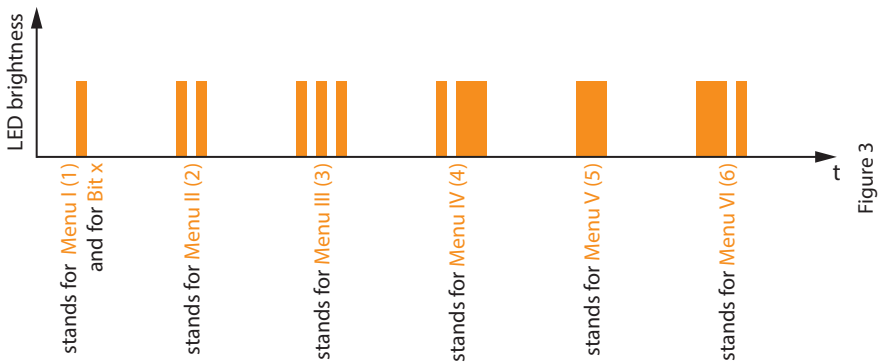
When pressing the LED button on the squeezer within approx. 3 seconds of powering on or resetting the software (per CANopen specifications), the WCAN sensor will enter configuration mode.

This is indicated by the LED light slowly fading on and off once. This **slow on and off** (approx. 2 s) means 'OK' (see Figure 2).



In configuration mode the LED will cycle through 6 different patterns.

The patterns are based on the Roman numerals, i.e. **lighting up for a short time** represents the Roman numeral I and **lighting up for a long time** represents the Roman numeral V (Figure 3).



In configuration mode the blinking patterns for menu items 1 (I) to 6 (VI) will be displayed in succession with an approx. 3 s pause between the blinking patterns. This blinking pattern loop is infinite and can only be exited by rebooting. To select one of these menu items, press the LED button 1 x when the respective blinking pattern appears. This will be acknowledged with 'OK' as shown in figure 2.

In addition to (,OK') and (,Bit x') two other light patterns can appear under a menu item:

1. : LED fading on and off rapid 1x (approx. 1 s) meaning High or 1
2. : LED fading on and off very quickly 3x (approx. 1 s) meaning Low or 0

7. OFFLINE CONFIGURATION VIA SQUEEZER

7.3 Menu items

As described above there are 6 menu items in configuration mode. The menu items contain the following functions:

Menu 1: View node address





Menu 2: Set node address

Menu 3: View baud rate

Menu 4: Set baud rate

Menu 5: View various CANopen parameters (autostart, termination, heartbeat, event timer, transmission type).

Menu 6: Set various CANopen parameters (autostart, termination, heartbeat, event timer, transmission type).

Pressing the button under e.g. menu item 1 (view node address) the system will first show  ('OK'), then  ('Bit x') followed by  or  depending on whether this bit position is 1 or 0. This is followed by 6 other 'Bit x'.

These 7 'Bit x' make a binary number which will be converted to a decimal.

Example: 

This corresponds to binary 1011001 and stands for node address 89.

At the end of the selected menu item 'OK' will again be output. After this it will continue with the display shown in Figure 3.

The procedure 'set' is similar to the 'view' procedure with the difference that after the 'Bit x' the LED will not signal 0 or 1 but instead the button will be monitored for activation. It will be monitored for approx. 3 seconds.







If the LED button is then pressed 1x whilst being monitored, a 1 will be entered at this bit position, if the button is not pressed a 0 will be added.

The last menu entry is also indicated with 'OK' except if the entry is outside the range of the respective menu (e.g. when selecting node address = 0 or baud rate = 0).

In this case it will show for approx. 2 s  („NOK“ - not OK).

A valid input value will already be stored to read-only memory in non-volatile form when the last menu entry 'OK' fades away. However, changes will only be applied on reboot.

Legend:

	lights up briefly, meaning Bit (x)		LED fades on and off 1x (approx. 3 s), meaning OK
....	Pause of < 1 s		Under menu selection: Menu not selected In menu: set to Low resp. 0
	LED quickly fades on and off 3x (approx. 1 s), meaning not set resp. Low or 0		Under menu selection: Menu selected In menu: set to High resp. 1
	LED fades on and off 1x (approx. 1 s), meaning set resp. High or 1		

7. OFFLINE CONFIGURATION VIA SQUEEZER

The table below lists all 6 menu items:

Menu	I (1)	II (2)	III (3)	IV (4)	V (5)	VI (6)
Function	Node address		Baud rate		view (read)	set (write)
	view (read)	set (write)	view (read)	set (write)		
Bit_6	= 0 = 1	= 0 = 1		-	= 0 = 1	= 0 = 1
Bit_5	= 0 = 1	= 0 = 1		-		↓
Bit_4	= 0 = 1	= 0 = 1		-	Autostart 0 = Pre-Operational 1 = Operational	
Bit_3	= 0 = 1	= 0 = 1		-	Terminierung 0 = ∞ Ω 1 = 120 Ω	
Bit_2	= 0 = 1	= 0 = 1	= 0 = 1	= 0 = 1	Heartbeat 0 = 0 ms 0 = 0 ms 1 = > 0 ms 1 = 500 ms	
Bit_1	= 0 = 1	= 0 = 1	= 0 = 1	= 0 = 1	Event-Timer 0 = 0 ms 0 = 0 ms 1 = > 0 ms 1 = 10 ms	
Bit_0	= 0 = 1	= 0 = 1	= 0 = 1	= 0 = 1	Transmission 0 = FFh 0 = FFh 1 ≠ FFh 1 = 01h	
Range	1 to 127		1 = 1 Mb/s 2 = 800 kb/s 3 = 500 kb/s 4 = 250 kb/s 5 = 125 kb/s 6 = 50 kb/s 7 = 20 kb/s		-	

(Please note, when changing a menu item all bits under this menu need to be set.)

All settings from the above table are also available in the CANopen specifications. In this case, changes made through the squeezer and per CANopen specification are linked logically, i.e. changes are mirrored.

7. OFFLINE CONFIGURATION VIA SQUEEZER

7.4 Factory settings

WCAN devices by default have the node ID 7 and a baud rate of 250 kb/s at the time of delivery. WCAN redundant devices have the node IDs 7 and 8. However, other options are available on request.

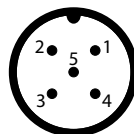
8. ELECTRICAL CONNECTION

Draw wire sensors with WCAN are available with cable output and connector output M12 (male). The pin assignment can be found in the sections below. The cable specifications can be found in the following chart:

	cable, 5 poles	cable, 8 poles
Cable type	TPE, flexible	
Diameter	Ø 6 mm	Ø 6.6 mm
Wire	0.34 mm ²	0.25 mm ²
Temperature	Fixed installation -30...+85 °C, flexible installation -20...+85 °C	

Assignment of simple and redundant CANopen (WCAN)

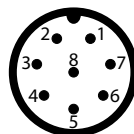
Signal	Pin	Cable colour
CAN_GND	1	BN
+V	2	WH
GND	3	BU
CAN_H	4	BK
CAN_L	5	GY



In case of a redundant CANopen (MH series only) both units share the same bus, with different node IDs (7 and 8).

Assignment of simple and redundant CANopen with offline configuration via Squeezer (WCANP)

Signal	Pin	Cable colour
CAN_GND	1	WH
+V	2	BN
GND	3	GN
CAN_H	4	YE
CAN_L	5	GY
MFL-1 *	6	PK
MFL-2 *	7	BU
n. c.	8	RD



* Multi-functional line for configuration via Squeezer

MFL-1 for the simple CANopen. MFL-2 for the redundant CANopen (MH series only).