



# **TSL- / TSR-CAN-03**

## **Temperature Sensor Probe with CANopen-Protokoll and LSS-Slave-Function (SW-Version 3.0)**

Fluid.iO Sensor + Control GmbH & Co. KG  
Stand: 2025-05-15

*The following documentation is property of Fluid.iO Sensor + Control GmbH & Co KG. Without the permission by Fluid.iO Sensor + Control reproduction, storage in information systems or passing on of this documentation, either completely or in an abridged version, is forbidden.*

<b>1</b>	<b>NETWORK MANAGEMENT</b>	<b>4</b>
<b>2</b>	<b>LIST OF THE SUPPORTED INDICES</b>	<b>6</b>
<b>3</b>	<b>SDO COMMUNICATION</b>	<b>7</b>
3.1	TELEGRAM STRUCTURE SDO	7
3.2	SDO STRUCTURE TO CiA-DS-301	7
3.2.1	Device Profile (Index 1000h)	7
3.2.2	Error Register (Index 1001h)	8
3.2.3	Synchronization Message (Index 1005h)	9
3.2.4	Device Name (Index 1008h)	9
3.2.5	Hardware Version (Index 1009h)	9
3.2.6	Software-Version (Index 100Ah)	10
3.2.7	Node ID (Index 100Bh)	10
3.2.8	Guard Time (Index 100Ch)	10
3.2.9	Lifetime Factor (Index 100Dh)	11
3.2.10	Store Parameters (Index 1010h)	11
3.2.11	Load Default Parameters (Index 1011h)	12
3.2.12	COB ID Emergency Object (Index 1014h)	12
3.2.13	Identity Object (Index 1018h)	13
3.3	DEVICE SPECIFIC SDO STRUCTURE TO CiA-DSP-404	14
3.3.1	Sensor Type (Index 6110h)	14
3.3.2	Physical Unit (Index 6131h)	15
3.3.3	Number of Decimal Position (Index 6132h)	15
3.3.4	Alarm Type (Index 6508h)	16
3.3.5	Alarm Action (Index 6509h)	16
3.3.6	Alarm State (Index 6600h)	17
3.3.7	Request of the AD Value (Index 7100h)	17
3.3.8	Offset-Shift (Index 7124h)	18
3.3.9	Request of the Process Value (Index 7130h)	18
3.3.10	Request of the Alarm Input Value (Index 7500h)	19
3.3.11	Alarm Level (Index 750Ah)	19
3.3.12	Alarm Hysteresis (Index 750Bh)	19
3.4	SDO ERROR MESSAGES	20
<b>4</b>	<b>PDO- COMMUNICATION</b>	<b>21</b>
4.1	SETUP TRANSMIT PDO	21
4.1.1	Index 1800h – Temperature channel	21
4.2	PDO MAPPING	22
4.2.1	Index 1A00h – Temperature channel	22
4.3	START IN OPERATIONAL MODE	22
4.4	PDO TRANSMIT	22
<b>5</b>	<b>EMERGENCY MESSAGES</b>	<b>23</b>
<b>6</b>	<b>NODE GUARDING</b>	<b>24</b>
<b>7</b>	<b>LSS SLAVE FUNCTIONS</b>	<b>24</b>
7.1	SWITCH MODE GLOBAL	25
7.2	SWITCH MODE SELECTIVE	25
7.3	CONFIGURE NODE ID	26
7.4	CONFIGURE BIT TIMING PARAMETERS	26
7.5	ACTIVATE BIT TIMING PARAMETERS	27
7.6	STORE CONFIGURATION PROTOCOL	28

# 1 Network Management

After applying of the supply voltage the CAN sensor transmits the CAN message called "boat up message". This is a CAN telegram with a 1 data byte (contents zero) and with the COB-Identifier 1792 + module-ID. If the module-ID is ID 16, the Identifier 1808 will receive. After this the sensor is in the Pre-Operational-Mode.

The following Objects are supported by the CAN sensor:

<i><b>Object</b></i>	<i><b>COB-ID (decimal)</b></i>	<i><b>Modul-ID</b></i>	<i><b>Remark</b></i>
Network Management	0	0	Start, Stop, Reset
SYNC	128	0	Transmit PDO
EMERGENCY	129 – 255	1-127	From the Sensor
PDO	385 – 511	1-127	From the Sensor
SDO	1409 – 1535	1-127	From the Sensor
SDO	1537 – 1663	1-127	To the Sensor
Node-Guarding	1793 – 1919	1-127	Bidirectional

## Network Management (COB-ID = 0)

Start Node

<b>ID</b>	<b>DLC</b>	<b>Byte 1</b>	<b>Byte 2</b>
<b>0</b>	<b>2</b>	<b>01h</b>	<b>Node</b>

Node = Module-ID, 0 = all Modules

The order "start Node" puts the CAN sensor into the Operational mode, i.e. then it can also send PDO (sending of measure information).

Stop Node

<b>ID</b>	<b>DLC</b>	<b>Byte 1</b>	<b>Byte 2</b>
<b>0</b>	<b>2</b>	<b>02h</b>	<b>Node</b>

Node = Module-ID, 0 = all Modules

This order stop sets the CAN sensor into the Stopped Mode. Only NMT commands can be received and Node-Guarding is carried out here.

Enter Pre-Operational Mode

<b>ID</b>	<b>DLC</b>	<b>Byte 1</b>	<b>Byte 2</b>
<b>0</b>	<b>2</b>	<b>80h</b>	<b>Node</b>

Node = Module-ID, 0 = all Modules

This order sets the CAN sensor into the Pre-Operational mode. The sensor is fully viable, merely PDOs cannot be sent.

**Reset Node**

<b>ID</b>	<b>DLC</b>	<b>Byte 1</b>	<b>Byte 2</b>
<b>0</b>	<b>2</b>	<b>81h</b>	<b>Node</b>

Node = Module-ID, 0 = all Modules

With the order "Reset Node" a Reset of the CAN sensor is executed. After the Reset the node is in the Pre-Operational mode and transmits the "boat up message" (see above).

**Reset Communication**

<b>ID</b>	<b>DLC</b>	<b>Byte 1</b>	<b>Byte 2</b>
<b>0</b>	<b>2</b>	<b>82h</b>	<b>Node</b>

Node = Module-ID, 0 = all Modules

With the order "Reset Communication" merely a Reset of the CAN controller is executed. After it the node is in the Pre-Operational mode and transmits the "boat up message" (see above).

## 2 List of the supported indices

The following indices are supported by the CAN sensor:

<i><b>Index</b></i>	<i><b>Name</b></i>
1000h	Device Profile
1001h	Error Register
1005h	COB-ID SYNC-Message
1008h	Device Name
1009h	Hardware Version
100Ah	Software Version
100Bh	Node-ID
100Ch	Guard-Time
100Dh	Lifetime-Factor
1010h	Store Parameter
1011h	Load Default Parameter
1014h	COB-ID Emergency Object
1018h	Identity Object
1800h	Setup for PDO-Transmit Temperature
1A00h	Request PDO Mapping Temperature
6110h	Sensor Type
6131h	Physical unit
6132h	Number of decimal position
6508h	Alarm Type
6509h	Alarm Action
6600h	Alarm State
7100h	Request of the AD-Value
7124h	Offset-Shift
7130h	Request of the process value
7500h	Request of the alarm input value
750Ah	Alarm level
750Bh	Alarm hysteresis

### 3 SDO Communication

#### 3.1 Telegram Structure SDO

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	8	CMD	Index		Sub-Index	Data bytes			

The Command byte (CMD) has the following function:

Read access of the server to CAN sensor:	40 h
CAN sensor answers:	42 h
Write access of the server to CAN sensor:	22 h
CAN sensor answers:	60 h

The LSB is transferred first at index and data bytes!

The area of the communication profile is in the indices of 1000 h – 1FFF h and contains all parameters which concern the CAN network. This area is defined in all CANopen devices.

#### 3.2 SDO Structure to CiA-DS-301

<i>Index</i>	<i>Name</i>
1000h	Device Profile
1001h	Error Register
1005h	COB-ID SYNC-Message
1008h	Device Name
1009h	Hardware Version
100Ah	Software Version
100Bh	Node-ID
100Ch	Guard-Time
100Dh	Lifetime-Factor
1010h	Store Parameter
1011h	Load Default Parameter
1014h	COB-ID Emergency Object
1018h	Identity Object

##### 3.2.1 Device Profile (Index 1000h)

**Example:** Read access of the server, Module-ID = 16d

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	40h	00	10h	00	00	00	00	00

For calculation of ID value please refer "Network Management"

The CAN-Sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	00	10h	00	94h	01h	22h	00h

The data bytes are fixed in accordance with CANopen standard DSP 404:

Byte5 + byte6 = 0194h = 404d

(Device profile Number: Measurement Devices)

Byte7 + byte8 = 0022h = 22h

(Additional information: The device contains an analog input and an alarm function)

The index 1000h has the status "read-only access". Sub indices aren't supported. Every faulty access is answered by sending a corresponding SDO error message. See "SDO error messages".

### 3.2.2 Error Register (Index 1001h)

The index 1001h is used to read out the Error register.

**Example:** Read access of the server, module ID = 16

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	40h	01h	10h	00	00	00	00	00

In the answer the Error register is transferred in bytes 5. The following errors are displayed in this version:

#### Generic Error

Bits 0 in the byte 5 is set.

The Generic Error is triggered by a fault at the ADC, i.e. if the result is small 5 or equal 1023. In this case there is a hardware defect.

#### EEPROM-Error

Bit 1 in the byte 5 is set.

This error occurs, if writing to the EEPROM of the CAN sensor is not possible (hardware defect).

#### Communication Error

Bit 4 in the byte 5 is set.

This error occurs at disturbances in the communication on the CAN bus, i.e. if one of the error counters of the CAN controller has exceeded the value of 95.

This bit is set too, if Node-Guarding is switched on and the NMT-Master doesn't poll the CAN sensor within the Node Life Time.

#### Alarm Error

Bit 5 in the byte 5 is set.

This error is active, if the CAN sensor is in the alert, i.e. at least one of the booth process values has exceeded the values of the alarm limits (look also SDO 6508h, 950Ah und 950Bh).

The index 1001h has the status "read-only access", Sub indices aren't supported. Every faulty access is answered by sending a corresponding SDO error message. See "SDO error messages".



### 3.2.3 Synchronization Message (Index 1005h)

Configuration and request of the configuration of the SYNC message. By means of the SYNC message a transmit can be triggered by PDOs (see PDO "communication"). The index is built up as follows:

<b>Index</b>	<b>Parameter</b>	<b>Access</b>
1005h	ID (32 bit)	Read/write

The 32 bit parameter area is built up as follows:

<b>Bit 31</b>	<b>Bit 30-11</b>	<b>Bit 10-0</b>
1	0	ID 11 Bit

The default ID is 80h. This ensures the SYNC messages a high priority on the CAN bus. The bit 31 shows, whether the device generates SYNC message themselves or not. If the bit 31 is set, the device processes SYNC messages. The CAN sensor can the SYNC message only processes, that's why bit 31 is always set, independently of what of the servers sends.

### 3.2.4 Device Name (Index 1008h)

By means of the index 1008h the device name of the manufacturer can be questioned. This is manufacturer specific and gets transferred as an ASCII text.

**Example:** Read access of the server, module ID = 16

<b>ID</b>	<b>DLC</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>	<b>Byte 8</b>
1552	8	40h	08h	10h	00	00	00	00	00

The CAN sensor answers:

<b>ID</b>	<b>DLC</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>	<b>Byte 8</b>
1424	8	42h	08h	10h	00	5Ah	54h	53h	00h

Byte 5 = 5Ah is ASCII = Z (ZILA)

Byte 6 = 44h is ASCII = T (Temperature)

Byte 7 = 53h is ASCII = S (Sensor)

The index 1008h has the status "read-only access", Sub indices aren't supported. Every faulty access is answered by sending a corresponding SDO error message. See "SDO error messages".

### 3.2.5 Hardware Version (Index 1009h)

By means of the index 1009h the hardware version can be questioned.

**Example:** Read access of the server, module ID = 16

<b>ID</b>	<b>DLC</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>	<b>Byte 8</b>
1552	8	40h	09h	10h	00	00	00	00	00

The CAN-Sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	09h	10h	00	33h	2Eh	30h	00h

So this corresponds to the version 3.0

The index 1009h has the status "read-only access", Sub indices aren't supported. Every faulty access is answered by sending a corresponding SDO error message. See "SDO error messages".

### 3.2.6 Software-Version (Index 100Ah)

By means of the index of 100Ah the software version can be questioned.

**Example:** Read access of the server, module ID = 16

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	40h	0Ah	10h	00	00	00	00	00

As an answer the CAN sensor sends the version number in ASCII coded form (see "hardware version").

The index of 100Ah has the status "read-only access", Sub indices aren't supported. Every faulty access is answered by sending a corresponding SDO error message. See "SDO error messages".

### 3.2.7 Node ID (Index 100Bh)

By means of the index of 100Bh the module ID can be questioned.

**Example:** Read access of the server, module ID = 16 for

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	40h	0Bh	10h	00	00	00	00	00

The CAN-Sensor transmits in Byte 5 the module ID:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	0Bh	10h	00	10h	00	00	00

The index of 100Bh has the status "read-only access", Sub indices aren't supported. Every faulty access is answered by sending a corresponding SDO error message. See "SDO error messages". The module Identifier can be edited only with the LSS (Layer Setting Service).

### 3.2.8 Guard Time (Index 100Ch)

By means of the index of 100Ch the Guard time can be questioned or be written as an 16 Bit Integer Value (Read-write-access). The Guard time is a parameter of the Node-Guarding (see there). Is the Guard time zero (default value), the Node-Guarding function is turned off.

**Example:** Read access of the server, module ID = 16 for it

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	40h	0Ch	10h	00	00	00	00	00

The answer is the Guard time in Byte 5 und 6 (in this example 500 ms):

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	0Ch	10h	00	F4h	01h	00	00

The status doesn't support any Sub indices. Every faulty access is answered by sending a corresponding SDO error message. See "SDO error messages".

### 3.2.9 Lifetime Factor (Index 100Dh)

By means of the index of 100Dh the Lifetime-Factor can be questioned or typed in as an 8 Bit integer value (Read-write-access). The Lifetime-Factor is a parameter of the Node-Guarding (see there). Lifetime factor multiplied with the Guard time is the period of time, within the NMT-Master must send at least a Remote-Request telegram to the CAN sensor. A Life Guarding Event otherwise is occurred (an Emergency message is transmitted). Is the Lifetime-Factor zero (default value), the Node-Guarding function is turned off.

**Example:** Read access of the server, module ID = 16 for it

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	40h	0Dh	10h	00	00	00	00	00

The CAN-Sensor answers with the Lifetime-Factor in Byte 5 (in this example 3):

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	0Ch	10h	00	03h	00	00	00

The status doesn't support any Sub indices. Every faulty access is answered by sending a corresponding SDO error message. See "SDO error messages".

### 3.2.10 Store Parameters (Index 1010h)

The parameters received by the CAN sensor are stored only briefly in the RAM, i.e. they are lost, if the operating voltage is turned off. That's why a storage of the parameters must be carried out in a nonvolatile store (EEPROM) after every parameter change or after the change of several parameters with the index 1010h.

Storage is triggered by sending the index 1010h with the message "save" (in ASCII) on the sub index 1 for the temperature parameters. The storage of the identifier of the synchronization message and of the node guarding parameters will be executed. The message for the storage of the temperature parameters has therefore the following construction:

**Example:** Write access of the server, module ID = 16

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	22h	10h	10h	01h	73h	61h	76h	65h

As an answer the CAN sensor sends after completed storage:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	60h	10h	10h	01h	73h	61h	76h	65h

### 3.2.11 Load Default Parameters (Index 1011h)

With help of the index 1011h the default parameter set of the device can be loaded.  
It contains the following parameters:

- 1 digit decimal place, e.g. full scale is 80.0 °C
- SYNC-ID is 80h
- PDO-ID is 180h + module ID
- Periodical transmit of the PDOs with a period of 1 sec
- Alarm type: Alarm "in window"
- Alarm action: Transmit PDO at admission and left the alarm window, send the Emergency message
- Alarm level is 50.0 °C
- Alarm hysteresis is 10.0 K

Loading the default parameters is triggered by sending the index 1011h with the message "load" (in ASCII) on the sub index 1. Therefore the message has the following construction:

**Example:** Write access of the server, module ID = 16 for it

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	22h	11h	10h	01h	6Ch	6Fh	61h	64h

The CAN-Sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	60h	11h	10h	01h	6Ch	6Fh	61h	64h

Loading the default parameter is carried out into the RAM of the device. If these parameters shall be stored nonvolatily, then the SDO 1010h has then to be sent (see above).

### 3.2.12 COB ID Emergency Object (Index 1014h)

Request of the COB ID of the Emergency telegram. Only read status. The COB-ID is 128 plus module ID. The index is built up as follows:

<i>Index</i>	<i>Parameter</i>	<i>Access</i>
1014h	ID (32 bit)	read

The 32-bit parameter range is built up as follow:

<i>Bit 31</i>	<i>Bit 30-11</i>	<i>Bit 10-0</i>
0	0	ID 11 Bit

### 3.2.13. Identity Object (Index 1018h)

The identity object contains basic information about the CANopen temperature sensor. By means of these details a single temperature sensor can be identified beyond all doubt in a net. Altogether 5 sub indices are supported, which have all of them only reads status:

Index	Sub-Index	Parameter	Access
1018h	0	Number of supported Sub indices	Read
	1	Vendor-ID	Read
	2	Product-Code	Read
	3	Main- and under-revision number	Read
	4	Serial number	Read

The Vendor ID is a 4 bytes long manufacturer number which the CiA assigns to the manufacturer of CANopen equipment. The Vendor ID for Fluid.iO Sensor + Control devices is:

00h 00h 00h 0C0h

**Example:** Read access of the server on Sub index 1, Module ID = 16

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	40h	18h	10h	01h	00	00	00	00

The CAN-Sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	18h	10h	01h	0C0h	00h	00h	00h

#### Product-Code

The product code is also 4 bytes long and is allocated by the manufacturer for the respective device class. The product code for Fluid.iO temperature sensors is 00h 00h 00h 02h

In response to a read access on Sub index 2 the CAN sensor sends:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	18h	10h	02h	02h	00h	00h	00h

#### Main- and Under revision number

The revision numbers correspond essentially to the software version number in which you subdivide still finer here: The Main revision number (byte 7 and 8) changes only with an increasing functionality i.e. functions join if new. The Under revision number changes if a fault in the software was removed at the same functionality e.g..

In response to a read access on Sub index 3 the CAN sensor sends:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	18h	10h	03h	02h	00h	01h	00h

#### Serial number

The serial number is also 4 bytes long and is allocated to every device at the production.  
In response to a read access on Sub index 4 the CAN sensor with the serial number "1" transmits:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	18h	10h	04h	01h	00h	00h	00h

### 3.3 Device Specific SDO Structure to CiA-DSP-404

The equipment specific SDO structure contains the following indices:

Index	Name
6110h	Sensor Type
6131h	Physical unit
6132h	Number of decimal position
6508h	Alarm Type
6509h	Alarm Action
6600h	Alarm State
7100h	Request of the AD-Value
7124h	Offset-Shift
7130h	Request of the process value
7500h	Request of the alarm input value
750Ah	Alarm level
750Bh	Alarm hysteresis

#### 3.3.1 Sensor Type (Index 6110h)

The request of the sensor type is carried out via the index 6110h. The index is built up as follows:

Index	Sub-Index	Parameter	Access
6110h	0	Number of supported sub indices	Read
	1	Temperature Sensor type	Read

**Example:** Read access of the server, Module ID = 16, Temperature channel

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	40h	10h	61h	01	00	00	00	00

The CAN-Sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	10h	61h	01	64h	00	00	00

Sensor-Typ = Byte 5 = 64h = 100d - Temperature Transducer.

### 3.3.2 Physical Unit (Index 6131h)

The index 6131h is an index with read-only access. He gives the physical unit. The index is built up as follows:

Index	Sub-Index	Parameter	Access
6131h	0	Number of supported sub indices	Read
	1	Physical unit for temperature	Read

**Example:** Read access of the server, Module ID = 16d, temperature channel

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	40h	31h	61h	01	00	00	00	00

The CAN-Sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	31h	61h	01	2Dh	00	00	00

The measurement unit is coded to CiA DRP 303-2 V1.0 and represented in byte 5:

2Dh = °C

The decade of the measurement unit e.g. milli-, mega- etc. contains byte 6

00h = 10<sup>0</sup>

### 3.3.3 Number of Decimal Position (Index 6132h)

The index 6132h indicates the number of decimal positions of the measure value. The index is built up as follows:

Index	Sub-Index	Parameter	Access
6132h	0	Number of supported sub indices	Read
	1	Number of decimal positions temperature	Read

The output of the measure values „temperature“ will be ever with one digit after the decimal point.

Read access of the server, Module ID = 16d, temperature channel (Modul-ID = 16d):

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	40h	32h	61h	01	00	00	00	00

The CAN-Sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	32h	61h	01	01h	00	00	00

### 3.3.4 Alarm Type (Index 6508h)

The index 6508h indicates the alarm type. The index is built up as follows:

Index	Sub-Index	Parameter	Access
6508h	0	Number of supported sub indices	Read
	1	Alarm-Type temperature channel	Read/Write

Alarm-Type	Alarm, if measure value...
0	Alarm off
2	Greater than or equal the level
3	Under the Level
6	In window (Level, Level + hysteresis)
7	Outside the window

**Example:** It is desired that the sensor alerts if the temperature measure value falls below the alarm threshold (alarm type = 3.)

Write access of the server, module ID = 16:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	22h	08h	65h	01	03h	00	00	00

The CAN-Sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	60h	08h	65h	01	03h	00	00	00

If the sent alarm type should be incorrect, the changes aren't taken on and the previous value comes back in the byte 5.

### 3.3.5 Alarm Action (Index 6509h)

The SDO 6509h fixes what for a reaction shall be carried out in the alarm case. The index is built up as follows:

Index	Sub-Index	Parameter	Access
6509h	0	Number of supported sub indices	Read
	1	Alarm Action number temperature channel	Read/Write

Alarm Action number	Meaning
Bit 0 = 1	Emergency Message is transmitted at the beginning and the end of alarm
Bit 1 = 1	Reserved
Bit 2 = 1	PDO is transmitted at the beginning of alarm
Bit 3 = 1	PDO is transmitted at the end of alarm



**Example:** It is desired, that the temperature channel of the sensor alerts with an "Emergency message" and sends a PDO at alarm end (alarm action number = Bit 0 and Bit 3 = 09h).  
Write access of the server, module ID = 16

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1552	8	22h	09h	65h	01	09h	00	00	00

The CAN-Sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	60h	09h	65h	01	09h	00	00	00

Should the sent alarm action number be incorrect, so the changes aren't taken on and the previous value comes back in the byte 5.

### 3.3.6 Alarm State (Index 6600h)

The SDO 6600h is an index with read-only access. With its help you can read the alarm status. The index is built up as follows:

Index	Sub-Index	Parameter	Access
6600h	0	Number of supported sub indices	Read
	1	Alarm Status temperature channel	Read

The Alarm Status (Byte 5) has got the following structure:

Alarm Status	Meaning
0	No Alarm
1	Alarm is active

By read access to Index 6600h the CAN sensor transmits the following answer:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	00h	66h	01	01	00	00	00

...if the CAN-Sensor is in alarm state, specified by SDO 6508h.

### 3.3.7 Request of the AD Value (Index 7100h)

The index 7100h is an index with read-only access. It gives you the pure AD values, delivered by the AD converters in the sensor. This value is not scaled. The index has the following construction:

Index	Sub-Index	Parameter	Access
7100h	0	Number of supported sub indices	Read
	1	AD value temperature	Read

At a read access on the index 7100h, you get the following answer:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	00h	71h	01	ADLo	ADHigh	00	00

Bytes 5 + 6 distributes the temperature measure value of the AD converter in high byte and Low byte. A request of the AD value is possible too with PDOs (see "PDO communication").

**Remark:** Although the AD value is only 16 bits wide, an index which is intended for 32 bit values was used (7xxxh), so that the index area is uniform (32 bits widely) for pure AD values and scaled measure values.

### 3.3.8 Offset-Shift (Index 7124h)

The index 7124h is an index with read and write access. This object can be used to do a zero adjustment by shifting the offset of the measurement value of the CAN sensor. The index has the following construction:

Index	Sub-Index	Parameter	Access
7124h	0	Number of supported sub indices	Read
	1	Offset value temperature	Read/Write

At a read access on the index 7124h, Sub-Index 1, you get the following answer:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	24h	71h	01	Byte1h	Byte2h	00	00

Byte 5 + 6 represent den offset value in a 16Bit-Integer according to two's complement. Byte1h is LSB and Byte2h is MSB.

**Remark:** The offset value needs to be multiplied by 10.

**Example:** For a positive shift of 0.6 K of the temperature value, the offset value has to be +6 (Byte1h = 06h, Byte2h = 00h).

### 3.3.9 Request of the Process Value (Index 7130h)

The index 7130h is an index with read-only access. It distributes the physical process values, temperature value. The index has the following construction:

Index	Sub-Index	Parameter	Access
7130h	0	Number of supported sub indices	Read
	1	Process value temperature	Read

At a read access on the index 7130h, you get the following answer:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	30h	71h	01	Byte1h	Byte2h	00	00

Byte 5 + 6 represent the process value "temperature" as a 16Bit integer value. Byte1h is LSB and Byte2h is MSB. The number of decimal positions is defined by the object 6132h and is always one.

**Example:** A value of 350d is read, if the temperature is 35.0 °C: Byte1h = 5Eh, Byte2h = 01h, Byte3h and Byte4h are zero. Negative temperature values shown in 32 bit two's complement representation.

### 3.3.10 Request of the Alarm Input Value (Index 7500h)

The index 7500h is an index with read-only access. By means of this object the process values temperature can be read, that is the input value of the alarm routine. The index has the following construction:

Index	Sub-Index	Parameter	Access
7500h	0	Number of supported sub indices	Read
	1	Process value temperature	Read

If there is a read access to Object 7500h, temperature channel, the CAN sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	00h	75h	01	Byte1h	Byte2h	00	00

The result is identical with the answer to the index 7130h.

### 3.3.11 Alarm Level (Index 750Ah)

The index 750Ah has got read/write access. By means of this object the alarm value is typed in (as temperature value). The index has the following construction:

Index	Sub-Index	Parameter	Access
750Ah	0	Number of supported sub indices	Read
	1	Alarm level temperature	Read/Write

If there is a read access to index 750Ah, temperature channel, the CAN sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	0Ah	75h	01	Byte1h	Byte2h	00	00

Byte 5 + 6 represent the registered value "temperature" as a 16Bit integer value. Byte1h is LSB and Byte2h is MSB. The number of decimal positions is defined by the object 6132h, i.e. always one. A value of -100d is read, if the temperature is -10 °C: Byte1h = 9Ch, Byte2h = FFh, (Remark: negative temperature values are in 16Bit two's complement representation)

### 3.3.12 Alarm Hysteresis (Index 750Bh)

The index of 750Bh has got read/write access. By means of this object the Hysteresis values is typed in (as temperature value), i.e. the distance between the upper alarm threshold and the lower

alarm threshold. Of course this value has meaning only then if the alarm type is "7", i.e. there is an alarm window. The index has the following construction:

Index	Sub-Index	Parameter	Access
750Bh	0	Number of supported sub indices	Read
	1	Hysteresis value temperature	Read/Write

If there is a read access to index 750Bh, temperature channel, the CAN sensor answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	0Bh	75h	01	Byte1h	Byte2h	00	00

Byte 5 + 6 represent the registered value "temperature hysteresis" as a 16Bit integer value. Byte1h is LSB and Byte2h is MSB. The number of decimal positions is defined by the object 6132h, i.e. always one.

**Example:** A value of 200d is read, if the temperature hysteresis is 20 K: Byte1h = C8h, Byte2h = 00h.

### 3.4 SDO Error Messages

At faulty access to SD objects you will get an error message as answer. An error message has always the following construction:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	8	80h	Index		Sub-Index	Additional Code		Error Code	Error Class

The ID of the message and the index and sub index always refer for the ID, on which the faulty access has taken place.

The error messages can show the following contents:

Additional Code	Error Code	Error Class	Meaning
01h 00h	04h	05h	Server-Command incorrect or unknown (neither write nor read)
00h 00h	01h	06h	Read access to an only write object
01h 00h	01h	06h	Write access to an only read object
11h 00h	09h	06h	Sub-Index doesn't exist
00h 00h	02h	06h	Object doesn't exist
00h 00h	00h	08h	Other errors (wrong code word)

## 4 PDO- Communication

By means of PDO (Process data Objects) communication it is possible to request certain values of the CAN sensor in simple and fast way. A firm PDO-Mapping is realized in the CAN sensor, i.e. the construction of the PDOs has been fixed.

The request of the PDOs is carried out via a synchronization message. A transmitting automatically with free adjustable time period also can be switched on.

Sending the PDOs is possible for the CAN sensor only in the Operational mode.

### 4.1 Setup Transmit PDO

#### 4.1.1 Index 1800h – Temperature channel

By means of the index 1800h the attitudes are carried out to be able to work with the transmit PDO, temperature channel.

A firm PDO-Mapping is realized at the CAN sensor, i.e. a PDO contains the scaled temperature value, the AD value and the alarm status.

The index has the following construction:

Index	Sub-Index	Parameter	Access
1800h	0	Number of supported sub indices	Read
	1	COB-ID PDO-Message	Read
	2	Type of transmit	Read/Write
	3	reserved	
	4	reserved	
	5	Timer (16 Bit)	Read/Write

Sub index 1 only can be read and contains the Identifier of the PDO message. It is in the temperature channel fixed 180h plus module ID.

By means of sub index 2 the transmit type can be adjusted. The CAN sensor supports the following types:

Transmit type	Meaning
1 – 240d	cyclically synchronous. The CAN sensor reacts to every n-th SYNC message (n = 1...240)
254 d	The CAN sensor sends independently a PDO after x ms. The value for x is typed in under Sub index 5.

If the transmit type is 254 and the value under sub index 05 is greater than 4ms, the CAN sensor sends independently the PDO, so as soon as it was put into the Operational mode.

A millisecond timer is loaded over the sub index 05, over which the PDO is sent automatically. Only cycle times greater or equal 5ms will be supported by the CAN sensor.

## 4.2 PDO Mapping

### 4.2.1 Index 1A00h – Temperature channel

By means of the index 1A00h the PDO-Mapping parameters for the temperature channel can be read out. The SDO index, sub index and the variable types for all data, which is written down on a PDO, are under this index.

A firm PDO-Mapping is realized at the CAN sensor, so that the index 1A00h has read-only access. The index has the following construction:

Index	Sub-Index	Parameter	Access
1A00h	0	Number of supported sub indices	Read
	1	„Map“-information for the scaled temperature values	Read
	2	„Map“-information for the AD values	Read
	3	„Map“-information for the alarm state	Read
	4	„Map“-information for the error register	Read

#### Example:

If there is a read access to Index 1A00h and sub index 1, you get the following answer:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	42h	00h	1Ah	01	10h	01h	30h	71h

Byte 5 gives the length of the data, which the SDO object provides. (10h = 16d = 16 Bit)

Byte 6 shows the Sub index of the SDO object, which provides the same data as the PDO. Bytes 7 and 8 give the index of the SDO object, which provides the same data as the PDO.

## 4.3 Start in operational mode

Normally, the sensor starts with a boot-up message and then enters pre-operational mode. To send PDOs, the sensor must be put into operational mode with a "Start Node" command. To enter operational mode immediately, use the command below. These settings are immediately written to the EEPROM upon execution. However, this command does not change the current state.

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1424	8	22h	00h	18h	03h	01h	00h	6Fh	6Bh
1424	8	22h	00h	18h	03h	00h	00h	6Fh	6Bh

## 4.4 PDO Transmit

The PDO transmit of the CAN temperature sensor (Modul-ID = 10h) is shown in the following table:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
01A0h	8	F2h	00h	00h	00h	F8h	02h	00h	00h

**Example:**

Byte 1 (LSB) + Byte 2 (MSB) represent the temperature measurement value: 00F2h=242d.

Because SDO object 6132h (number decimal places) has a value of „1“ , the temperature is 24.2°C.

Byte 3 and Byte 4 are always zero.

Byte 5 + Byte 6 is the AD value (02F8h = 760d), used for the temperature calculation.

Byte 7 shows the „alarm status“ (SDO object 6600h).

Byte 8 contains the “error register” (SDO object 1001h).

## 5 Emergency Messages

Emergency messages will be sent in the fault case of the CAN sensor independently. It has to be respected the difference between SDO error messages at a faulty access to a SDO object and the "real" error messages as an Emergency message.

Emergency message will be sent automatically **only one time** at the appearance of the faults mentioned below. The transmit isn't repeated, even if the fault is on for some time. The transmit is carried out compulsory, merely the emergency message triggered by alarm you can switch off by means of SDO 6509h (Bit 0 must be reset).

If a fault disappears, then a message is also sent with an error code 00h 00h. The same happens if all faults are removed.

The Identifier of the Emergency message is 128 + module ID.

An Emergency message has the following construction:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
144	8	Error Code		Error register	Manufacturer Specific Error Field				

The following error codes are supported:

Error Code	Meaning	Bits in error register
00h 00h	Fault removed	
11h 00h	Generic Error (ADC)	Bit 0 is set
50h 00h	EEPROM-Error	Bit 1 is set
81h 00h	Communication Error	Bit 4 is set
81h 30h	Life Guarding Event	Bit 4 is set
FFh 00h	Alarm Error	Bit 5 is set

The detailed description of the faults you can find under SDO 1001h (error register).

Only the faults are shown in the error code, that joined currently or disappeared (Error code 00h 00h). The faults still being on will be displayed in error register (byte 3). See also index 1001h.

The Manufacturer Specific Error Field is, at the moment, unused, i.e. byte4 to byte8 are still zero.

## 6 Node Guarding

Node guarding is the cyclical node supervision by a NMT-Master. This is important at nodes which don't send PDOs periodically. If these nodes have got errors, then this is recognized not or too late. This will endanger the complete system security, if there is no node guarding.

Node guarding means: The NMT master is polling every node periodical with a node specific Remote transmission Request telegram (ID is 1792 plus module ID). The node answers with its communication status (Pre-Operational, Operational, Stopped mode). If this answer isn't carried out within a particular time or the communication status doesn't agree with it at the NMT-Master stored, so the Master executes a Node-Guarding Event.

Reversed the node "waits" for the telegrams of the Master, after he has received one for the first time. The distance of the Remote transmission Request telegrams must be less than the Node Life time of the CAN sensor. The Node Life time is calculated by multiplication of the Guard time (SDO 100Ch) with the Life-time Factor (SDO 100Dh). Is this Node Life time exceeded, a Life Guarding Event is triggered in the node. This concretely means at the CAN sensor, that an Emergency message is sent with the error code 81h 30h and the Bit 4 is set in the error register, until a telegram of the NMT-Master has come in again.

Node Guarding can be deactivated by writing zero in SDO 100Ch or in SDO 100Dh.

## 7 LSS Slave Functions

Among other things the Layer Setting Service (LSS) serves to change the module ID and the bit timing parameters (baud rate) at a CANopen device over the CAN net. A Master-Slave structure is used: A LSS master exists in the CAN net and a LSS slave software exists in the CANopen devices. Certain Identifier are reserved: The LSS-Master uses the COB ID 2021, the LSS slaves answer with COB ID 2020.

The following services exist:

### **Switch mode**

LSS-Master gives switch-over command, by which the LSS slave switches from operation mode (normal mode) into the configuration mode.

#### **Switch mode global:**

Change of all CANopen devices of the net into the Configuration mode without a further examination.

#### **Switch fashion selective:**

Only one node in the network switches into the Configuration mode. The Master needs the LSS address of the Slaves which corresponds with the Identify-Object 1018h.

### **Configure Node ID**

Brief change of the node number of the CANopen device. The node number (node ID) must be between 1 and 127 and only one node may be in the Configuration mode.

### **Configure bit timing parameter**

Brief change of the baud rate, which is provided in a baud rate table. Only one node should be in the Configuration mode. The change is only prepared, not executed.

### **Activate bit timing parameter**



With this command the nodes, which are in the Configuration mode, asked to activate the changed baud rate. This must happen for all knots of a network simultaneously. The master gives a delay time in its telegram, which gets twice effective: Slave receives command, waits for the delay time, activates the new baud rate and waits again for the delay time, before he may send. The Delay time you must choose so, that the slowest node can carry out the baud rates switchover in the given time. It shall be prevented, that nodes are in the net with different baud rates at a particular time.

### **Store Configuration**

Nonvolatile storage of the Node ID and baud rate in the EEPROM of the node, so that these can get effective also after a restart.

## **7.1 Switch Mode Global**

### **Switch for all Slaves in Configuration mode or back in operation mode**

LSS-Master sends:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2021	8	04h	X	00	00	00	00	00	00

X = 0: switch in Operation Mode, 1 = 1: switch in Configuration Mode

Byte 3...8 are reserved

The LSS - Slaves don't send a confirmation

## **7.2 Switch Mode Selective**

### **Switch of a single slave into Configuration Mode**

LSS-Master sends:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2021	8	40h	Vendor-ID	Vendor-ID	Vendor-ID	Vendor-ID	00	00	00

Byte 6...8 are reserved

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2021	8	41h	Product Code	Product Code	Product Code	Product Code	00	00	00

Byte 6...8 are reserved

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2021	8	42h	Revis. Nr.	Revis. Nr.	Revis. Nr.	Revis. Nr.	00	00	00

Byte 6...8 are reserved

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2021	8	43h	Serial Nr.	Serial Nr.	Serial Nr.	Serial Nr.	00	00	00

Byte 6...8 are reserved

LSS-Slave answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2020	8	44h	X	00	00	00	00	00	00

X = 0: Node is still in Operation Mode, x = 1: The switchover in Configuration Mode is carried out  
Byte 3...8 are reserved

### 7.3 Configure Node ID

#### Changing the node number (node ID) of a slave

LSS-Master sends:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2021	8	11h	x	00	00	00	00	00	00

X = new Node-ID  
Byte 3...8 are reserved

LSS-Slave answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2020	8	11h	Error-Code	Spec-Error	00	00	00	00	00

Byte 4...8 are reserved

#### Error-Code

0: o.k.

1: Node-ID out of range (1-127)

255: special error which is treated in byte3 (not implemented)

### 7.4 Configure Bit Timing Parameters

#### Changing of the baud rate of a slave

LSS-Master sends:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2021	8	13h	Table Selector	Table Index	00	00	00	00	00

Byte 4...8 are reserved

Table Selector: always 0 for Standard CiA table

Table Index	Baud rate	supported
0	1000 kBaud	no
1	800 kBaud	no
2	<b>500 kBaud</b>	<b>yes</b>
3	<b>250 kBaud</b>	<b>yes</b>
4	<b>125 kBaud</b>	<b>yes</b>
5	<b>100 kBaud</b>	<b>yes</b>
6	<b>50 kBaud</b>	<b>yes</b>
7	<b>20 kBaud</b>	<b>yes</b>
8	<b>10 kBaud</b>	<b>yes</b>

LSS-Slave answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2020	8	13h	Error-Code	Spec-Error	00	00	00	00	00

Byte 4...8 are reserved

#### Error-Code

0: o.k.

1: Bit Timing is not supported

255: special error which is treated in byte3 (not implemented)

## 7.5 Activate Bit Timing Parameters

### Activation of the new baud rate at all Slaves in Configuration mode

LSS-Master sends:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2021	8	15h	Switch Delay Low	Switch delay High	00	00	00	00	00

Byte 4...8 are reserved

Switch delay: delay time in ms

The LSS - Slaves don't send a confirmation

## 7.6 Store Configuration Protocol

### Nonvolatile storage of the changed data

LSS-Master sends:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2021	8	17h	00	00	00	00	00	00	00

Byte 2...8 are reserved

LSS-Slave answers:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
2020	8	17h	Error-Code	Spec-Error	00	00	00	00	00

Byte 4...8 are reserved

### Error-Code

0: o.k.

2: EEPROM error

255: special error which is treated in byte3 (not implemented)