Technical Documentation

– 0630 –

Digital Pressure Transmitter

CANopen® Protocol

1-6-30-628-058

SUCO Robert Scheuffele GmbH & Co. KG,
Keplerstraße 12-14, 74321 Bietigheim-Bissingen,

Tel.:+49-7142-597-0, Fax: +49-7142-597-19
Web: www.suco.de , E-Mail: info@suco.de

Table of content

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

1 History

Version	Date	Name	Change
1.00	12.09.2016	TB	Document created
1.01	15.03.2018	RW	Minor revision
1.02	19.06.2018	RW	Full revision
1.03	20.06.2018	RW	LSS Function revision

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

2 General Information

CANopen® is an open communication profile based on CAN (Controller Area Network), a bus system actually developed for data transfer in motor vehicles. CAN is internationally standardized in ISO 11898. CANopen is a widely used CAN application layer, developed by CAN in Automation (CiA®) which actively supports the international standardization of CAN protocols. CANopen consists of the protocol definitions (communication profile) and of the device profiles that define the data contents for the various device classes. So, the communication profile describes the "How" of the communication and the device profile defines the "What". For details see CiA DS-301 and CiA DS-404.

This technical documentation describes the basics for the transmission of pressure values, which are measured with a CANopen transmitter and especially with the SUCO 0630 pressure transmitter. This documentation doesn't represent the final stage of development of the pressure transmitter 0630. That means further changes or implementations are possible.
There is no warranty that all features described in this version of the document will be available for future development stages.

The following numeral systems are used in this documentation:

Index	Decimal	Binary	Hexadecimal
0	0d0	0b0000	0x0
1	0d1	0b0001	0x1
2	0d2	0b0010	0x2
3	0d3	0b0011	0x3
4	0d4	0b0100	0x4
5	0d5	0b0101	0x5
6	0d6	0b0110	0x6
7	0d7	0b0111	0x7
8	0d8	0b1000	0x8
9	0d9	0b1001	0x9
10	0d10	0b1010	0xA
11	0d11	0b1011	0xB
12	0d12	0b1100	0xC
13	0d13	0b1101	0xD
14	0d14	0b1110	0xE
15	0d15	0b1111	0xF
	d is the decimal notation	***b*** is the binary notation	***x*** is the hexadecimal notation

Leading zeros are only written down if they are necessary for the meaning of the value.

3 Default Settings

SUCO 0630 Pressure Transmitter default communication settings:

Transmission bit rate: 250 kbit/s
Node-ID: 0x20 (0d32)

These settings can be changed via the LSS protocol according to CiA DS-305 (see chapter 8).

The Boot-up and heartbeat consumer message is sent on CAN-ID 0x720 (0x700 + 0x20).
TPDOs' (measuring signal) are sent on CAN-ID 0x1A0 (0x180 + 0x20) and 0x2A0 (0x280 + 0x20), if activated.

Vendor-ID, revision number, product code and serial number can be read out with object index: 0x1018 sub-indices 1 – 4

The "Switch State Selective" function of the LSS protocol is only restrictively supported.
For using Switching State Selective, you have to apply the Vendor-ID, the Product code, Revision number and the Serial number. This allows a specific transmitter to be adjusted separately. By using the "Switched State Global" function, all nodes in the network react to the same Node-ID and bit rate changes.

To configure the 0630 transmitter only, use the "Switch State Selective" mode or connect it to a separate CAN network. In the latter case, you can use the "Switch State Global" command from the LSS protocol.

The device supports the 11-Bit Identifier specified according to CAN 2.0A as well as the 29-Bit Identifier specified according to CAN 2.0B. The physical CAN transmission of the SUCO 0630 pressure transmitter is defined according to ISO 11898-2 (high-speed CAN) and can be used up to transmission rates of 1 Mbit/s.

The following bus lengths should not be exceeded to avoid message collision or latency problems:

Transmission rate	Bus length
125 kbit/s	500 m
250 kbit/s	250 m
500 kbit/s	100 m
1000 kbit/s	25 m

Note: The actual usable bus length can be shorter, due to connectors or stub cables.

4 CANopen Communication Protocol

4.1 Network Management

The organization CAN in Automation (CiA®) created several network management rules for initiation and control of a CAN-network. These rules can be found in CiA DS-302.
In this chapter the most important features to start working with the SUCO 0630 pressure transmitter are described.

The CAN-messages are sent in packages that look as follows:

ID = 0x700 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x00	not used						

Data marked as “not used” have not to be sent to the network.
They can be ignored and the length of the CAN-message has to be adapted.

4.1.1 NMT Service

When powering on the transmitter a boot-up message will be sent out. This message contains only a single data byte, filled with 0x00. It’s created according to CiA DS-302.

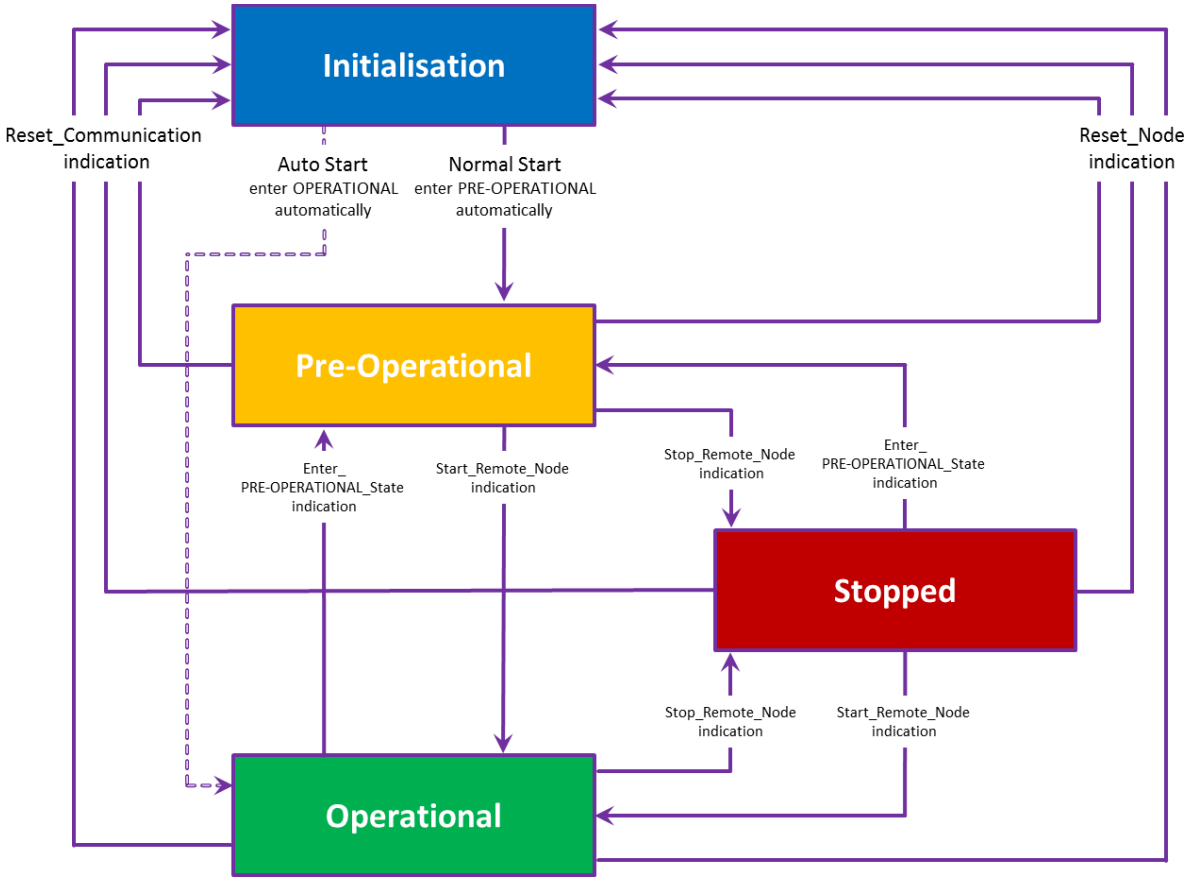
By default the SUCO 0630 transmitter has the Node-ID 32 (0x20).
So the boot-up message looks as follows.

ID = 0x720

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x00	not used						

With this information you are able to calculate the Node-ID from every received boot-up message.

After the boot-up process is completed the device will transition automatically into Pre-Operational State (see graphic below). If no NMT-Master is available, the device can be programmed to transition into Operational State right after the boot-up process via the “Auto Start” NMT startup Object (refer to chapter 4.1.2).



With the commands of an NMT master a certain one or all nodes in the network can be brought into different states. These commands are structured as follows:

ID = 0x000

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
CS	Node-ID	not used					

CS is the Command Specifier.

Following commands are possible:

CS: 0x01 – Enter Operational State / Start Node
0x02 – Enter Stopped State / Stop Node
0x80 – Enter Pre-Operational State
0x81 – Reset Node
0x82 – Reset Communication

Node-ID: 0x00 – Broadcast-ID, message affects every node in the network
0x01…0x7F – single node activated, all other nodes ignore this message

To obtain a pressure value, the node must be set to "Operational Mode" with its Node-ID or via the broadcast.

The following table shows which types of communication support are available in the individual states.

	PDO	SDO	SNYC	EMCY	NMT	LSS	HBP	TIME	Boot-Up
Initialization	-	-	-	-	-	-	-	-	(X)
Pre-Operational	-	X	X	X	X	X	X	X	-
Operational	X	X	X	X	X	X	X	X	-
Stopped	-	-	-	-	X	X	X	X	-

X stands for available

4.1.2 NMT Startup and Error Behavior

The device can be configured to change its NMT State autonomously for specific circumstances.

For the use of the 0630 transmitter in a network without a master, the device can be configured by changing the entries of object 0x1F80 so that it automatically switches to the operating state immediately after initialization.

ID = 600 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x23	0x80	0x1F	0x00	Startup	0x00	0x00	0x00

Startup:
- 0x00 Enter the Pre-Operational State after the initialization autonomously (default value)
- 0x02 Enter the Operational State after the initialization autonomously and execute the NMT start remote node with Node-ID equals zero ("Start All Nodes")
- 0x08 Enter the Operational State after the initialization autonomously

In the event of a serious device failure, by default the 0630 will enter the Pre-Operational State. With the following object 0x1029, this behavior can be changed to enter an alternative state or to remain in its current state.

ID = 600 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x2F	0x29	0x10	Case	State	not used		

Case: 0x01 Communication error
0x03 Analog Input error

State: 0x00 Change to Pre-Operational State (only if in Operational State)
0x01 No change of the NMT State
0x02 Change to Stopped State

4.1.3 Heartbeat Protocol

The intention of the heartbeat protocol is to monitor the working conditions of a node. So the node-status is sent out every adjusted timeslot. The node sends a short message that can be processed by the system-control unit. It looks as follows:

ID = 0x700 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
Status	not used						

Status: 0x00 Boot-up in process
0x04 Node Stopped
0x05 Node in Operational Mode
0x7F Node in Pre-Operational Mode

For activating the Heartbeat, you have to adjust the heartbeat-timer via SDO protocol (refer to chapter 4.2) in object 0x1017.

ID = 0x600 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x2B	0x17	0x10	0x00	Value (LSB)	Value (MSB)	0x00	0x00

The timer can be set via a 16-bit value. Each increasing hex number represents one millisecond.

For activating the heartbeat protocol with a 10 second periodic timer you have to send the following message:

ID = 0x600 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x2B	0x17	0x10	0x00	0x10	0x27	0x00	0x00

0x2710 = 0d10000 → 10 s

Writing the value 0d0 (0x0000) to the corresponding bits disables the heartbeat protocol.

The changes described above can only be made in the Operational and Pre-Operational States. Once activated, the heartbeat protocol sends the status of the node in all three states.

4.2 Service Data Object

The Service Data Object (SDO) is used for reading and writing data out of and into the object dictionary of a CANopen device.

The communication always takes place in a peer-to-peer connection between two nodes. A message from one node (SDO request) is always acknowledged by the other (SDO response).

The structure of a SDO message can be seen below.

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
CS	Object Index (LSB)	Object Index (MSB)	Sub-index	Data	Data	Data	Data

CS: Command Specifier
Object index: inquired object number
Sub-index: inquired object sub number

The table below shows the different command specifier of a SDO including the abort codes in case an error appears.

Command Specifier		Description
SDO Request		
0x22		Write data – size not specified
0x23		Write data – 4 Bytes
0x27		Write data – 3 Bytes
0x2B		Write data – 2 Bytes
0x2F		Write data – 1 Byte
0x40		Read data – size not specified
0x60/0x70		Toggling command for segmented reading
SDO Response		
0x60		Write access successful
0x41		Segmented Transfer Initiated
0x42		Read access successful – size not specified
0x43		Read access successful – 4 Byte
0x47		Read access successful – 3 Byte
0x4B		Read access successful – 2 Byte
0x4F		Read access successful – 1 Byte
0x80		Error occurred while writing/reading an object – Abort Code in Data Bytes
Abort Codes	0x0504 0001	Client/server command specifier not valid or unknown
	0x0601 0000	Unsupported access to an object
	0x0601 0001	Attempt to read a write only object
	0x0601 0002	Attempt to write a read only object
	0x0602 0000	Object does not exist in the object dictionary
	0x0604 0042	The number and length of the objects to be mapped would exceed PDO length
	0x0606 0000	Access failed due to hardware error
	0x0609 0011	Sub-index does not exist
	0x0609 0030	Value range of parameter exceeded (only for write access)
	0x0609 0031	Value of parameter written too high

4.2.1 Segmented Reading

Receiving a 0x41 in the command specifier of an SDO response means, that the requested value to be read contains more than 4 Byte and will be split into several messages of up to 7 byte each. The data bytes contain the number of bytes (data size) that needs to be transferred.

ID = 0x580+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x41	Object index (LSB)	Object index (MSB)	Sub-index	Data Size			

The subsequent requests alternate with a toggling command (TC) until all data bytes have been transmitted, starting with 0x60.

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
TC	Zero						

The following answers contain a toggling response (TR) and up to 7 bytes of data.

ID = 0x580+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
TR	Data Packages						

Bit # 4 changes its condition according to the toggling command byte to 0b or 1b.
In the last transmitted segment the toggling response (bit 3 – 1) contains the amount of not used data bytes in that last segment.

TR Bits	
7 – 5	000b
4	0b or 1b – toggling bit
3 – 1	# of bytes **NOT** containing any data
0	1b if last segment, else 0b

This device only supports segmented reading, writing is not allowed.

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

4.3 Synchronization Object

The CiA defined a synchronization message to prevent a bus overload. This message has to be sent out by a network master and looks as follows:

ID = 0x080

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
not used							

A SYNC message is sent with the ID 0x080 and an empty data field. If the transmitter should send out a PDO as a reaction to a SYNC message, the TPDO parameters in Object 0x1800 must be adjusted as described in (CiA) specification and chapter (4.6.1.2). Note that the Node must be put into Operational State before transmission can begin.
On default, the SUCO 0630 transmitter is set to a timer-based PDO transmission, which will transmit a PDO every 250 ms periodically.

With the object 0x1005 it is possible to configure the COB-ID of the Synchronization Object. The device is *not* able to generate SYNC messages.

Object index 0x1005 / Sub-index 0x00

Data bits				
MSB				LSB
31	30	29	28 11	10 0
x	0b (generate)	0b (base frame)	0 0000h (not used)	11-bit CAN-ID

By default the 8 byte value is set to 0x00000080

4.4 Emergency Object

Emergency messages are used for displaying an internal device error. If an error occurs, an emergency message is being sent once with a structure as seen below. The EMCY Identifier consists of the hex value 0x080 and the Node-ID. The transmitted Emergency Error Code identifies the error defined by means of CiA DS-301 and DS-404. Once the error vanishes or all errors have been fixed, the Error Code 0x0000 is being sent by the device.

ID = 0x080 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
LSB MSB Emergency Error Code		Error Register (0x1001)	Specific Code		not used (0x000000)		

The content of the Error Register (Object index 0x1001) provides error information and can be read out via an SDO or mapped to a PDO.

Object index 0x1001 / Sub-index 0x00

Data bits	
MSB 7 … 1	LSB 0
0000000b (not used)	0b or 1b no error/ADC error

The Pre-Defined Error Field (Object No. 0x1003) provides an error history for errors that have occurred and were reported via the emergency object. It contains the Error Code and a description of the occurred errors.

A sample of Error Codes and their descriptions can be found in the table below.

Error Code	Description	Error Type
0x0000	–	Error reset or no error
0x8100	Communication - generic	Monitoring – Communication
0x8110	CAN Overrun (objects lost)	Monitoring – Communication
0x8120	CAN in error passive mode	Monitoring – Communication
0x8130	Heartbeat error	Monitoring – Communication
0x8140	Recovered from bus off	Monitoring – Communication
0x8150	CAN-ID collision	Monitoring – Communication
0x5030	Internal sensor error	CANopen device hardware
	Description	Specific Code
	AD-Watchdog error	0xC555
	Bonding wire broken	0xCFCF

The COB-ID of an emergency message can be read and changed in the COB-ID EMCY object (0x1014).

Object index 0x1014 / Sub-index 0x00

Data bits				
MSB 31	30	29	28 … 11	10 … 0 LSB
0b/1b (valid/invalid)	0b (reserved)	0b (base frame)	0 0000h (not used)	11-bit CAN-ID

4.5 Time Stamp Object

The 0630 Transmitter can be configured to produce or to consume Time Stamp Objects.
When used as a TIME consumer, the millisecond-based timestamp can be synced to a network-wide time base.
The typical CAN-ID for the timestamp object according to the CANopen specifications is 0x100h. The data delivered by this message looks as follows:

ID = 0x100

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
Days		Milliseconds				not used	

The behavior and the CAN-ID of the Time Stamp Object can be altered using the “COB-ID Time Stamp” Object index 0x1012.

Object index 0x1012 / Sub-index 0x00

Data bits				
MSB				LSB
31	30	29	28 11	10 0
1b/0b (consuming enabled)	0b/1b (producing enabled)	0b (base frame)	0 0000h (not used)	11-bit CAN-ID (00100000000b) default

The two objects 0x3140 and 0x3141 have access to read the current time stamp value of the CANopen device and can be mapped on a TPDO. If a producer of a network-wide time base is available in the CAN network, the obtained time stamp values correspond to the following values:

- *0x3140 – Days since 1 January, 1984*
- *0x3141 – Milliseconds after midnight*

4.6 Process Data Object (PDO)

The process data object is used to transmit (real-time) process values (e.g. pressure value). A PDO can be configured to send data between 1 and 8 bytes in size. Compared to SDOs, no protocol overhead is used. The transmission takes place according to the "broadcasting" principle; one node transmits its data to all other nodes. The transmission of PDOs is only possible in the Operational State. Furthermore, the data to be transmitted can be mapped individually. Further information can be found in chapter (4.6.2).

4.6.1 PDO-Transmission Parameter

The SUCO 0630 has an implemented TX-stage to adjust several transmitting parameters.
As already mentioned by default the pressure value is sent out periodically ("timer-driven").
For generating "SYNC-driven" or "measuring event driven" messages you have to adjust the "Transmission Type" field at object 0x1800/0x02 (i.e. Object Index = 0x1800 and Sub-index = 0x02). Take care to set the transmitter in Pre-Operational or in Operational Network Management State.
To receive the adjusted transmission type of the node you have to send out following message:

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x40	0x00	0x18	0x02	0x00	0x00	0x00	0x00

You will receive the following response:

ID = 0x580 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x42	0x00	0x18	0x02	0x**XX**	0x00	0x00	0x00

In Byte #4 the transmitting information is coded. Following codes are possible:

Byte 4	1...240 (0x01 – 0xF0)	value transmission every Nth sync message
	241...253 (0xF1 – 0xFD)	reserved values, setting will be ignored
	254 (0xFE)	timer driven transmission (default value)
	255 (0xFF)	measuring event driven transmission

4.6.1.1 Timer-Driven TPDO

By default the transmission of a PDO is timer based and its transmission time is set to a period of 250 milliseconds. The transmission parameters can be adjusted in object 0x1800/0x05 (i.e. Object index = 0x1800 and Sub-index = 0x05). In Byte 4 and 5 the adjusted transmission time is coded 16 bit long directly in milliseconds.

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x22	0x00	0x18	0x05	0x**XX**	0x**XX**	0x00	0x00

The node will answer as follows:

ID = 0x580+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x60	0x00	0x18	0x05	0x00	0x00	0x00	0x00

The value of 0x0000 will disable the transmission. Values lower than 10 ms are not recommended.

To return to the timer-driven transmission from another transmission event, the transmission parameter must be set as followed:

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x22	0x00	0x18	0x02	0xFE	0x00	0x00	0x00

The node will answer as follows:

ID = 0x580+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x60	0x00	0x18	0x02	0x00	0x00	0x00	0x00

Then, as already mentioned, the transmission time must be set to the desired value.

4.6.1.2 SYNC-Driven TPDO

By adjusting the transmission parameter to a value between 0x01 and 0xF0 (1 – 240), a TPDO will be sent out with every Nth SYNC message.

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x22	0x00	0x18	0x02	0x**XX**	0x00	0x00	0x00

The node will answer as follows:

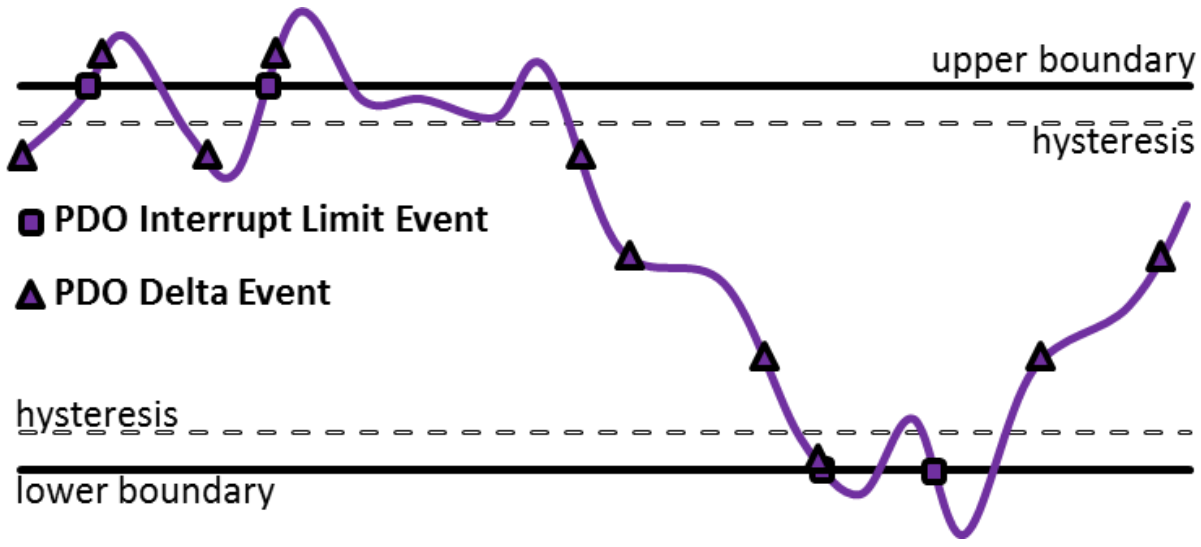
ID = 0x580+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x60	0x00	0x18	0x02	0x00	0x00	0x00	0x00

4.6.1.3 Measurement Event Driven TPDO

PDOs' triggered by reaching specific values of the measured pressure/temperature can be set by writing 0xFF to the 2nd sub-index of the TPDO communication parameter. There are two possible events to trigger a PDO, the "Delta Event" and the "Interrupt Limit Event".

The functional principle can be seen in the graphic below.



Interrupt Limit Event

For the “Interrupt Limit Event” there is an upper and a lower boundary, which in each case have to be crossed to trigger the transmission of a PDO. A hysteresis, which corresponds to 1% of the total measuring span ensures, that no permanent transmission is possible.

The following steps have to be taken to setup this event: Write 0xFF to Object index = 0x1800 with Sub-index = 0x02

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x22	0x00	0x18	0x02	0xFF	0x00	0x00	0x00

Disable the event timer in the communication parameter, if it’s still active.

ID = 600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x22	0x00	0x18	0x05	0x00	0x00	0x00	0x00

Set the lower boundary by writing its value to the 1st sub-index of the objective index 0x6134 for pressure measuring and to the 2nd sub-index for temperature measuring.

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x22	0x34	0x61	0x01	0x**XX**	0x**XX**	0x**XX**	0x**XX**

Set the upper boundary by writing its value to the 1st sub-index of the objective index 0x6135 for pressure measuring and to the 2nd sub-index for temperature measuring.

ID = 600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x22	0x35	0x61	0x01/02	0x**XX**	0x**XX**	0x**XX**	0x**XX**

In order to disable this event, set the upper boundary to a value of 0x7F7FFFFF and the lower boundary to a value of 0xFF7FFFFF.

Delta Event

The "Delta event" is triggered if the current process value differs from the last transmitted process value by more than the value set in the Delta object.
The following steps have to be taken to setup this event: Write 0xFF to Object index = 0x1800 and Sub-index = 0x02

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x22	0x00	0x18	0x02	0xFF	0x00	0x00	0x00

Set the delta value by writing its value to the 1st sub-index of the objective index 0x6133 for pressure measuring and to the 2nd sub-index for temperature measuring.

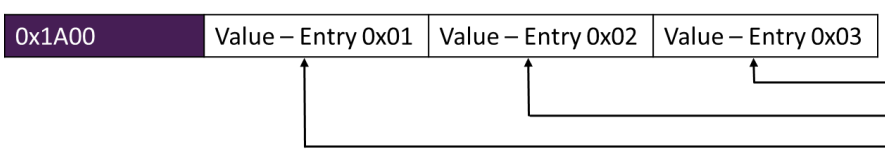
ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x22	0x33	0x61	0x01/02	0x**XX**	0x**XX**	0x**XX**	0x**XX**

In order to disable this event, write a value of 0 to Object index = 0x6133 with Sub-index = 0x01 respective Sub-index = 0x02.

4.6.2 TPDO-Mapping Parameter

Object #	Object Sub #	Description	Length
0x6130	0x01	Process Value	32 bit
0x6150	0x01	Analog Input Status	8 bit
0x3140	0x00	Time Stamp – Days	16 bit
Object #	**Object Sub #**	**Object Content**	
0x1A00	0x00	3	
0x1A00	0x01	0x6130 0x01 0x20	
0x1A00	0x02	0x6150 0x01 0x08	
0x1A00	0x03	0x3140 0x00 0x10	



Both TPDOs' can be mapped with a valid number (the combined length of the mapped objects must not exceed the limit of 8 byte (64bit)) of process variables.

Changing the Mapping of a TPDO can be done in Pre-Operational or Operational Mode.

Objects allowed for mapping:

Object Index	Sub-index	Description	Data Length
0x1001	0x00	Error Register	Uint8 (0x08)
0x6150	0x01	Analog Input Status – Channel 1	Uint8 (0x08)
	0x02	Analog Input Status – Channel 2	Uint8 (0x08)
0x7100	0x01	Field Value – Channel 1	Uint16 (0x10)
	0x02	Field Value – Channel 2	Uint16 (0x10)
0x6130	0x01	Process Value – Channel 1	Float32 (0x20)
	0x02	Process Value – Channel 2	Float32 (0x20)
0x7130	0x01	Process Value – Channel 1	Uint16 (0x10)
	0x02	Process Value – Channel 2	Uint16 (0x10)
0x9130	0x01	Process Value – Channel 1	Uint32 (0x20)
	0x02	Process Value – Channel 2	Uint32 (0x20)
0x3140	0x00	Time Stamp – Days	Uint16 (0x10)
0x3141	0x00	Time Stamp – Milliseconds	Uint32 (0x20)

The following steps have to be undertaken in case of re-mapping a specific TPDO:

- "Destroy" the TPDO by setting bit 31 to 1b of sub-index 01h of the according TPDO Communication Parameter.
- Disable mapping by setting sub-index 00h of the according TPDO mapping parameter to 00h.
- Modify mapping by changing the values of the corresponding sub-indices.
- Enable mapping by setting sub-index 00h of the according TPDO mapping parameter to the number of mapped objects.
- "Create" the TPDO by setting bit 31 to 0b of sub-index 01h of the according TPDO communication parameter.

Example

Mapping the float32 pressure Process Value (Object index 0x6130, Sub-index 0x01) and its UInt16 Field Value (Object index 0x7100, Sub-index 0x01) to the first TPDO.

Step 1: "destroy" TPDO1 by setting bit 31 to 1b.

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x23	0x00	0x18	0x01	0xA0	0x01	0x00	0x80

Step 2: disable the mapping of TPDO1.

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x2F	0x00	0x1A	0x00	0x00	0x00	0x00	0x00

Step 3: mapping of the Process Value (Object index 0x6130, Sub-index 0x01) to the 1st entry. Byte 4 defines the length (32bit) of the process variable

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x23	0x00	0x1A	0x01	0x20	0x01	0x30	0x61

Step 4: mapping of the Physical Unit (Object index 0x6131, Sub-index 0x01) to the 2nd entry.

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x23	0x00	0x1A	0x02	0x10	0x01	0x00	0x71

Step 5: enable mapping of TPDO1 by writing the number of two process variables to it.

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x2F	0x00	0x1A	0x00	0x02	0x00	0x00	0x00

Step 6: “create” TPDO1 by setting bit #31 back to 0b0.

ID = 0x600+Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x23	0x00	0x18	0x01	0xA0	0x01	0x00	0x00

4.6.3 Receiving pressure values

To receive a pressure value from the SUCO 0630 pressure transmitter, send following CAN message:

ID = 0x000

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x01	0x00	not used					

After starting the network the transmitter will send its pressure value (by default) periodically with a period of 250 milliseconds. Adjusting the periodic time is described in chapter (4.6.1.1).

ID = 0x180 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
Value (LSB)	Value (MSB)	not used					

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

5 Communication Specific Functions

All objects of the (CiA DS-301) communication profile which were not discussed in the previous chapters are listed and explained below.

0x1000 – Device Type

- Data Type: Unsigned Integer 32
- Access: Constant

This object describes the type of the device and its functionality. By reading the object you get the following message.

ID = 0x580 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x43	0x00	0x10	0x00	0x94	0x01	0x02	0x80

The lower 16 bit (0x0194) carries the Device Profile Number – 404. The upper 16 bit contains additional information according to DS-404.

0x1002 – Manufacturer Status Register

- Data Type: Unsinged Integer 32
- Access: Read Only

The object contains information for the status of the first (pressure) and second (temperature) measurement channel.

Object index 0x1002 / Sub-index 0x00

Data bits					
MSB					LSB
31	16	15	8	7	0
Not used		Status Byte – Temperature		Status Byte – Pressure	

Further information regarding the status bytes can be found in the entry of object 0x6150 (refer to chapter 6).

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

0x1008 – Manufacturer Device Name

- Data Type: Visible String
- Access: Constant

This object shall provide the name of the device as given by the manufacturer. By reading this object a segmented transfer will be initiated (refer to chapter 4.2.1)

ID = 0x580 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x41	0x08	0x10	0x00	0x0C	0x00	0x00	0x00

By following the instructions in this Chapter, you will receive the ASCII-Code for “0630-CANopen”

0x1009 – Manufacturer Hardware Version

- Data Type: Visible String
- Access: Constant

This object contains the manufacturer hardware version description. The version number is supplied as ASCII code with the following structure:

ID = 0x580 + Node-ID

Data				
Byte 0				Byte 7
0	1	2	3	4
v	Y	Y	W	W

“vYYWW” – YY represents Year, WW represents the Week

0x100A – Manufacturer Software Version

- Data Type: Visible String
- Access: Constant

This object contains the manufacturer hardware version description. The version number is supplied as ASCII code with the following structure:

ID = 0x580 + Node-ID

Data				
Byte 0				Byte 7
0	1	2	3	4
v	Y	Y	W	W

“vYYWW” – YY represents Year, WW represents the Week

0x1010 – Store Parameters

- Data Type: Unsigned Integer 32
- Access: Read & Write

With this object, parameters of objects with write access can be stored in a non-volatile memory. To avoid accidental saving of parameters, the save command is only executed if the ASCII word "evas" (which means “save” in reverse order) is written to the first sub-index (0x01). Successful saving is acknowledged with 0x60 SDO command specifier.
ID = 0x620 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x23	0x10	0x10	0x01	0x73	0x61	0x76	0x65

By reading the sub-index (0x01) of this object, the device provides information about its storage functionality according to DS-301, in this case it returns the value 1b (binary format), which stands for "Save parameter on command".

0x1011 – Restore default parameters

- Data Type: Unsigned Integer 32
- Access: Read & Write

With this object, all parameters except the bit rate and the Node-ID are restored to the factory settings. To avoid accidental restoring of parameters, the command is only executed if the ASCII word "daol" (which means “load” in reverse order) is written to the first sub-index (0x01).

Successful restoring is acknowledged with 0x60 SDO command specifier.

ID = 0x620 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x23	0x11	0x10	0x01	0x6C	0x6F	0x61	0x64

By reading the sub-index (0x01) of this object, the device provides information about its storage functionality according to DS-301, in this case it returns the value 1b (binary format), which stands for "CANopen device restores parameters”.

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

0x1018 – Identity Object

- Data Type: Unsigned Integer 32
- Access: Read Only

This object contains general information about the device. The specific information assigned to the single sub-indices is listed below:

- 0x01 – Vendor-ID (contains unique value that is assigned to each manufacturer)
- 0x02 – Product Code
- 0x03 – Revision Number
- 0x04 – Serial Number

ID = 0x580 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x43	0x18	0x10	0x01	0x2D	0x04	0x00	0x00

"042D" is the unique Vendor-ID for SUCO CANopen products, provided by the (CiA).

To configure the device in "Switch Selective Mode" (refer to chapter 8.2), all of the number mentioned must be used.

0x1020 – Verify Configuration

- Data Type: Unsigned Integer 32
- Access: Read & Write

In this object, the date and time of the last configuration can be stored in order to check at a later point in time whether the stored configuration corresponds to the expected configuration.

Object index 0x1020

Sub-index	Description
0x01	Configuration date
0x02	Configuration time

The first sub-index (0x01) represents the number of whole days since January 1, 1984. The second sub-index represents the number of milliseconds since midnight.

Example: Mai 01, 2016, 11:36:12 AM

- 11809 days (0x2E21) in Subentry 0x01
- 41772000 milliseconds (0x27D63E0) in Subentry 0x02

A change in one of the device registers causes the stored values to be set to zero.

6 Device Specific Functions

0x6110 – AI Sensor Type

- Data Type: Unsigned Integer 16
- Access: Constant

This object indicates which type of sensor is connected to the analog input.

Sub-index #1 defines the primary measurement channel, sub-index #2 the secondary measurement channel.

Object index 0x6110

Sub-index	Value	Description
0x01	0x5A (090d)	Sensor type: pressure transducer
0x02	0x64 (100d)	Sensor type: temperature transducer

0x6114 – ADC Sample Rate

- Data Type: Unsigned Integer 32
- Access: Constant

This object gives information about the reciprocal conversion rate of the A/D measurement channels.

Sub-index #1 defines the primary measurement channel, sub-index #2 the secondary measurement channel.

Object index 0x6114

Sub-index	Value	Description
0x01	0x3E8 (1000d)	Reciprocal sample rate – channel 1
0x02	0x3E8 (1000d)	Reciprocal sample rate – channel 2

0x7100 – Field Value

- Data Type: Integer 16
- Access: Read Only

Depending on the sub-index, this object contains a pressure or a temperature value that is not yet scaled to the physical unit of the quantity to be measured.

Object index 0x7100

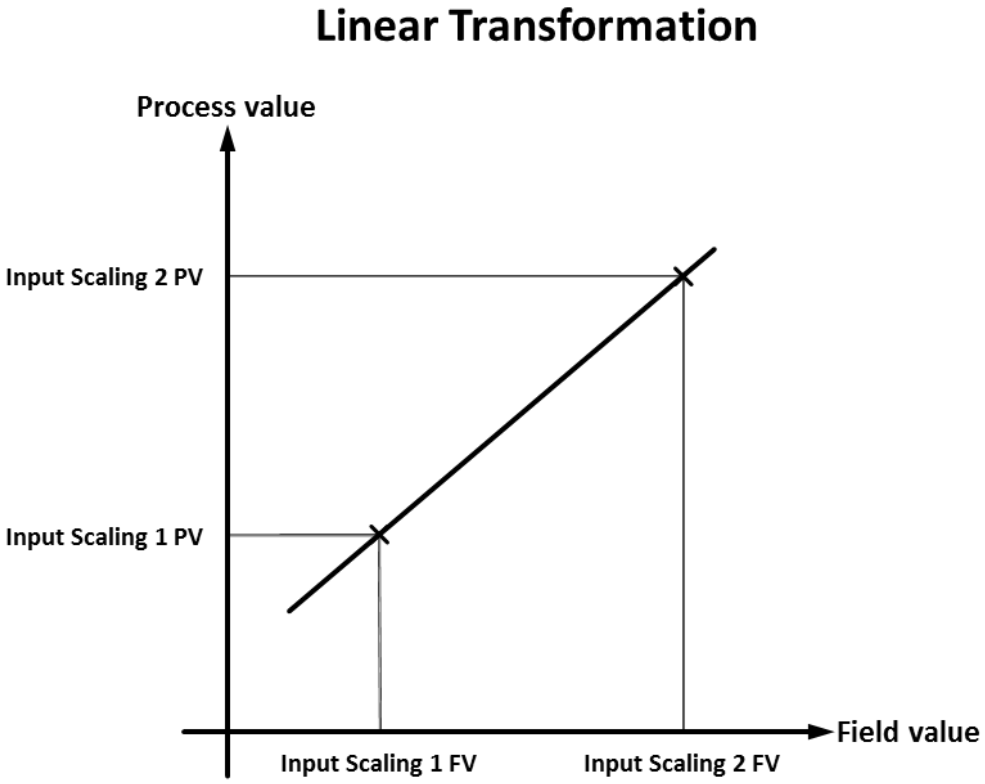
Sub-index	Description
0x01	Unscaled pressure value
0x02	Unscaled temperature value

0x7120 / 0x7122 – FV Input Scaling

- Data Type: Integer 16
- Access: Constant

The objects contain the field values for the first and second calibration point with which the linear transformation is performed and **cannot be changed**. The first calibration point corresponds to the field value 0; the second calibration point corresponds to the field value of 20000.

The linear transformation converts the measured values of the analog-to-digital converter into the physical dimension SI units of the measured quantity.



0x6/9121 / 0x6/9123 – PV Input Scaling

- Data Type: Float 32 / Integer 32
- Access: Read & Write / Read & Write

These objects contain the process values for the lower (0x6/9121) and the upper (0x6/9123) calibration point of the linear transformation. Integer values can be set with the objects 9121/9123 and floating point numbers with the objects 6121/6123.

Example: If the pressure of a 0 to 10 bar transmitter is to be displayed in PSI instead of bar, the first calibration point must be set to 0 and the second calibration point to 145,038 (0x431109BA in hexadecimal floating point notation).
To avoid confusion, the physical unit in object 0x6131 should also be changed correctly.

0x6126 / 0x6127 – PV Scaling Factor / PV Scaling Offset

- Data Type: Float 32 / Float 32
- Access: Read & Write / Read & Write

The objects 0x6126 (Scaling Factor) and 0x6127 (Scaling Offset) offer a further possibility to convert the field values into process value. The formula for the calculation is as follows:

$$PV = FV \cdot Factor - Offset$$

Keep in mind to set the offset first, as changing the scaling factor results in internal calculations that include the offset value.

Example:

If the pressure of a 0 to 10 bar transmitter is to be displayed in PSI instead of bar, set the Scaling Offset to 0 and calculate the Scaling Factor as follows:

$$SF = \frac{145.038}{20000} = 0.0072519$$

After writing the calculated value (0.0072519 equals 0x3BEDA159 in hexadecimal floating point notation) into the scaling factor object, the process value is calculated according to the formula.

Note: To avoid inconsistencies when changing the scaling factor and offset, the calibration points of the objects X121h and X123h are automatically adjusted.

0x6/7/9130 – Process Value

- Data Type: Float32 / Integer 16 / Integer 32
- Access: Read Only / Read Only / Read Only

These objects contain the result of the linear transformation in their respective data type and display the measured quantity scaled in the physical unit of the process value.

Object index 0x6130/0x7130/0x9130

Sub-index	Description
0x01	pressure value
0x02	temperature value

0x6131 – Physical Unit

- Data Type: Unsigned Integer 32
- Access: Read & Write

The physical unit of the process values is defined in this object. The coding of the physical units and prefixes is done according to the CiA DR – 303-2.

Object index 0x6131

Data			
Byte 0			Byte 7
8	7	6	5 0
Prefix (0x00)	SI Numerator (0x**XX**)	SI Denominator (0x00)	Not used (0x000000000000)

By default the SI Numerator for the primary measuring channel will be 0x4E (“bar”) and 0x2D (“degree Celsius”) for the secondary measurement channel. Further examples for SI Numerators that can be taken:

- 0x22 (“Pascal”)
- 0xAB (“PSI”)
- 0x05 (“Kelvin”)
- 0xAC (“degree Fahrenheit”)

Note: Changing the physical unit in this object has no direct effect on the output process values. Change your Scaling Parameters 0x6126 & 0x6127 accordingly to your desired Physical Unit or vice versa.

0x6132 – Decimal Digits Process Value

- Data Type: Unsigned Integer 8
- Access: Read & Write

This object indicates the number of decimal digits following the decimal point. This will be done for PVs of the Integer16 and Integer32 data types.

Example: If the number of decimal places is set to 3, an actual pressure value of 5.749 bar is displayed as a 5749 integer value.

Object index 0x6132 / Sub-index index 0x01/ 0x02

Data				
Byte 0				Byte 7
7 4	3	2	1	0
Not used	X	X	X	X

The number of decimal places corresponds to the set bit value, which can be between 0 - 9 for Int32 and 0 – 4 for Int16.

0x6133 – Analog Input Interrupt Delta Input Process Value

- Data Type: Float 32
- Access: Read & Write

This object displays the configured interrupt delta. The delta value must correspond to the physical unit of the process value and has to be set in the floating point notation. A delta value of 0 deactivates this function.

Object index 0x6133

Sub-index	Description
0x01	delta value for the pressure measuring channel
0x02	delta value for the temperature measuring channel

Note: The functional principle and the application can be found in chapter (4.6.1.3)

0x6134 – Analog Input Interrupt Lower Limit Input Process Value

- Data Type: Float 32
- Access: Read & Write

This object indicates the configured lower limits for the interrupt-enabled analog inputs. The lower interrupt value must correspond to the physical unit of the process value and has to be set in the floating point notation. In order to disable this event, set the lower boundary to a value of 0xFF7FFFFF.

Object index 0x6134

Sub-index	Description
0x01	lower interrupt value for the pressure measuring channel
0x02	lower interrupt value for the temperature measuring channel

Note: The functional principle and the application can be found in chapter (4.6.1.3)

0x6135 – Analog Input Interrupt Upper Limit Input Process Value

- Data Type: Float 32
- Access: Read & Write

This object shall indicate the configured upper limits for interrupt-enabled analog inputs. The upper interrupt value must correspond to the physical unit of the process value and has to be set in the floating point notation. In order to disable this event, set the upper boundary to a value of 0x7F7FFFFF.

Object index 0x6135

Sub-index	Description
0x01	upper interrupt value for the pressure measuring channel
0x02	upper interrupt value for the temperature measuring channel

Note: The functional principle and the application can be found in chapter (4.6.1.3)

0x6150 – Status Byte

- Data Type: Unsigned Integer 8
- Access: Read Only

This object provides information on the internal status of the analog input channels.

Object index 0x6150 / Sub-index 0x01 / 0x02

Data			
Byte 0			Byte 7
7 3	2	1	0
Not used	Negative overload	Positive overload	Not valid

Sub-index #1 defines the primary measurement channel, sub-index #2 the secondary measurement channel.

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

7 SUCO Specific Functions

Some further features besides the CANopen protocol were added to the SUCO 0630 CANopen transmitter.

0x3124 / 0x3125 – FV Offset

- Data Type: Integer 16 / Float 32
- Access: Read & Write / Read & Write

This object defines an offset value which is directly subtracted from the field values. By default, the field values are between 0 and 20000; the offset value can be subtracted as either an integer 16 or float 32 variable (see flow chart in the Appendix).

The value changes for the different measuring channels are accessible as shown in the table below:

Object index 0x3124 / 0x3125

Sub-index	Description
0x01	pressure measuring channel
0x02	temperature measuring channel

0x3126 – Autozero

- Data Type: Unsigned Integer 32
- Access: Write Only

This object will define the current field value as the new zero point by writing the ASCII word "zero" into its directory. This means that the current pressure value is subtracted from the field values as in object 0x3124/0x3125.

This can be done for the pressure measurement channel (sub-index 0x01) as well as for the temperature measurement channel (sub-index 0x02).

Object index 0x3126 / Sub-index 0x01 / 0x02

Data bits			
MSB			LSB
32 25	24 17	16 9	8 0
0x6F	0x72	0x65	0x7A
“o”	“r”	“e”	“z”

Due to the little Endian, the ASCII word has to be written back-to-front.

Note: Only field values that deviate at least ±10% from the original zero point are accepted as a new zero point.

0x3130 / 0x3131 – Extreme Values – Operating / Service Life

- Data Type: Float 32 / Float 32
- Access: Read Only / Read Only

These objects store the occurred global minima and maxima of the field values for both measurement channels. While the values stored in the "Extreme Values - Operating" object (0x3130) are volatile and are erased in the event of a power off, the values stored in the "Extreme Values - Service Life" object (0x3131) are retained.

The field values are stored before offset and scaling calculations are performed. By reading the values, the current settings for the linear transformation, the limit and the offset are applied to them.

Object index 0x3130 / 0x3131

Sub-index	Description
0x01	Minimum – pressure measuring channel
0x02	Maximum – pressure measuring channel
0x03	Minimum – temperature measuring channel
0x04	Maximum – temperature measuring channel

The stored values can be manually deleted by writing the ASCII word "rset" into the sub-index 0x00.

Object index 0x3130 / 0x3131 Sub-index 0x00

Data bits			
MSB			LSB
32 25	24 17	16 9	8 0
0x74	0x65	0x73	0x72
“t”	“e”	“s”	“r”

Due to the little Endian, the ASCII word has to be written back-to-front.

0x3200 / 0x3201 / 0x3202 – Process Value Limit

- Data Type: Unsigned Integer 8 / Float 32 / Float 32
- Access: Read & Write / Read & Write / Read & Write
-

These objects allow limiting the process values with an upper and lower limit, both for the pressure measuring channel and for the temperature measuring channel.

Object index 0x3200/0x3201/0x3202

Sub-index	Description
0x01	pressure measuring channel
0x02	temperature measuring channel

Reading one of the sub-indices of object 0x3200 contains the information whether a limit is activated on the corresponding measurement channel or not. A “1” indicates that a limitation is active; a “0” indicates that it is deactivated.

Object 0x3201 and 0x3202 can be used for setting a lower and/or upper limitation for the measuring channels.

For deactivation of the lower (0x3201) or upper (0x3202) limitation, the float 32 values 0x7F7FFFFFFFF or 0xFF7FFFFFFF must be written into the respective measuring channel.
By writing a "0" into one of the sub-indices of object 0x3200, both the lower and the upper limit of the corresponding measurement channel are deactivated, whereby the float values mentioned above are written into the respective registers.

0x4000 – Endianness – Measured Values

- Data Type: Unsigned Integer 8
- Access: Read & Write

With this object the byte order, also called "Endianness", of the measured value data can be changed. They can be listed after the Little Endian (CANopen standard) by writing a “0” to the object or the Big Endian by writing a “1” to the object.

ID = 600 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x2F	0x00	0x40	0x00	Endianness	0x00	0x00	0x00

Endianness:

- 0x00 = Little Endian
- 0x01 = Big Endian

Example:

A pressure value of 8.74 bar as a Float 32 data type (0x410BD70A) will be displayed as follows:

Big Endian			
0x41	0x0B	0xD7	0x0A
Little Endian			
0x0A	0xD7	0x0B	0x41

Note: Changing the byte order only affects the output of measured values via SDOs and TPDOs.

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

0x4001 – Configure Node-ID

- Data Type: Unsigned Integer 8
- Access: Read & Write

This object allows the node ID of the device to be configured without having to use the LSS (see chapter 8.3.).

Object index 0x4001

Sub-index	Description
0x01	Changes to the Node-ID, but no changes to the respective COB-IDs
0x02	Changes to the Node-ID with corresponding COB-ID assignment

Changing the Node-ID by writing a new node number into the sub-index 0x01 is equivalent to changing it via the LSS and has no effect on the its COB IDs.
By writing a new node number to sub-index 0x02, the Node-ID of the device and all COB-IDs with a Node-ID influence are changed.

- COB-ID EMCY: 0x080 + **new Node-ID**
- COB-ID TPDO1: 0x180 + **new Node-ID**
- COB-ID TPDO2: 0x280 + **new Node-ID**

Note: To store these changes permanently, they must to be written to the EEPROM via the "Store Parameters" object (0x1010).

After successfully modifying the Node-ID, a restart of the device is simulated, including the transmission of a boot-up message. Notice that the restart is performed according to the current settings in the "NMT Startup Behavior" object (0x1F80).

0x4010 – Hardware Settings

- Data Type: Unsigned Integer 8
- Access: Constant (sub-index 0x02, Read & Write, if switchable termination is included)

This object contains information about an existing integration of switchable termination and galvanic isolation (not yet implemented, reserved for future development) in the device.

Object index 0x4010

Sub-index	Description
0x01	Indicates whether a switchable termination resistor is implemented: - 0 = not integrated - 1 = integrated
0x02	Enables the termination to be switched on and off if it is implemented
0x03	Indicates whether galvanic isolation is integrated: - 0 = not integrated - 1 = integrated (not yet implemented, reserved for future development)

If a switchable termination is implemented, it can be switched on and off by writing to the sub-index 0x02:

ID = 0x600 + Node-ID

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x2F	0x10	0x40	0x02	Switch	0x00	0x00	0x00

Switch:

- 0x00 = disable
- 0x01 = enable

0x4020 – Comment String

- Data Type: Visible String
- Access: Read & Write

This object can store a user specific comment. By default, the string is empty.

Object index 0x4020

Sub-index	Description
0x01 – 0x4F	String Symbols
0x50	Null Termination Symbol (Read Only)

Note: The entire comment can contain up to 78 string symbols (plus zero termination).
This means that only one symbol is permitted per sub-index. A segmented SDO transmission is not possible.

0x4021 – Service String

- Data Type: Visible String
- Access: Read & Write

This object can store a user specific service message. By default, the string is empty.

Object index 0x4021

Sub-index	Description
0x01 – 0x3B	String Symbols
0x3C	Null Termination Symbol (Read Only)

Note: The entire message can contain up to 58 string symbols (plus zero termination).
This means that only one symbol is permitted per sub-index. A segmented SDO transmission is not possible.

0x4022 – Device Name

- Data Type: Visible String
- Access: Read & Write

This object is used to change the content and the output of the Manufacturer Device Name (0x1008, chapter 5). By default, the string stores the name "0630-CANopen".

Object index 0x4022

Sub-index	Description
0x01 – 0x3B	String Symbols
0x3C	Null Termination Symbol (Read Only)

Note: The entire name can contain up to 58 string symbols (plus zero termination).
This means that only one symbol is permitted per sub-index. A segmented SDO transmission is not possible.

0x4030 – Configure Serial Number

- Data Type: Unsinged Integer 32
- Access: Read & Write

This object stores a 64 bit long serial number. The lower 32 bits are required when using the Switch Mode Selective of the LSS function (refer to Chapter 8). They are also displayed by object 0x1018.

Object index 0x4030

Sub-index	Description
0x01	Lower 32 bit
0x02	Upper 32 bit

Note: Writing a value to the lower 32 bits (sub-index 0x01) sets the upper 32 bits to zero (sub-index 0x02). It is therefore always necessary to change the lower 32 bits first.

0x4031 – Service Time

- Data Type: Unsigned Integer 32
- Access: Read & Write

This object can store the date, e.g. in *Unix Time Format*, of the last service on the device.

Object index 0x4031

Sub-index	Description
0x01	Lower 32 bit
0x02	Upper 32 bit

Note: Writing a value to the lower 32 bits (sub-index 0x01) sets the upper 32 bits to zero (sub-index 0x02). It is therefore always necessary to change the lower 32 bits first.

8 LSS-Slave Function

The SUCO 0630 Pressure Transmitter has an integrated LSS-Slave functionality.
By using the LSS (Layer Setting Service) you can set the parameters of Node-ID and Bitrate within the CAN network. The changes can be made globally for all nodes in the network using the Switch State Global Protocol, or selectively for only one special node in the network using the Switch State Selective Protocol. The CAN messages required for the individual LSS functions are described in the following sections.

8.1 Switch State Global Protocol

Use this service to toggle the CAN nodes between the waiting mode and configuration mode. The following CAN message must be sent by an LSS master in order to change the operating mode globally:

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x04	Mode	0x00	0x00	0x00	0x00	0x00	0x00

Byte 1 allows you to set the destination mode for all nodes in the network.

Mode:

- 0x00 = Waiting State
- 0x01 = Configuration State

This command does not generate a response from the nodes in the network.

8.2 Switch State Selective Protocol

To switch a single node, the LSS slave address of the node must be known to the LSS master. You must address the individual nodes via the identification objects that can be retrieved via the Identity Object (0x1018) of the device or via the LSS Inquiry Protocol (see Chapter 8.6). The following 4 CAN messages must be sent by the LSS master.

1) Sending the Vendor-ID

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x40	Vendor ID (LSB)	Vendor ID	Vendor ID	Vendor ID (MSB)	0x00	0x00	0x00

The SUCO Vendor-ID consists of the 4 Byte long number, 0x0000042D. It must be transmitted in a mirrored way. That means the last byte must be transmitted at first.

As a result:

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x40	0x2D	0x04	0x00	0x00	0x00	0x00	0x00

2) Sending the Product-Code

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x41	Product Code (LSB)	Product Code	Product Code	Product Code (MSB)	0x00	0x00	0x00

The Product Code consists of a 4 Byte long code. It must be transmitted in a mirrored way. That means the last byte must be transmitted at first.

As a result:

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x41	0x**XX**	0x**XX**	0x**XX**	0x**XX**	0x00	0x00	0x00

3) Sending the revision number

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x42	Revision (LSB)	Revision	Revision	Revision (MSB)	0x00	0x00	0x00

The revision number consists of a 4 Byte long code. It must be transmitted in a mirrored way. That means the last byte must be transmitted at first.

As a result:

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x42	0x**XX**	0x**XX**	0x**XX**	0x**XX**	0x00	0x00	0x00

4) Sending the serial number

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x43	Serial (LSB)	Serial	Serial	Serial (MSB)	0x00	0x00	0x00

The serial number consists of a 4 Byte long code. It must be transmitted in a mirrored way. That means the last byte must be transmitted at first.
The serial number of the SUCO 0630 transmitter is preset by default by the "Z**...**" number engraved on the SW 22 hexagonal surface. The number can be changed via object 0x4030. For customer support and questions, however, the *engraved* serial number remains valid.

As a result:

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x43	0x**XX**	0x**XX**	0x**XX**	0x**XX**	0x00	0x00	0x00

After sending all required identification objects, the LSS slave responds with the following message:

ID=0x7E4

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x44	0x00	0x00	0x00	0x00	0x00	0x00	0x00

All changes to be made, only affect this particular node.

8.3 Configure Node-ID Protocol

To configure the Node-ID, you must ensure that if you run this service in "Switch State Global" mode, you only run it in a 1:1 wiring configuration. Otherwise, multiple nodes can be configured in the same way and your network can become inconsistent.

Send the following CAN message to configure the Node-ID:

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x11	NID	0x00	0x00	0x00	0x00	0x00	0x00

In byte 1 the new node ID is set by the LSS master. This must be in the range between 1 and 127.

NID = 0x01 to 0x7F

The node responds with:

ID = 0x7E4

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x11	Error	Specific Error	0x00	0x00	0x00	0x00	0x00

Error: 0x00 – protocol successfully completed
0x01 – Node-ID out of range
0x02 to 0xFF – reserved, always 0

After the change, the new node ID is only stored volatilely.

If the node goes through a power cycle before receiving the "store" command, it is restarted with the old node ID.

For storing changes to the EEPROM, see chapter 8.5.

8.4 Configure Bit timing Parameters Protocol

To configure the Bit timing parameter, you must ensure that if you run this service in "Switch State Global" mode, you only run it in a 1:1 wiring configuration. Otherwise, multiple nodes can be configured in the same way and your network can become inconsistent.
After the change, the bit rate is only stored volatilely. If the node goes through a power cycle before receiving the "store" command, it is restarted with the old bit rate.
For storing changes to the EEPROM, see chapter 8.5.

To configure the bit rate, the LSS master must send the following CAN message:

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x13	Table Selector	Table Index	0x00	0x00	0x00	0x00	0x00

The Table Selector is always set to 0x00 for the use of the standard CiA table.
The options for the Table Index are as follows:

Table Index	Bit rate [kbit/s]	supported
0	1000	Yes
1	800	No, not recommended by CiA
2	500	Yes
3	250	Yes
4	125	Yes
5	100	Yes
6	50	Yes
7	20	Yes
8	10	Yes

The node responds with:

ID = 0x7E4

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x13	Error	Specific Error	0x00	0x00	0x00	0x00	0x00

Error: 0x00 – protocol successfully completed
0x01 – Bit rate not supported
0x02 to 0xFF – reserved, always 0

8.5 Store Configuration Parameters

The following command must be executed to ensure that all changes made in the LSS are permanently stored in the EEPROM and not only volatilely stored.

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x17	0x00	0x00	0x00	0x00	0x00	0x00	0x00

The node responds with:

ID = 0x7E4

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x17	Error	Specific Error	0x00	0x00	0x00	0x00	0x00

Error: 0x00 – Protocol successfully completed
0x01 – store configuration not supported
0x02 – storage media access error
0x03 to 0xFF – reserved, always 0

Once all changes have been successfully saved, they are not yet activated. There are two possible ways to do this:

- Switching the power supply on and off.
- A soft reboot of the device with the NMT command "Node Reset".

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

8.6 Inquiry Protocols

To get information about the current node ID and the identification objects of a device, these can be requested via this protocol of the LSS.
The following requests may only be executed if only one LSS Slave is in configuration mode.

ID = 0x7E5

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
cs	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Command specifier (cs):

- 0x5A – Inquire Vendor-ID
- 0x5B – Inquire Product Code
- 0x5C – Inquire Revision Number
- 0x5D – Inquire Serial Number
- 0x5E – Inquire Node-ID

The node responds accordingly:

ID = 0x7E4

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
cs	Number – Identity Object				0x00	0x00	0x00

Respond for the identification object inquiry.

ID = 0x7E4

Data							
Byte 0							Byte 7
0	1	2	3	4	5	6	7
0x5E	Node-ID	0x00	0x00	0x00	0x00	0x00	0x00

Respond for the Node-ID inquiry.

9 Object Dictionary

This chapter describes the entries in the object dictionary. The entries are defined either by the CiA in the communication profile (DS 301) and in the device profile (DS 404) or by the manufacturer himself.

Communication Profile – DS 301

Object index	Sub-index	Description	Data Type	Access Type
Device Type				
0x1000	0x00	default value: 0x00020194	Uint32	RO
Error Register				
0x1001	0x00	initial value: 0x00	Uint8	RO
Manufacturer Status Register				
0x1002	0x00		Uint32	RO
Pre-Defined Error Field				
0x1003	0x00	number of errors [0 - 254]	Uint32	RW
	0x01 – 0xFE	detailed info: chapter 4.4.	Uint32	RO
COB-ID SYNC				
0x1005	0x00	default CAN-ID: 0x080	Uint32	RW
Manufacturer Device Name				
0x1008	0x00	default value: “0630-CANopen”	Visible_String	CONST
Manufacturer Hardware Version				
0x1009	0x00	default value: “vYEARWEEK\0”	Visible_String	CONST
Manufacturer Software Version				
0x100A	0x00	default value: “vYEARWEEK\0”	Visible_String	CONST
Store Parameters				
0x1010	0x00	number of entries [1]	Uint8	CONST
	0x01	save all parameters	Uint32	RW
Restore Default Parameters				
0x1011	0x00	number of entries [1]	Uint8	CONST
	0x01	reset all parameters to factory settings, except bit rate and node-ID	Uint32	RW
COB-ID Time				
0x1012	0x00	default value: 0x80000100 (disable)	Uint32	RW
COB-ID EMCY				
0x1014	0x00	default CAN-ID: 0x080 + node-ID	Uint32	RW
Consumer Heartbeat Time				
0x1016	0x00	number of entries [1]	Uint8	CONST
	0x01	default: disabled	Uint32	RW
Producer Heartbeat Time				
0x1017	0x00	default: disabled – info: chapter 4.1.3.	Uint16	RW
Identity Object				
0x1018	0x00	number of entries [4]	Uint8	CONST
	0x01	Vendor-ID	Uint32	RO
	0x02	Product Code	Uint32	RO
	0x03	Revision Number	Uint32	RO
	0x04	Serial Number	Uint32	RO

Verify Configuration				
0x1020	0x00	number of entries [2]	Uint8	CONST
	0x01	number of days since January 1, 1984	Uint32	RW
	0x02	number of milliseconds after midnight	Uint32	RW
Error Behavior				
0x1029	0x00	number of entries [3]	Uint8	CONST
	0x01	Communication error	Uint8	RW
	0x02	n.a.	Uint8	RW
	0x03	analog input error	Uint8	RW
Transmit PDO1 Communication Parameter				
0x1800	0x00	number of entries [5]	Unit8	CONST
	0x01	COB-ID PDO1: 0x180 + node-ID (enable)	Uint32	RW
	0x02	Transmission Type	Uint8	RW
	0x03	n.a.	-	-
	0x04	n.a.	-	-
	0x05	Event Timer	Uint16	RW
Transmit PDO2 Communication Parameter				
0x1801	0x00	number of entries [5]	Unit8	CONST
	0x01	COB-ID PDO2: 0x280 + node-ID (disable)	Uint32	RW
	0x02	Transmission Type	Uint8	RW
	0x03	n.a.	-	-
	0x04	n.a.	-	-
	0x05	Event Timer	Uint16	RW
Transmit PDO1 Mapping Parameter				
0x1A00	0x00	number of mapped entries [1 – 8]	Unit8	RW
	0x01 – 0x08	mapping entry of the n^{th} object	Uint32	RW
Transmit PDO2 Mapping Parameter				
0x1A01	0x00	number of mapped entries [1 – 8]	Unit8	RW
	0x01 – 0x08	mapping entry of the n^{th} object	Uint32	RW
NMT-Startup				
0x1F80	0x00	default: 0x00 – info: chapter 4.1.2.	Uint32	RW

Device Profile – DS 404

Object index	Sub-index	Description	Data Type	Access Type
Analog Input Sensor Type				
0x6110	0x00	number of entries [2]	Uint8	CONST
	0x01	sensor type channel 1: 0x5A (CiA-DS404)	Uint16	CONST
	0x02	sensor type channel 2: 0x64 (CiA-DS404)	Uint16	CONST
Analog Input ADC Sample Rate				
0x6114	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – default: 0x3E8	Uint32	CONST
	0x02	measuring channel 2 – default: 0x3E8	Uint32	CONST
Analog Input Scaling 1 Process Value				
0x6121	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – 1st scaling point	Float32	RW
	0x02	measuring channel 2 – 1st scaling point	Float32	RW
Analog Input Scaling 2 Process Value				
0x6123	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – 2nd scaling point	Float32	RW
	0x02	measuring channel 2 – 2nd scaling point	Float32	RW
Analog Input Scaling Factor				
0x6126	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – scaling factor PV	Float32	RW
	0x02	measuring channel 2 – scaling factor PV	Float32	RW
Analog Input Scaling Offset				
0x6127	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – scaling offset PV	Float32	RW
	0x02	measuring channel 2 – scaling offset PV	Float32	RW
Analog Input Process Value				
0x6130	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – PV	Float32	RO
	0x02	measuring channel 2 – PV	Float32	RO
Analog Input Physical Unit Process Value				
0x6131	0x00	number of entries [2]	Uint8	CONST
	0x01	default: 0x004E0000 – (bar)	Uint32	RW
	0x02	default: 0x002D0000 – (°C)	Uint32	RW
Analog Input Decimal Digits Process Value				
0x6132	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – decimal points	Uint8	RW
	0x02	measuring channel 2 – decimal points	Uint8	RW
Analog Input Interrupt Delta Input Process Value				
0x6133	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – default value: 0x00 (disabled)	Float32	RW
	0x02	measuring channel 2 – default value: 0x00 (disabled)	Float32	RW

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

Analog Input Interrupt Lower Limit Input Process Value				
0x6134	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – default value: 0xFF7FFFFF (disabled)	Float32	RW
	0x02	measuring channel 2 – default value: 0xFF7FFFFF (disabled)	Float32	RW
Analog Input Interrupt Upper Limit Input Process Value				
0x6135	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – default value: 0x7F7FFFFF (disabled)	Float32	RW
	0x02	measuring channel 2 – default value: 0xFF7FFFFF (disabled)	Float32	RW
Analog Input Status				
0x6150	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – internal status	Uint8	RO
	0x02	measuring channel 1 – internal status	Unt8	RO
Field Value				
Ox7100	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – FV	Int16	RO
	0x02	measuring channel 2 – FV	Int16	RO
Analog Input Scaling 1 Field Value				
0x7120	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – 1st scaling point	Int16	RO
	0x02	measuring channel 2 – 1st scaling point	Int16	RO
Analog Input Scaling 2 Field Value				
0x7122	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – 2nd scaling point	Int16	RO
	0x02	measuring channel 2 – 2nd scaling point	Int16	RO
Analog Input Process Value				
0x7130	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – PV	Int16	RO
	0x02	measuring channel 2 – PV	Int16	RO
Analog Input Scaling 1 Process Value				
0x9121	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – 1st scaling point	Float32	RW
	0x02	measuring channel 2 – 1st scaling point	Float32	RW
Analog Input Scaling 2 Process Value				
0x9123	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – 2nd scaling point	Int32	RW
	0x02	measuring channel 2 – 2nd scaling point	Int32	RW
Analog Input Process Value				
0x9130	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – PV	Int32	RO
	0x02	measuring channel 2 – PV	Int32	RO

Manufacturer Specific Profile

Object index	Sub-index	Description	Data Type	Access Type
Field Value Integer Offset				
0x3124	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – offset FV	Int16	RW
	0x02	measuring channel 2 – offset FV	Int16	RW
Field Value Float Offset				
0x3125	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – offset FV	Float32	RW
	0x02	measuring channel 2 – offset FV	Float32	RW
Autozero				
0x3126	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – info: chapter 6.X.	Uint32	WO
	0x02	measuring channel 2 – info: chapter 6.X.	Uint32	WO
Extreme Values - Operating				
0x3130	0x00	number of entries [4]	Uint8	CONST
	0x01	measuring channel 1 – global minimum	Float32	RO
	0x02	measuring channel 1 – global maximum	Float32	RO
	0x03	measuring channel 2 – global minimum	Float32	RO
	0x04	measuring channel 2 – global maximum	Float32	RO
Extreme Values - Service Life				
0x3131	0x00	number of entries [4]	Uint8	CONST
	0x01	measuring channel 1 – global minimum	Float32	RO
	0x02	measuring channel 1 – global maximum	Float32	RO
	0x03	measuring channel 2 – global minimum	Float32	RO
	0x04	measuring channel 2 – global maximum	Float32	RO
Time Stamp – Days				
0x3140	0x00	corresponds to a TIME producer	Uint16	RO
Time Stamp – Milliseconds				
0x3141	0x00	corresponds to a TIME producer	Uint32	RO
Process Value Limit Indicator				
0x3200	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – 0x01 (enabled)	Uint8	RW
	0x02	measuring channel 2 – 0x00 (disabled)	Uint8	RW
Lower Process Value Limit				
0x3201	0x00	number of entries [2]mar	Uint8	CONST
	0x01	measuring channel 1 – 0x00	Float32	RW
	0x02	measuring channel 2 – 0xFF7FFFFF	Float32	RW
Upper Process Value Limit				
0x3202	0x00	number of entries [2]	Uint8	CONST
	0x01	measuring channel 1 – 0x7F7FFFFF	Float32	RW
	0x02	measuring channel 2 – 0x7F7FFFFF	Float32	RW
Endianness – Measured Values				
0x4000	0x00	Default value: 0x00 (Little Endian)	Uint8	RW
Configure Node-ID				
0x4001	0x00	number of entries [2]	Uint8	CONST
	0x01	read/set Node-ID – no COB-ID changes	Unit8	RW
	0x02	read/set Node-ID – COB-ID changes	Uint8	RW

SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de

Hardware Settings				
0x4010	0x00	number of entries [3]	Uint8	CONST
	0x01	indicates if a switchable termination is present	Boolean	CONST
	0x02	enable/disable termination	Boolean	RW
	0x03	Indicates if a galvanic bus isolation is present	Boolean	CONST
Comment String				
0x3020	0x00	number of available entries [80]	Uint8	CONST
	0x01 – 0x4F	string symbols	Visible_String	RW
	0x50	0x00 (null-termination symbol)	Visible_String	RO
Service String				
0x4021	0x00	number of available entries [60]	Uint8	CONST
	0x01 – 0x3B	string symbols	Visible_String	RW
	0x3C	0x00 (null-termination symbol)	Visible_String	RO
Device Name				
0x4022	0x00	number of available entries [60]	Uint8	CONST
	0x01 – 0x3B	string symbols	Visible_String	RW
	0x3C	0x00 (null-termination symbol)	Visible_String	RO
Configure Serial Number				
0x4030	0x00	number of entries [2]	Uint8	CONST
	0x01	lower 32 bits (should be written first)	Uint32	RW
	0x02	upper 32 bits	Uint32	RW
Service Time				
0x4031	0x00	number of entries [2]	Uint8	CONST
	0x01	lower 32 bits (should be written first)	Uint32	RW
	0x02	upper 32 bits	Uint32	RW

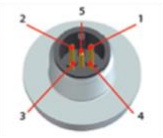
10 Electrical Characteristics and Specifications

10.1 Operation Conditions

Symbol	Parameter	Conditions	Min	Type	Max	Unit
V_{CC}	Supply Voltage	Without galv. isolation	9[1)]	32	35	V
I_{CC}	Supply Current	V_{CC} = 9 V[1)]	12		14	mA
		V_{CC} = 32 V	3		5	mA
T_{op}	Operating temperature		-40		125	°C
		If Switchable Termination is included	-40		105	°C
T_{amb}	Ambient temperature		-40		105	°C
T_{comp}	Compensated temperature range		-20		85	°C
$V_{CAN\_H/L}$	Voltage on CANH / CANL	Related to GND	-32		32	V

The device is protected against miswiring at voltages up to 32 volts

10.2 Electrical Connector and Pin Assignment

M12 DIN EN 61076–2–101 A CiA–DR303–1

1: NC
2: U_{V+}
3: GND
4: CAN–High
5: CAN–Low
x ~ 60 mm
d ~ Ø 22 mm

10.3 Protocol Specifications

The 0630 operates according to the following CANopen CiA Draft Standards:

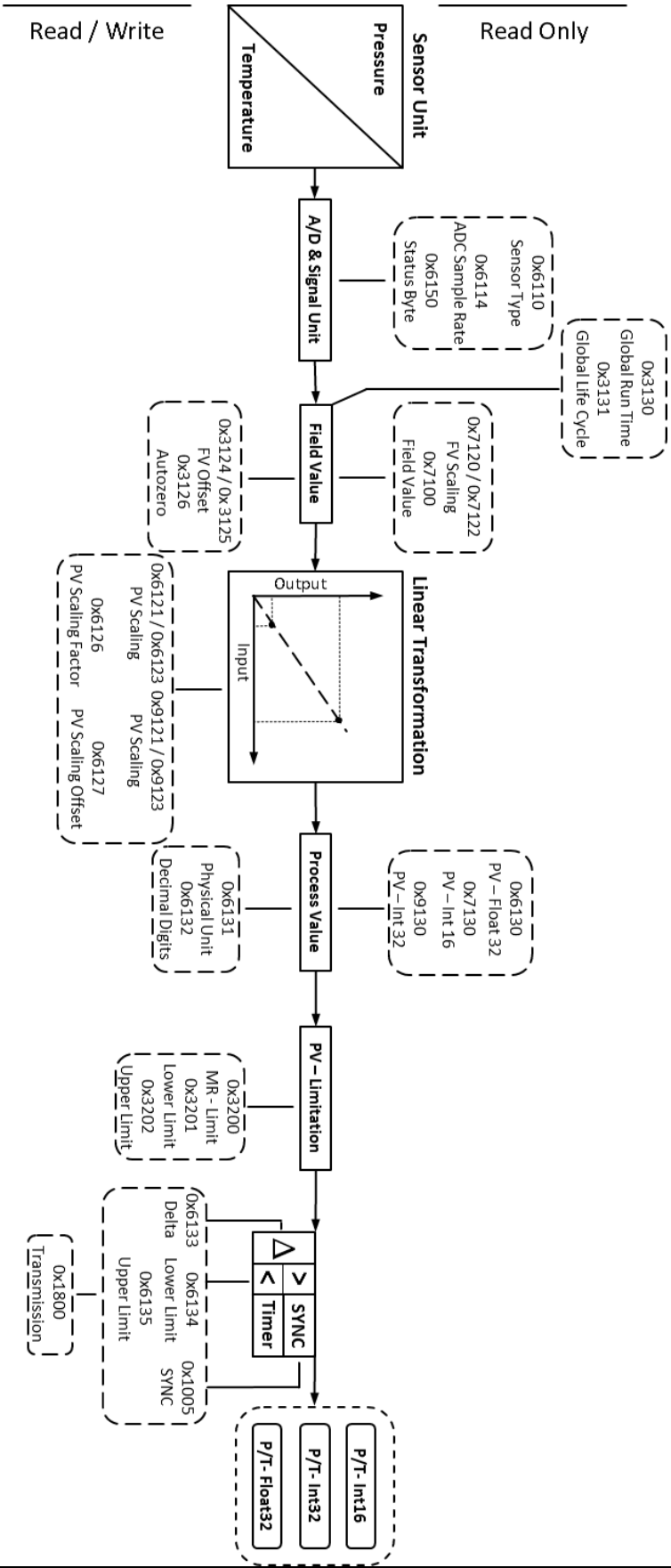
- DS – 404 Device profile for measuring devices and closed-loop controllers
- DS – 301 Application layer and communication profile
- DS – 302 Additional application layer functions
- DR – 303-1 Recommendation – Part 1: Cabling and Connector Pin Assignment
- DR – 303-2 Recommendation – Part 2: Representation of SI units and prefixes
- DS – 305 Layer setting services (LSS) and protocols

1) Absolute Minimum - When designing the application network, consider the wiring resistances

11 Abbreviation Index

#: symbol for the word “number”
CAN: Controller Area Network
CiA®: CAN in Automation
COB: Communication Object
COB-ID: Communication Object Identifier.
DLC: Data Length Code
DS: Draft Specification
LSS: Layer Setting Service.
PDO: Process Data Object.
TPDO: Transmit PDO
SDO: Service Data Object.
MSB: Most Significant Bit
LSB: Least Significant Bit
HBP: Heartbeat Protocol
SYNC: Synchronization
EMCY: Emergency
NMT: Network Management

Appendix



SUCO Robert Scheuffele GmbH & Co.KG, Keplerstraße 12-14, 74321 Bietigheim-Bissingen, Tel.:+49-7142-597-0, www.suco.de, info@suco.de