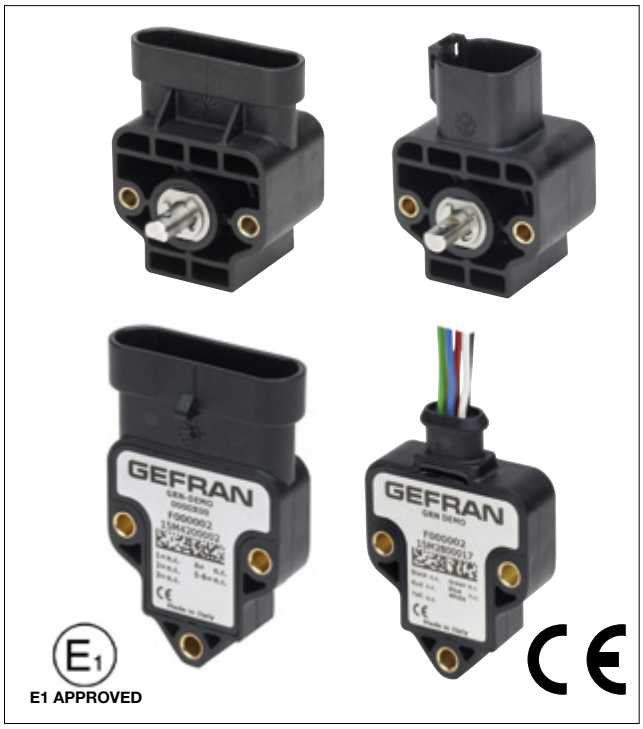
GEFRAN

CANopen GRA-GRN

digital output



Code 85199C **Edition** 06-2020

SUMMARY

1. INTRODUCTION

The GRA/GRN contactless angle sensor (in HALL technology) implements the functions of a CAN BUS network slave device conforming to standard CANopen protocol proposed by C.i.A. (Can in Automation) and described in the document entitled “CANOpen Application Layer and Communication Profile DS 301 v. 4.2” and in other documents mentioned below.

Other reference documents used are C.i.A. DS-406 Device Profile for Encoders V3.1 (not completely implemented) and C.i.A. DSP-305 Layer Setting Services and Protocol V1.1.1.

This document describes the standard CANopen implementations created. It is addressed to CANopen network system integrators and to CANopen device designers who already know the content of the above-mentioned standards defined by C.i.A..

The details of aspects defined by CANopen do not pertain to the purpose of this text. For further information on the protocol you can also contact us via e-mail: at http://www.gefran.com/it/it/messages/new or contact the GEFRAN office nearest to you.

Definition and Shortening

CAN: Controller Area Network.
Describes a serial communication bus that implements the “physical” level 1 and the “data link” level 2 of the ISO/OSI reference model.

CAL: CAN Application Layer.
Describes implementation of the CAN in the level 7 “application” of the ISO/OSI reference model from which CANopen derives.

CMS: CAN Message Specification.
CAL service element. Defines the CAN Application Layer for the various industrial applications.

COB: Communication Object.
Unit of transport of data in a CAN network (a CAN message). A maximum of 2048 COBs may be present in a CAN network, each of which may transport from 0 to a maximum of 8 bytes.

COB-ID: COB Identifier.
Identifying element of a CAN message. The identifier determines the priority of a COB in case of multiple messages in the network.

D1 – D8: Data from 1 to 8.
Number of bytes in the data field of a CAN message.

DLC: Data Length code.
Number of data bytes transmitted in a single frame.

ISO: International Standard Organization.
International authority providing standards for various merchandise sectors.

NMT: Network Management.
CAL service element. Describes how to configure, initialize, manage errors in a CAN network.

PDO: Process Data Object.
Process data communication objects (with high priority).

RXSDO: Receive SDO.
SDO objects received from the remote device.

SDO: Service Data Object.
Service data communication objects (with low priority). The value of this data is contained in the “Objects Dictionary” of each device in the CAN network.

TXPDO: Transmit PDO.
PDO objects transmitted by the remote device.

TXSDO: Transmit SDO.
SDO objects transmitted by the remote device.

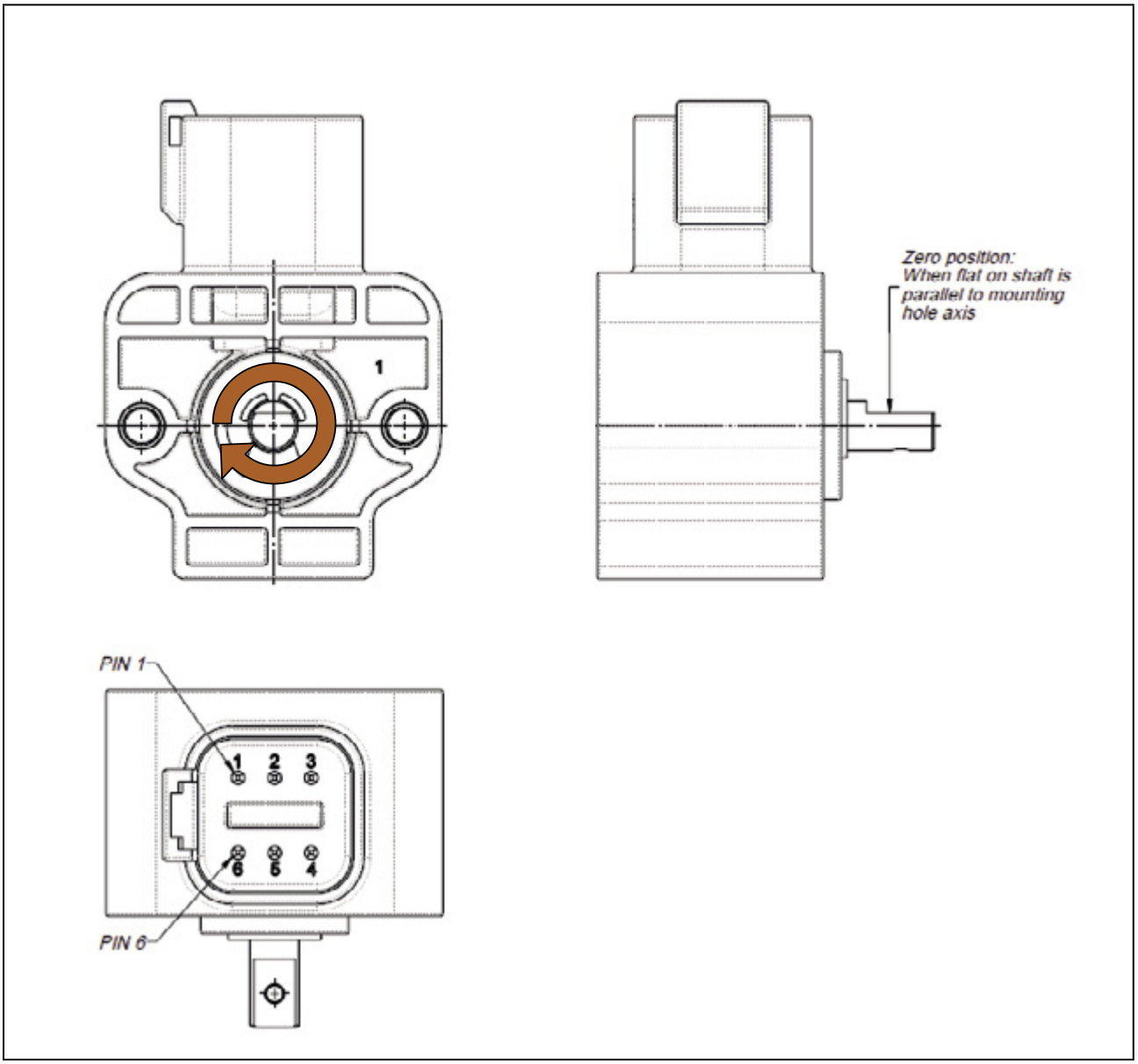
N.B.: The numbers followed by the suffix “h” represent a hexadecimal value, with suffix “b” a binary value, and with suffix “d” a decimal value. The value is decimal unless specified otherwise.

2. ELECTRICAL CONNECTIONS

For the connections refer to the table below:

DEUTSCH DT04-6P	Meaning
1	OV (GND)
2	+Vs (+9 … +36 Vdc)
3	NC
4	NC
5	CAN-L
6	CAN-H

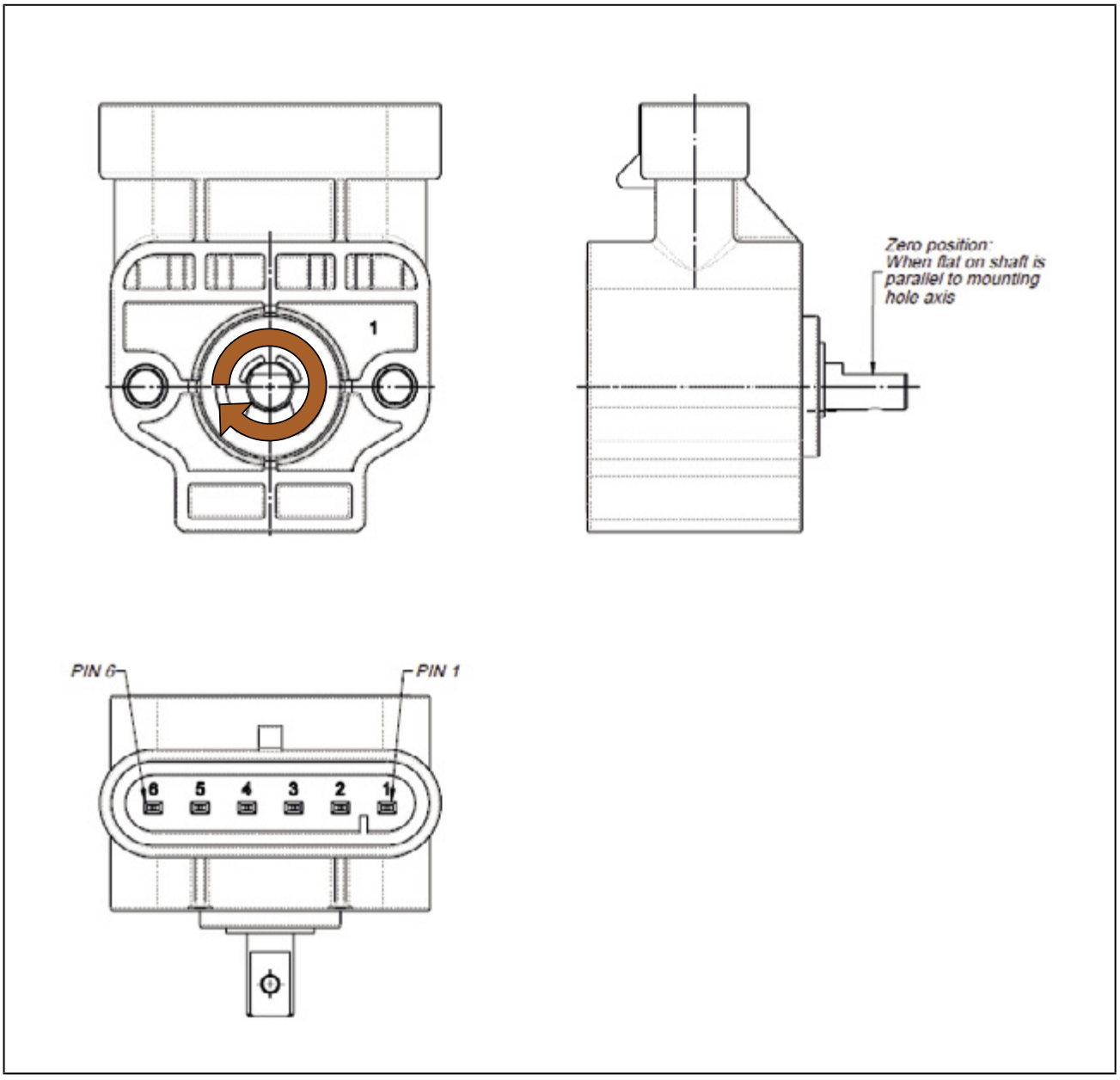
Note: please make sure that the CANbus is terminated.
The impedance measured between CAN H and CAN L must be 60 ohm that means the cable must be connected to a 120 ohm resistor on each ends of the bus line. Internally the transducer is not terminated with the resistor of 120 ohm.
Do not confuse the signal lines of the CAN bus, otherwise communication with the transducer is impossible.



For the connections refer to the table below:

AMP Superseal 6 P 282108-1	Meaning
1	OV (GND)
2	+Vs (+9 … +36 Vdc)
3	NC
4	NC
5	CAN-L
6	CAN-H

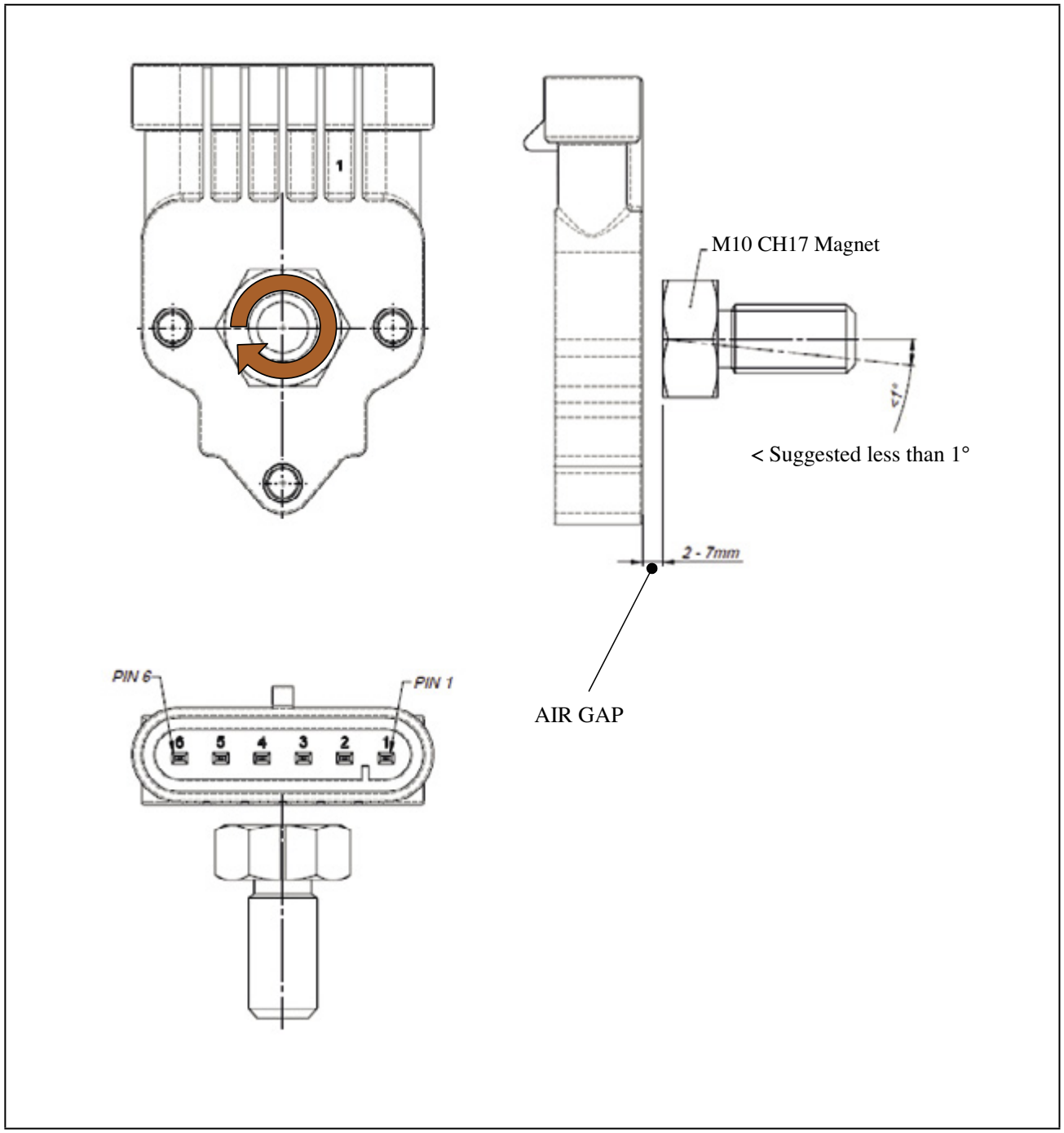
Note: please make sure that the CANbus is terminated.
The impedance measured between CAN H and CAN L must be 60 ohm that means the cable must be connected to a 120 ohm resistor on each ends of the bus line. Internally the transducer is not terminated with the resistor of 120 ohm.
Do not confuse the signal lines of the CAN bus, otherwise communication with the transducer is impossible.



For the connections refer to the table below:

AMP Superseal 6 P 282108-1	Meaning
1	OV (GND)
2	+Vs (+9 … +36 Vdc)
3	NC
4	NC
5	CAN-L
6	CAN-H

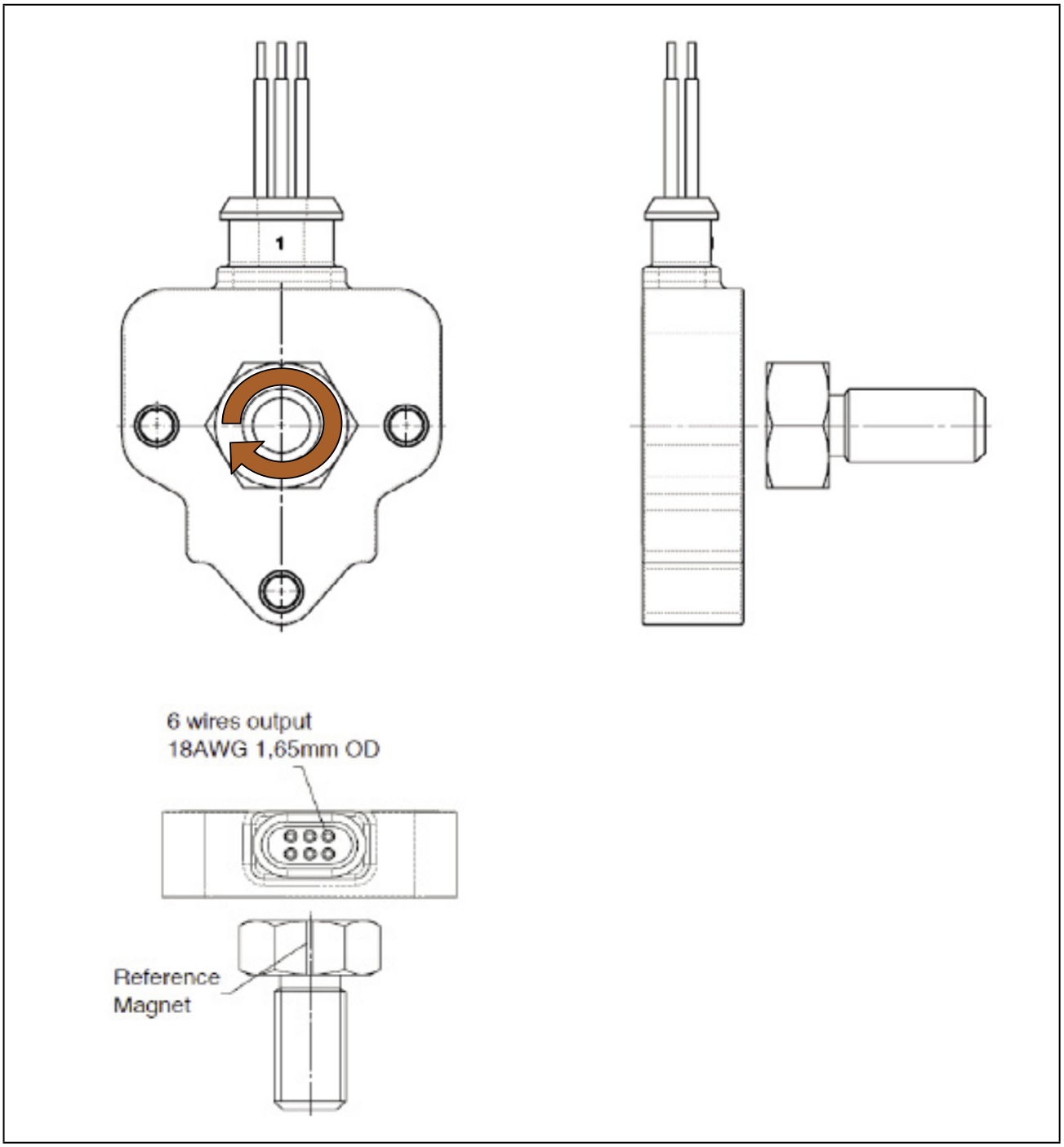
Note: please make sure that the CANbus is terminated.
The impedance measured between CAN H and CAN L must be 60 ohm that means the cable must be connected to a 120 ohm resistor on each ends of the bus line. Internally the transducer is not terminated with the resistor of 120 ohm.
Do not confuse the signal lines of the CAN bus, otherwise communication with the transducer is impossible.



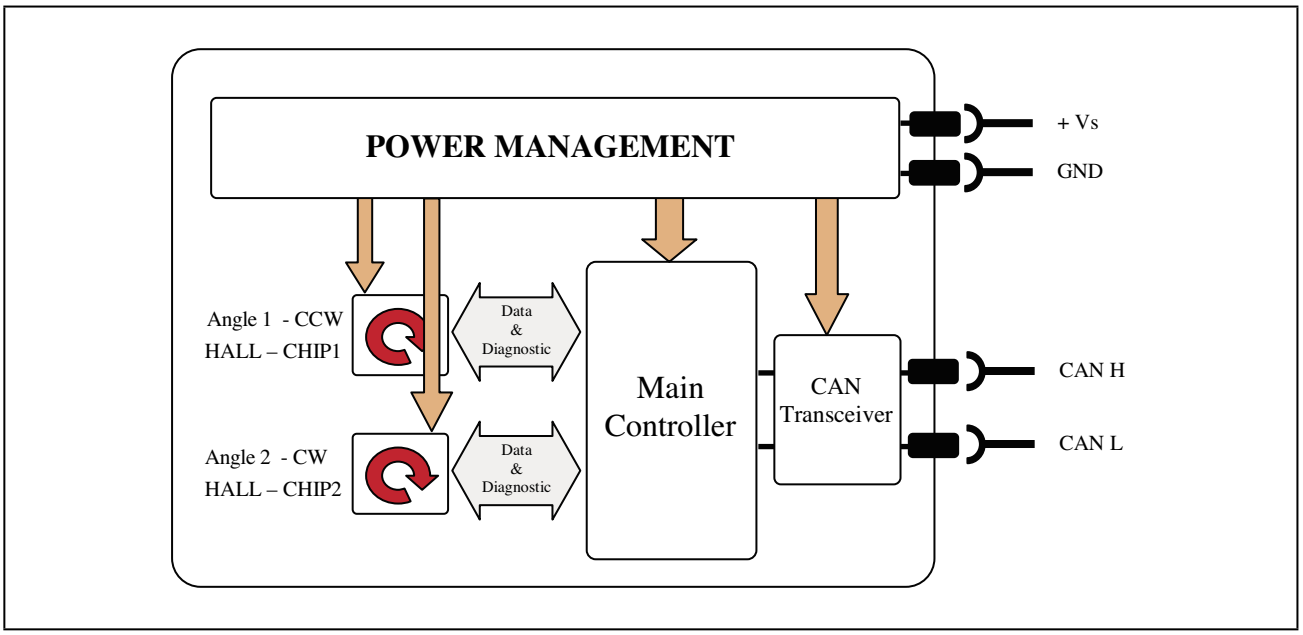
For the connections refer to the table below:

6 wires output 18AWG 1,65mm OD	Meaning
BLACK	GROUND
RED	+ SUPPLY 1
YELLOW	NC
GREEN	NC
BLUE	CAN-L
WHITE	CAN-H

Note: please make sure that the CANbus is terminated.
The impedance measured between CAN H and CAN L must be 60 ohm that means the cable must be connected to a 120 ohm resistor on each ends of the bus line. Internally the transducer is not terminated with the resistor of 120 ohm.
Do not confuse the signal lines of the CAN bus, otherwise communication with the transducer is impossible.

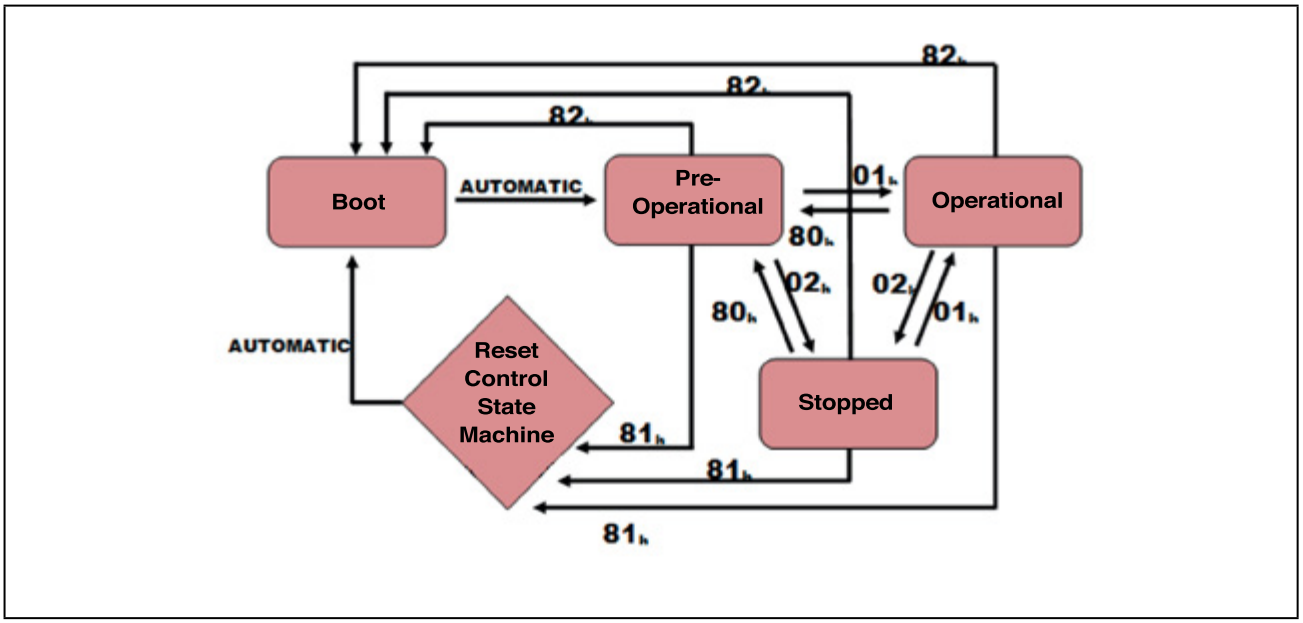


Functional Block:



3. NETWORK MANAGEMENT (NMT)

The device supports CANopen network management functionality NMT Slave (Minimum Boot Up).



Every CANopen device contains an internal Network Management server that communicates with an external NMT master. One device in a network, generally the host, may act as the NMT master.

Through NMT messages, each CANopen device's network management server controls state changes within its built-in **Communication State Machine**.

This is independent from each node's operational state machine, which is device dependant and described in **Control State Machine**.

It is important to distinguish a CANopen device's operational state machine from its Communication State Machine.

CANopen sensors and I/O modules, for example, have completely different operational state machines than servo drives.

The "**Communication State Machine**" in all CANopen devices, however, is identical as specified by the DS301.

NMT messages have the highest priority. The 5 NMT messages that control the Communication State Machine each contain 2 data bytes that identify the node number and a command to that node's state machine.

Table 1 shows the 5 NMT messages supported, and **Table 2** shows the correct message construction for sending these messages.

Table 1

NMT Message	COB-ID	Data Byte 1	Data Bytes 2
Start Remote Node	0	01h	Node-ID*
Stop Remote Node	0	02h	Node-ID*
Pre-operational State	0	80h	Node-ID*
Reset Node	0	81h	Node-ID*
Reset Communication	0	82h	Node-ID*
* Node-ID = Drive address (from 1 to 7Fh)			

Table 2

Arbitration Field	Data Field								
COB-ID	RTR	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
000h	0	See table 1	See table 2	These bytes not sent					

4. BAUD RATE

Baud rate can be configurable via Layer Setting Services and Protocol (LSS) and via SDO communication (index 0x5999). This parameters are called LSS parameters (marking LSS-PARA).

The default Baud rate is 250kbit/s.

Important Note:
Changing this parameter can disturb the network! Use this service only if one device is connected to the network!

5. Node-ID and resolution

Node-ID can be configurable via Layer Setting Services and Protocol (LSS) and via SDO communication (index 0x5999). This parameters are called LSS parameters (marking LSS-PARA).
The resolution can be configurable by using manufacturing specific object 0x2100

The default Node-ID is 7F.
The default resolution is 0.1°

Important Note:
Changing this parameter can disturb the network! Use this service only if one device is connected to the network!

6. PARAMETER SETTINGS

All object dictionary parameters (objects with marking PARA) can be saved in a special section of the internal EEPROM and secured by checksum calculation.
The special LSS parameters (objects with marking LSS-PARA), also part of the object dictionary, will be also saved in a special section of the internal EEPROM and secured by checksum calculation.
Due to the internal architecture of the microcontroller the parameter write cycles are limited to 100,000 cycles.

7. RESTORE DEFAULT PARAMETERS

All object dictionary parameters (objects with marking PARA) can be restored to factory default values via SDO communication (index 0x1011).

8. HEARTBEAT

The heartbeat mechanism for this device is established through cyclic transmission of the heartbeat message done by the heartbeat producer.
One or more devices in the network are aware of this heartbeat message. If the heartbeat cycle fails from the heartbeat producer the local application on the heartbeat consumer will be informed about that event.
The implementation of either guarding or heartbeat is mandatory.
The device supports Heartbeat Producer functionality.
The producer heartbeat time is defined in object 0x1017.

Heartbeat Message

COB-ID	Byte	0
700+Nodo-ID	Content	NMT State

9. ERROR HANDLING

Principle

Emergency messages (EMCY) shall be triggered by internal errors on device and they are assigned the highest possible priority to ensure that they get access to the bus without delay (EMCY Producer). By default, the EMCY contains the error field with pre-defined error numbers and additional information.

Error Behavior (object 0x4000)

If a serious device failure is detected the object 0x4000 specifies, to which state the module shall be set:

0: pre-operational
1: no state change (default)
2: stopped

EMCY Message

The EMCY COB-ID is defined in object 0x1014. The EMCY message consists of 8 bytes. It contains an emergency error code, the contents of object 0x1001 and 5 byte of manufacturer specific error code. This device uses only the 1st byte as manufacturer specific error code.

Byte	**Byte1** **Byte2**	**Byte3**	**Byte4**	**Byte5**	**Byte6** **Byte7** **Byte8**
Description	Error Code [1)]	Error Register (object 0x1001[2)])	Manufacturer specific error code (object 0x4001)	Manufacturer specific error code (always 0x00)	Manufacturer specific error code NOT IMPLEMENTED (always 0xFF)

[1)] 0x1000 as Generic Error
[2)] Always 0

Supported Manufacturer Specific Error Codes (object 0x4001)

Manufacturer Specific Error Code (bit field)	**Description**
0x01	Angle 1 sensor chip1 internal error
0x02	Angle 2 sensor chip2 internal error
0x04	Angle mismatch (Angle 1 vs Angle 2) error, object 0x2103 NOT IMPLEMENTED
0x10	Program checksum error
0x40	LSS Parameter checksum error
0x83	Magnetic field too large or Magnetic field too low

10. SDO COMMUNICATION

The device fulfils the SDO Server functionality.
With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.
Structure of SDO-request by the Master

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

Structure of SDO-answer by the Slave

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580+Node-ID	8	RES	Index		Sub-Index	Data	Data	Data	Data

Write Access, Data Transfer from Host to Slave

Each access to the object dictionary is checked by the slave for validity. Any write access to nonexistent objects, to read-only objects or with a non-corresponding data format are rejected and answered with a corresponding error message.

CMD determines the direction of data transfer and the size of the data object:
23 hex Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex Sending of 1-byte data (byte 5 contains an 8-bit value)

The slave answers:
RES Response of the slave:
60 hex Data sent successfully
80 hex Error,

Read Access, Data Transfer from Slave to Host

Any read access to non-existing objects is answered with an error message.
CMD determines the direction of data transfer:
40 hex read access (in any case)
The slave answers:
RES Response of the slave:
42 hex Bytes used by node when replying to read command with 4 or less data
43 hex Bytes 5...8 contain a 32-bit value
4B hex Bytes 5, 6 contain a 16-bit value
4F hex Byte 5 contains an 8-bit value
80 hex Error,

11. PDO COMMUNICATION and Angle Calculation

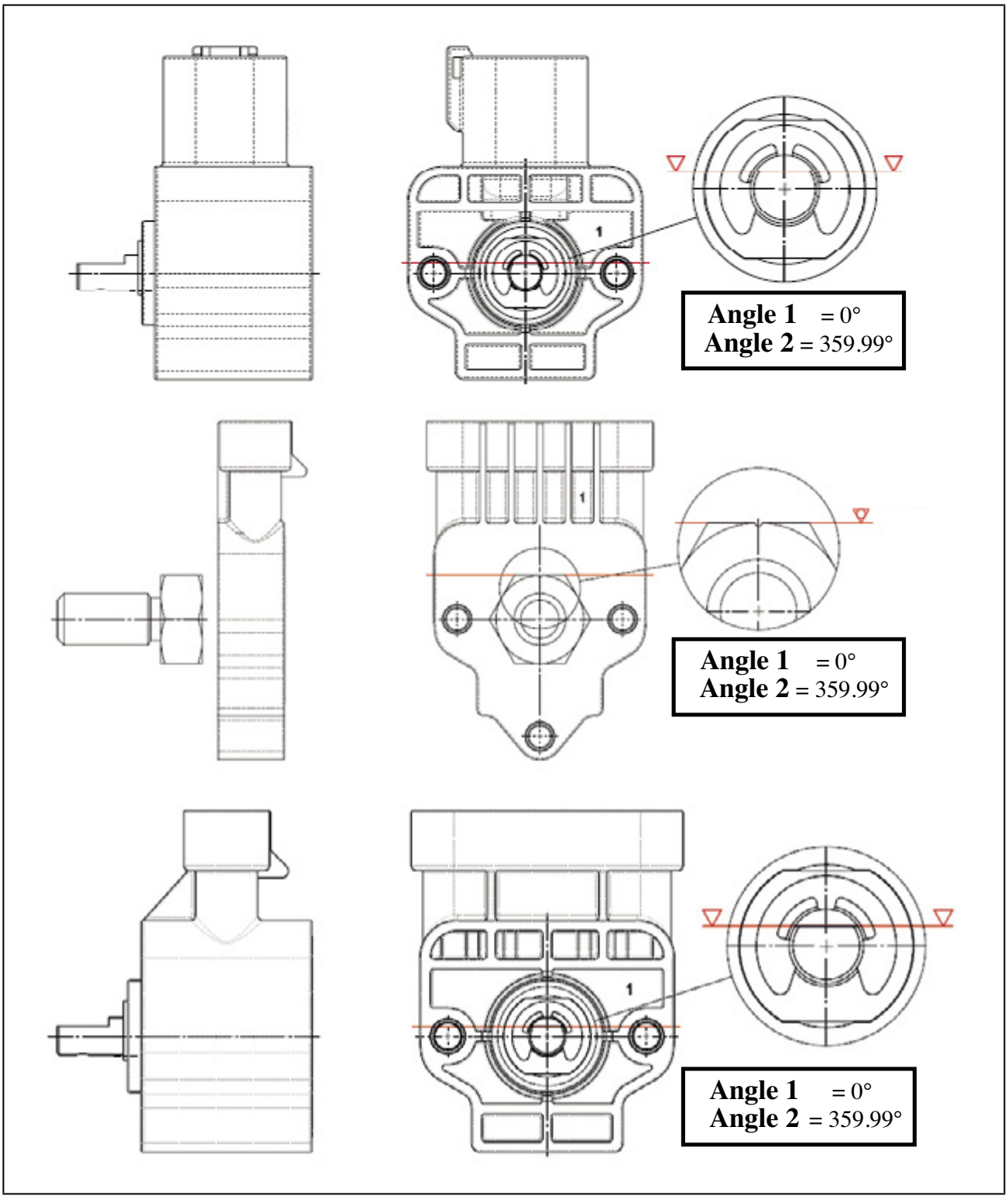
Transmit PDO #0

This PDO transmits asynchronously the position value of the angle sensor. The Tx PDO #0 shall be transmitted cyclically, if the cyclic timer (object 0x1800.5) is programmed > 0. Values between 1ms and 65535 ms shall be selectable by parameter settings. The Tx PDO #0 will be transmitted by entering the "Operational" state.

Byte	**Byte1**	**Byte2**	**Byte3**	**Byte4**	**Byte5 Byte6 Byte7**	**Byte8**
Description	ANGLE 1 object (0x2110.1) High-Byte	ANGLE 1 object (0x2110.2) Low-Byte	ANGLE 2 object (0x2110.3) High-Byte	ANGLE 2 object (0x2110.4) Low-Byte	(0xFF)	Error Code (object 0x4001)
Tx PDO #0 with default mapping when object 0x5001 = 0 (big endian)						

Byte	**Byte1**	**Byte2**	**Byte3**	**Byte4**	**Byte5 Byte6 Byte7**	**Byte8**
Description	ANGLE 1 object (0x2110.1) Low-Byte	ANGLE 1 object (0x2110.2) High-Byte	ANGLE 2 object (0x2110.3) Low-Byte	ANGLE 2 object (0x2110.4) High-Byte	(0xFF)	Error Code (object 0x4001)
Tx PDO #0 with default mapping when object 0x5001 = 1 (little endian)						

In the page an example of PDO mapping is reported in the case of Angle 1 = 0.00° and Angle 2 = 359.99°.



READING RESOLUTION +/- 0.1° (see manufacturing specific object 0x2100 and example (7) at the end of this manual)

Example of PDO mapping for

Angle 1 = 0.0° and Angle 2 = 359.9° (Node-ID = 02h, resolution 0.1°, zero degree point = 0.0°, CCW and big endian)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
182h	00h	00h	0Eh	0Fh	FFh	FFh	FFh	00h

Angle 1:

Byte 1 MSB (00h) = 00h Byte 2 LSB (00h) = 00h

Angle 1 = 0000h to decimal 0d (resolution 0.1°) = 0.0°

Angle 2:

Byte 3 MSB (00h) = 0Eh Byte 4 LSB (00h) = 0Fh

Angle 2 = 0E0Fh to decimal 3599d (resolution 0.1°) = 359.9°

READING RESOLUTION +/- 0.01° (see manufacturing specific object 0x2100 and example (8) at the end of this manual)

Example of PDO mapping for:

Angle 1 = 0.0° and Angle 2 = 359.9° (Node-ID = 02h, resolution 0.1°, zero degree point = 0.0°, CCW and big endian)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
182h	00h	00h	8Ch	9Fhh	FFh	FFh	FFh	00h

Angle 1:

Byte 1 MSB (00h) = 00h Byte 2 LSB (00h) = 00h

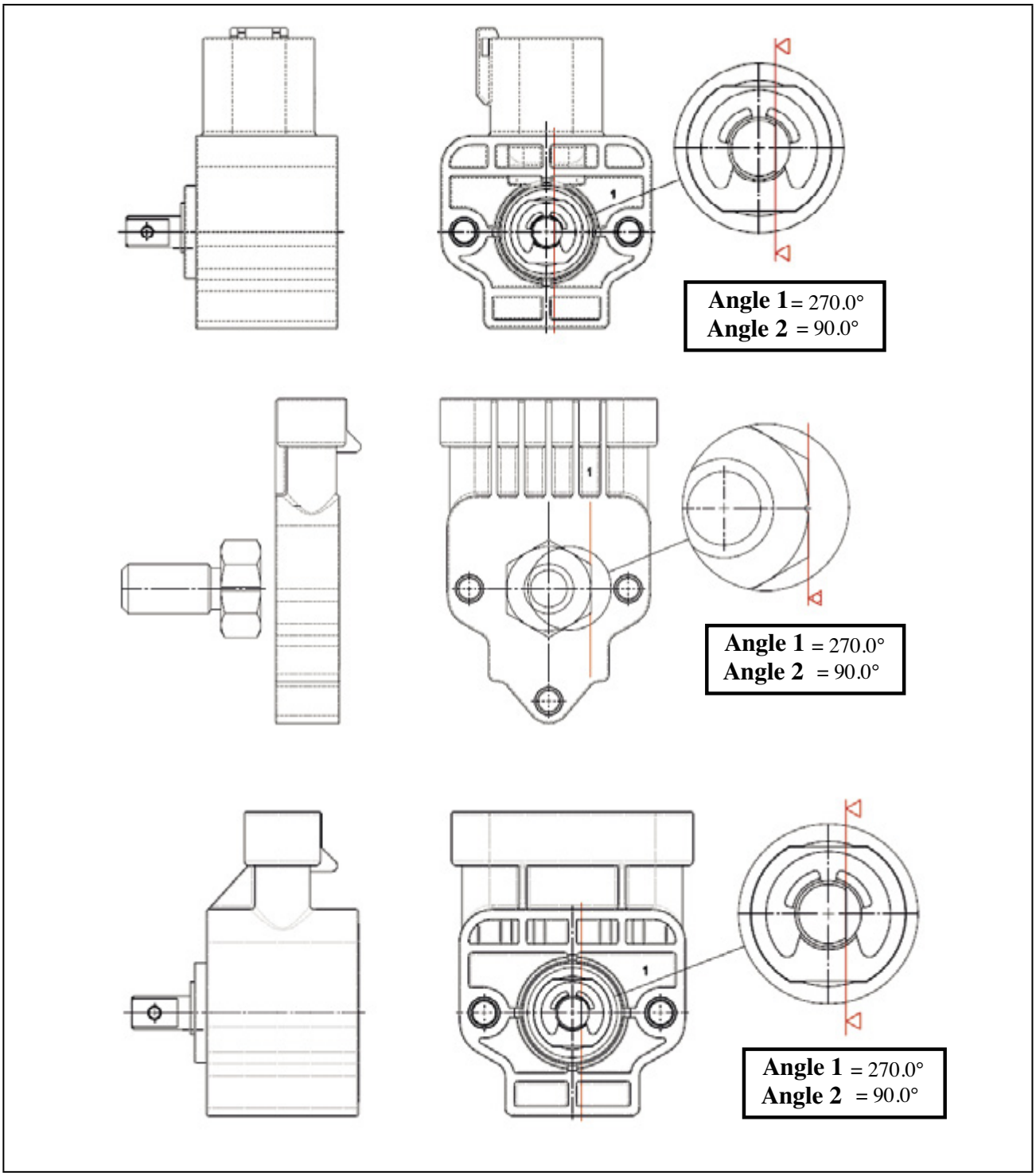
Angle 1 = 0000h to decimal 0d (resolution 0.01°) = 0.00 °

Angle 2:

Byte 3 MSB (00h) = 8Ch Byte 4 LSB (00h) = 9Fh

Angle 2 = 8C9F to decimal 35999d (resolution 0.01°) = 359.99°

In the page an example of PDO mapping is reported in the case of Angle 1 = 270.0° and Angle 2 = 90.0°.



READING RESOLUTION +/- 0.1° (see manufacturing specific object 0x2100 and example (7) at the end of this manual)

Example of PDO mapping for Angle 1 = 270.0° and Angle 2 = 90° (Node-ID = 02h, resolution 0.1°, zero degree point = 0.0°, CCW and big endian)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
182h	0Ah	8Ch	03h	84h	FFh	FFh	FFh	00h

Angle 1:
Byte 1 MSB (00h) = 0Ah Byte 2 LSB (00h) = 8Ch Angle 1 = 0A8Ch to decimal 0d (resolution 0.1°) = 270.0°
Angle 2:
Byte 3 MSB (00h) = 03h Byte 4 LSB (00h) = 84h Angle 2 = 0384h to decimal 900d (resolution 0.1°) = 90.0°

READING RESOLUTION +/- 0.01° (see manufacturing specific object 0x2100 and example (8) at the end of this manual)

Example of PDO mapping for Angle 1 = 270.00° and Angle 2 = 90.00° (Node-ID = 02h, resolution 0.1°, zero degree point = 0.0°, CCW and big endian)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
182h	69h	78h	23h	28h	FFh	FFh	FFh	00h

Angle 1:
Byte 1 MSB (00h) = 69h Byte 2 LSB (00h) = 78h Angle 1 = 6978h to decimal 27000d (resolution 0.01°) = 270.00°
Angle 2:
Byte 3 MSB (00h) = 23h Byte 4 LSB (00h) = 28h Angle 2 = 2328h to decimal 9000d (resolution 0.01°) = 90.00°

Figure 1 below shows the angle calculation for ANGLE 1 and ANGLE 2.

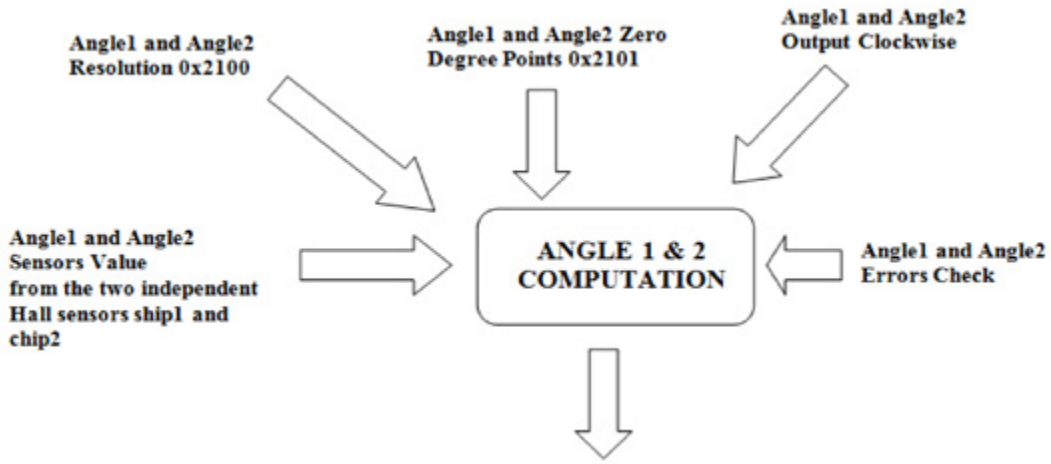


Figure 1 Angle computation

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
182h	00h	00h	0Eh	0Fh	FFh	FFh	FFh	00h

Angle 1 Angle 2

Important note: the two angles Angle 1 and Angle 2 are obtained independently from each other (i.e. in the GRA/GRN series contactless rotary sensors there are two on-board HALL chips) in a sort of redundant configuration.

12. CANopen FEATURES SUMMARY

Communication Profile

The parameters which are critical for communication are determined in the Communication profile.
This area is common for all CANopen devices.

Index	Sub Index	Name	Type	Access	Default value	Comments
1000h		Device Profile	Unsigned 32	Ro	0x00000000	No standardized device profile adopted
1001h		Error Register	Unsigned 8	Ro	0x00	Always ZERO
1008h		Manufacturer Device Name	String	Ro	“GRA” or “GRN”	Refer to GEFRAN products catalogue: GRA: Contactless Rotary Sensors with Shaft GRN: Contactless Rotary Sensors without Shaft
1009h		Manufacturer Hardware Version	String	Ro	“1.00”	
100Ah		Manufacturer Software Version	String	Ro	“1.14”	
1010h	0	Number of Entries	Unsigned 8	Ro	1	“save” (0x65766173) to store all parameters (objects with marking PARA)
	1	Save all Parameters	Unsigned 32	Rw		
1011h	0	Restore Default Parameters	Unsigned 8	Ro		"load" (0x64616F6C) to restore all parameters (objects with marking PARA and LSS-PARA).
	1	Restore all Parameters	Unsigned 32	Rw		
1014h	0	Emergency ID	Unsigned 32	Rw	0x80 + Nodo-ID	
1017h	0	Producer Time / Heart Beat	Unsigned 16	Rw	0	Min= 0 & Max=65535 with unit = 1ms If 0: NOT USED
1018h	0	Identity Object	Unsigned 8	Ro	4	Refer to “Gefran Product Overview CANopen” Gefran Vendor ID:0x0000093
	1	Vendor ID	Unsigned 32	Ro	0x0000093	
	2	Product Code	Unsigned 32	Ro	0x0000064	
	3	Revision Number	Unsigned 32	Ro	0x0000001	
	4	Serial Number	Unsigned 32	Ro	0x0000000	
1200h	SDO Server Parameter					
	0	Number of Entries	Unsigned 8	Ro	2	
	1	COB-ID Client to Server (Rx)	Unsigned 32	Ro	0x600+ Nodo-ID	
	2	COB-ID Server to Server (Tx)	Unsigned 32	Ro	0x580+ Nodo-ID	
1800h	0	1st Transmit PDO Parameter	Unsigned 8	Ro	Default valueRo	CommentsRo
	1	COB-ID	Unsigned 32	Ro	180h + Nodo-ID	Comments180h NodoID
	2	Transmission Type	Unsigned 8	Rw	254	Asynchronous transmission.
	3	Inhibit Time	Unsigned 16	Ro	0	Min= 0 & Max=65535
	4	Reserved	//	//	14	Min 14 Max14 with unit 1ms
	5	Timer	Unsigned 16	Rw	100	Min= 4 & Max=65535
1A00h	Tx PDO Mapping Parameter					Objects
	0	Number of entries	Unsigned 8	Ro	8	
	1	1st Mapped Object	Unsigned 32	Ro	0x21100108	0x2110.1 0x2110.2 0x2110.3 0x2110.4 0x2110.5 0x2110.6 0x2110.7 0x4001
	2	2nd Mapped Object	Unsigned 32	Ro	0x21100208	
	3	3st Mapped Object	Unsigned 32	Ro	0x21100308	
	4	4th Mapped Object	Unsigned 32	Ro	0x21100408	
	5	5th Mapped Object	Unsigned 32	Ro	0x21100508	
	6	6th Mapped Object	Unsigned 32	Ro	0x21100608	
	7	7th Mapped Object	Unsigned 32	Ro	0x21100708	
	8	8th Mapped Object	Unsigned 32	Ro	0x40010008	

Ro = the parameter can be read only
Rw = the parameter can be read and also written
Wo = the parameter can be written only

Manufacturer Specific Profile Objects

In this section you will find the manufacturer specific profile indices for the transducer.

Index	Sub Index	Name	Type	Access	Default value	Comments
2000h		Angle Value				Angle sensor1 and Angle sensor2 in a single chip with 360° fullscale and a resolution of about 0.022°/bit Min= 0 & Max=16383
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 Sensor Value	Unsigned 16	Ro	0	
	2	Angle 2 Sensor Value	Unsigned 16	Ro	0	
2001h		Angle FILTER function				Min= 0 & Max=255 0:Not used NOT IMPLEMENTED
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 FILTER	Unsigned 8	Rw	0	
	2	Angle 2 FILTER	Unsigned 8	Rw	0	
2011h		Angle Sensor Process Data				
	0	Number of Entriers	Unsigned 8	Ro	3	
	1	Angle 1 Process data	Unsigned 16	Ro	0	Angle 1 Process data
	2	Angle 2 Process data	Unsigned 16	Ro	0	Angle 2 Process data
	3	Angle 1 & 2 Process data	Unsigned 32	Ro	0	Angle 1 & 2 Process data sent at the same time
2100h		Angle RESOLUTION				The Angle1 and Angle2 RESOLUTION accepted values are: 1000d: 1 Deg/bit 100d: 0.1 Deg/bit 22d:0.01 Deg/bit (14 bits effective resolution 0.02°)
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 Resolution	Unsigned 16	Rw	100	
	2	Angle 2 Resolution	Unsigned 16	Rw	100	
2101h		Angle ZERO Degree Point				The Angle1 and Angle2 ZERO Degree Point have to be in relation with the maximum allowed degree Min= 0 & Max=16383
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 ZERO Degree Point	Unsigned 16	Ro	0	
	2	Angle 2 ZERO Degree Point	Unsigned 16	Ro	0	
2102h		Angle Clockwise				0: CCW 1:CW
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 Clockwise	Unsigned 8	Rw	0	
	2	Angle 2 Clockwise	Unsigned 8	Rw	1	
2103h		Angle Maximum Difference			0	NOT IMPLEMENTED
2110h		Angle1 and Angle2 values				
	0	Number of Entries	Unsigned 8	Ro	4	
	1	Angle 1 Output Value Byte 0	Unsigned 8	Ro	0	Unit: 0x2100 MSB when PDO coding Big Endian (index 0x5001) is used Min=0 & Max=255
	2	Angle 1 Output Value Byte 1	Unsigned 8	Ro	0	Unit: 0x2100 MSB when PDO coding Little Endian (index 0x5001) is used Min=0 & Max=255
	3	Angle 1 Output Value Byte 0	Unsigned 8	Ro	0	Unit: 0x2100 MSB when PDO coding Big Endian (index 0x5001) is used Min=0 & Max=255
	4	Angle 1 Output Value Byte 1	Unsigned 8	Ro	0	Unit: 0x2100 MSB when PDO coding Little Endian (index 0x5001) is used Min=0 & Max=255

Ro = the parameter can be read only
Rw = the parameter can be read and also written
Wo = the parameter can be written only

Manufacturer Specific Profile Objects

In this section you will find the manufacturer specific profile indices for the transducer

Index	Sub Index	Name	Type	Access	Default value	Comments
4000h		Error Behavior - PARA	Unsigned 8	Rw	1	0: Pre-operational 1: no state change 2: stopped Min=0 & Max=255
4001h		Error Code	Unsigned 8	Ro	0	0: no error Min=0 & Max=255
5000h		Automatic NMT Start after Power-On - PARA	Unsigned 8	Rw	1	0:not activated 1: activated Min=0 & Max=1
5001h		PDO coding standard used - PARA	Unsigned 8	Rw	0	0: Big Endian 1: Little Endian Min=0 & Max=1
5999h		LSS Parameter				
	0	Number of Entries	Unsigned 8	Ro	3	
	1	Baud rate – LSS-PARA	Unsigned 16	Rw	250	Possible values: 50 kbit/s 125 kbit/s 250 kbit/s 500 kbit/s 800 kbit/s 1000 kbit/s Min=50 & Max=1000 IMPORTANT NOTE: use this service only if one device is connected to the network
	2	Node-ID - LSS-PARA	Unsigned 8	Rw	2	Min=1 & Max=127 IMPORTANT NOTE: use this service only if one device is connected to the network
	3	Store - LSS-PARA	Unsigned 32	Wo		“save” (0x65766173) to store all LSS parameters (object with marking LSS-PARA) IMPORTANT NOTE: use this service only if one device is connected to the network

Ro = the parameter can be read only
Rw = the parameter can be read and also written
Wo = the parameter can be written only

13. COMMUNICATION EXAMPLES

Example 1) How to change the Baud Rate Setting from 250 kbaud to 500 kbaud

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600+Node-ID	8	CMD	Index		Sub-Index	Data			

CMD *determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex Sending of 1-byte data (byte 5 contains an 8-bit value)

Structure of SDO-answer by the Slave

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580+Node-ID	8	CMD	Index		Sub-Index	Data			

RES *Response of the slave*:
60 hex Data sent successfully
80 hex Error,

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	2Bh	99h	59h	01h	F4h	01h	00h	00h

Object:

5999h		LSS Parameter				
	0	Number of Entries	Unsigned 8	Ro	3	
	1	Baud rate – LSS-PARA	Unsigned 16	Rw	500	Possible values: 50 kbit/s 125 kbit/s 250 kbit/s 500 kbit/s 800 kbit/s 1000 kbit/s Min=50 & Max=1000 **IMPORTANT NOTE: use this service only if one device is connected to the network**
	2	Node-ID – LSS-PARA	Unsigned 8	Rw	2	Min=1 & Max=127 **IMPORTANT NOTE: use this service only if one device is connected to the network**
	3	Store - LSS-PARA	Unsigned 32	Wo		"save" (0x65766173) to store all LSS parameters (object with marking LSS-PARA) **IMPORTANT NOTE: use this service only if one device is connected to the network**

The supported baudrate are listed in the following table:

Byte5	Byte6	BaudRate
32h	00h	50Kbaud
7Dh	00h	125Kbaud
FAh	00h	250Kbaud
F4h	01h	500Kbaud
20h	03h	800Kbaud
E8h	03h	1Mbaud

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	99h	59h	01h	00h	00h	00h	00h

With the aim to save new Baud Rate write the “save” command as below:

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	23h	99h	59h	03h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code “**s**”
61h = ASCII code “**a**”
76h = ASCII code “**v**”
65h = ASCII code “**e**”

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	99h	59h	03h	00h	00h	00h	00h

IMPORTANT NOTE:
After setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

Example 2) How to change the ID-Node from 0x03h (3d) to 0x06h (6d)

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600+Nodo-ID	8	CMD	Index		Sub-Index	Data			

CMD *determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex Sending of 1-byte data (byte 5 contains an 8-bit value)

Structure of SDO-answer by the Slave

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580+Nodo-ID	8	CMD	Index		Sub-Index	Data			

RES *Response of the slave*:
60 hex Data sent successfully
80 hex Error,

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	2Fh	99h	59h	02h	06h	00h	00h	00h

Object:

5999h		LSS Parameter				
	0	Number of Entries	Unsigned 8	Ro	3	
	1	Baud rate – LSS-PARA	Unsigned 16	Rw	250	Possible values: 50 kbit/s 125 kbit/s 250 kbit/s 500 kbit/s 800 kbit/s 1000 kbit/s Min=50 & Max=1000 **IMPORTANT NOTE: use this service only if one device is connected to the network**
	2	Node-ID – LSS-PARA	Unsigned 8	Rw	6	Min=1 & Max=127 **IMPORTANT NOTE: use this service only if one device is connected to the network**
	3	Store - LSS-PARA	Unsigned 32	Wo		“salva” (0x65766173) per archiviare tutti i parametri LSS (oggetto con marcatura LSS-PARA) **IMPORTANT NOTE: use this service only if one device is connected to the network**

The supported Node-ID are 0x01 to 0x7F :

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	99h	59h	02h	00h	00h	00h	00h

With the aim to save new Baud Rate write the “save” command as below:

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	23h	99h	59h	03h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code “**s**”
61h = ASCII code “**a**”
76h = ASCII code “**v**”
65h = ASCII code “**e**”

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	99h	59h	03h	00h	00h	00h	00h

IMPORTANT NOTE:
After setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

Exemple 3) How to activate an automatic NMT Start after Power ON (the PDO will be send automatically after power ON)

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600+Nodo-ID	8	CMD	Index		Sub-Index	Data			

CMD *determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex Sending of 1-byte data (byte 5 contains an 8-bit value)

Structure of SDO-answer by the Slave

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580+Nodo-ID	8	CMD	Index		Sub-Index	Data			

RES *Response of the slave:*
60 hex Data sent successfully
80 hex Error,

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	2Fh	00h	50h	00h	01h	00h	00h	00h

Object:

5000h		Automatic NMT Start after Power-On - PARA	Unsigned 8	Rw	1	0: not activated 1: activated Min=0 & Max=1

With the aim to save functionality write the “save” command as below:

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code “**s**”
61h = ASCII code “**a**”
76h = ASCII code “**v**”
65h = ASCII code “**e**”

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	10h	10h	01h	00h	00h	00h	00h

IMPORTANT NOTE:
After setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

Example 4) How to change the PDO rate (time interval) from 100 ms to 20 ms

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600+Node-ID	8	CMD	Index		Sub-Index	Data			

CMD *determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex Sending of 1-byte data (byte 5 contains an 8-bit value)

Structure of SDO-answer by the Slave

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580+Node-ID	8	CMD	Index		Sub-Index	Data			

RES *Response of the slave:*
60 hex Data sent successfully
80 hex Error,

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	2Bh	00h	18h	05h	14h	00h	00h	00h

Object:

1800h	0	1st Transmit PDO Parameter	Unsigned 8	Ro		
	1	COB-ID	Unsigned 32	Ro	180h+ Node-ID	
	2	Transmission Type	Unsigned 8	Rw	254	Asynchronous transmission.
	3	Inhibit Time	Unsigned 16	Ro	0	Min=0 & Max=65535 with unit=1ms
	4	Reserved	//	//		
	5	Timer	Unsigned 16	Rw	20	Min= 4 & Max=65535 with unit=1ms

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	00h	18h	05h	00h	00h	00h	00h

With the aim to save functionality write the “save” command as below:

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code::

73h	61h	76h	65h

Where:
73h = ASCII code “**s**”
61h = ASCII code “**a**”
76h = ASCII code “**v**”
65h = ASCII code “**e**”

La risposta dopo memorizzazione corretta è:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	10h	10h	01h	00h	00h	00h	00h

IMPORTANT NOTE:
After setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

Example 5) How to set the ZERO degree point to Angle 1 (example with resolution ± 0.1°)

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600+Node-ID	8	CMD	Index		Sub-Index	Data			

***CMD** determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex Sending of 1-byte data (byte 5 contains an 8-bit value)

Structure of SDO-answer by the Slave

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580+Node-ID	8	CMD	Index		Sub-Index	Data			

***RES** Response of the slave:*
60 hex Data sent successfully
80 hex Error,

Write (in the example the Node-ID = 0x03)
If the actual value of the Angle 1 is 02h 65 h = 0265 h = 613d = 61.3 ° with the aim to move the Angle 1 to ZERO add to Byte 5 and Byte 6 the values below:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	2Bh	01h	21h	01h	65h	02h	00h	00h

Object:

2101h		Angle ZERO Degree Point				The Angle1 and Angle2 ZERO Degree Point have to be in relation with the maximum allowed degree Min= 0 & Max=16383
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 ZERO Degree Point	Unsigned 16	Rw	613	
	2	Angle 2 ZERO Degree Point	Unsigned 16	Rw	0	

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	01h	21h	01h	00h	00h	00h	00h

With the aim to save functionality write the “save” command as below:

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code “**s**”
61h = ASCII code “**a**”
76h = ASCII code “**v**”
65h = ASCII code “**e**”

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	10h	10h	01h	00h	00h	00h	00h

IMPORTANT NOTE:
After setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

Example 6) How to set the ZERO degree point to Angle 2 (example with resolution ± 0.1°)

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600+Node-ID	8	CMD	Index		Sub-Index	Data			

***CMD** determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex Sending of 1-byte data (byte 5 contains an 8-bit value)

Structure of SDO-answer by the Slave

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580+Node-ID	8	CMD	Index		Sub-Index	Data			

***RES** Response of the slave:*
60 hex Data sent successfully
80 hex Error,

Write (in the example the Node-ID = 0x03)
If the actual value of the Angle 2 is 02h 65 h = 0265 h = 613d = 61.3 ° with the aim to move the Angle 2 to ZERO add to Byte 5 and Byte 6 the values below:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	2Bh	01h	21h	02h	65h	02h	00h	00h

Object:

2101h		Angle ZERO Degree Point				The Angle1 and Angle2 ZERO Degree Point have to be in relation with the maximum allowed degree Min= 0 & Max=16383
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 ZERO Degree Point	Unsigned 16	Rw	0	
	2	Angle 2 ZERO Degree Point	Unsigned 16	Rw	613	

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	01h	21h	02h	00h	00h	00h	00h

With the aim to save functionality write the “save” command as below:

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code "**s**"
61h = ASCII code "**a**"
76h = ASCII code "**v**"
65h = ASCII code "**e**"

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	10h	10h	01h	00h	00h	00h	00h

IMPORTANT NOTE:
After setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

Example 7) How to set the resolution to ± 0.1° on Angle 1 and Angle 2 (the current setting resolution is ± 0.01°)

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600+Node-ID	8	CMD	Index		Sub-Index	Data			

CMD *determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex Sending of 1-byte data (byte 5 contains an 8-bit value)

Structure of SDO-answer by the Slave

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580+Node-ID	8	CMD	Index		Sub-Index	Data			

RES *Response of the slave:*
60 hex Data sent successfully
80 hex Error

With the aim to set the resolution on Angle 1 to ± 0.1° send write the PDO (in the example the Node-ID = 0x03):

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	2Bh	00h	21h	01h	64h	00h	00h	00h

Object:

2100h		Angle RESOLUTION				The Angle1 and Angle 2 RESOLUTION accepted values are: 1000d: 1 Deg/bit 100d: 0.1 Deg/bit 22d:0.01 Deg/bit (14 bits effective resolution 0.02°)
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 Resolution	Unsigned 16	Rw	100	
	2	Angle 2 Resolution	Unsigned 16	Rw	22	

With the aim to set the resolution on Angle 2 to ± 0.1° send write the PDO (in the example the Node-ID = 0x03):

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	2Bh	00h	21h	02h	64h	00h	00h	00h

Object:

2100h		Angle RESOLUTION				The Angle1 and Angle 2 RESOLUTION accepted values are: 1000d: 1 Deg/bit 100d: 0.1 Deg/bit 22d:0.01 Deg/bit (14 bits effective resolution 0.02°)
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 Resolution	Unsigned 16	Rw	100	
	2	Angle 2 Resolution	Unsigned 16	Rw	100	

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	00h	21h	01h	00h	00h	00h	00h

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	00h	21h	02h	00h	00h	00h	00h

With the aim to save functionality write the “save” command as below:

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code “**s**”
61h = ASCII code “**a**”
76h = ASCII code “**v**”
65h = ASCII code “**e**”

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	10h	10h	01h	00h	00h	00h	00h

IMPORTANT NOTE:
After setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

Example 8) How to set the resolution to ± 0.01° on Angle 1 and Angle 2 (the current setting resolution is ± 0.1°)

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600+Node-ID	8	CMD	Index		Sub-Index	Data			

CMD *determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex Sending of 1-byte data (byte 5 contains an 8-bit value)

Structure of SDO-answer by the Slave

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580+Node-ID	8	CMD	Index		Sub-Index	Data			

RES *Response of the slave*:
60 hex Data sent successfully
80 hex Error,

With the aim to set the resolution on Angle 1 to ± 0.01° send write the PDO (in the example the Node-ID = 0x03):

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	2Bh	00h	21h	01h	16h	00h	00h	00h

Object:

2100h		Angle RESOLUTION				The Angle1 and Angle 2 RESOLUTION accepted values are: 1000d: 1 Deg/bit 100d: 0.1 Deg/bit 22d:0.01 Deg/bit (14 bits effective resolution 0.02°)
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 Resolution	Unsigned 16	Rw	22	
	2	Angle 2 Resolution	Unsigned 16	Rw	100	

With the aim to set the resolution on Angle 2 to ± 0.01° send write the PDO (in the example the Node-ID = 0x03):

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	2Bh	00h	21h	02h	16h	00h	00h	00h

Object:

2100h		Angle RESOLUTION				The Angle1 and Angle 2 RESOLUTION accepted values are: 1000d: 1 Deg/bit 100d: 0.1 Deg/bit 22d:0.01 Deg/bit (14 bits effective resolution 0.02°)
	0	Number of Entries	Unsigned 8	Ro	2	
	1	Angle 1 Resolution	Unsigned 16	Rw	22	
	2	Angle 2 Resolution	Unsigned 16	Rw	22	

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	00h	21h	01h	00h	00h	00h	00h

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	00h	21h	02h	00h	00h	00h	00h

With the aim to save functionality write the “save” command as below:

Write (in the example the Node-ID = 0x03)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
603h	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code “**s**”
61h = ASCII code “**a**”
76h = ASCII code “**v**”
65h = ASCII code “**e**”

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
583h	60h	10h	10h	01h	00h	00h	00h	00h

IMPORTANT NOTE:
After setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

Example 9) How to send the command RESTORE

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa

Structure of SDO-request by the Master

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

CMD *determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex Sending of 1-byte data (byte 5 contains an 8-bit value)

Structure of SDO-answer by the Slave

COB-ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

RES *Response of the slave:*
60 hex Data sent successfully
80 hex Error,

Write (in the example the Node-ID = 0x7F)

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
67Fh	23h	11h	10h	01h	6Ch	6Fh	61h	64h

Object:

1011h	1	Load all parameters	Unsigned 8	Wo		"load" (0x64616F6C) to restore all parameters (objects with marking PARA and LSSPARA).

The answer after successful storing you will receive is:

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
5FFh	60h	11h	10h	01h	00h	00h	00h	00h

IMPORTANT NOTE:
After setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

NOTES

NOTES