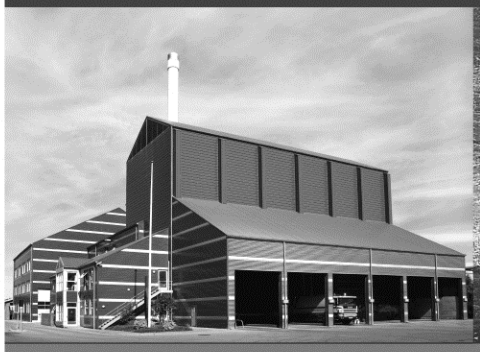
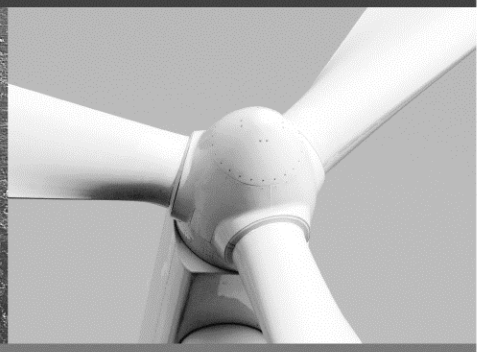
DEIF

-power in control

CE

XL/BW series





CAN specification

DEIF A/S · Frisenborgvej 33 · DK-7800 Skive
Tel.: +45 9614 9614 · Fax: +45 9614 9615
info@deif.com · www.deif.com

Document no.: 4189350023D

Table of contents

1. Dual CANopen specification

This document describes the standard serial interface to the XL products.

The CANopen application is based on:

- *CiA Draft Standard 301- Application Layer and Communication Profile - Version 4.02*
- *CiA Draft Standard Proposal 302 - Framework for CAN open Managers and Programmable CAN open Devices - Version 3.3.0*
- *CiA Draft Standard Proposal 305 - Layer Setting Services and Protocol - Version 1.1.1*

The XL indicator is always the slave device. Each XL indicator requires its own Node-ID. It is not the purpose of this protocol to describe all the functionalities of the CANopen communication, which is implemented and running according to the CANopen standards and needs no consideration from the user. Please use the website **www.can-cia.org** to download a detailed explanation of the CANopen description.

If the instrument has been reset during normal use, e.g. due to power droop, then the instrument must be initialised again, and values for pointer and illumination must be updated. If the instrument is arranged as a self-starting device, the initialising will automatically be carried out, when the next package of data is sent to the instrument.

EDS file

The EDS file (Electronic Data Sheets) can be used as configuration tools for the implementation of the CANopen features supported in the XL indicator.

Transfer on the CANbus line

Multiplexed Process Data Objects (MPDO) provide direct access to application objects within a device. They are used to perform real-time transfers of short blocks of high priority data. Each MPDO message must contain one COB-ID (Communication Object-Identifier), index and sub-index and two bytes of data.

MPDO messages are used for transfer of pointer and dimmer values.

Service Data Objects (SDO) can be used for access with read/write attributes of all objects (some objects can only be accessed with reading or writing attributes) implemented in the XL indicator regarding CANopen, e.g. the master can read the COB-ID of an MPDO telegram with SDO transfer from the slave device, using index 0x1400, sub-index 0x01 and the read attribute.

Network Management Transfer (NMT) is used to control the application in the slave device from the master.

Baud rate

Baud rate	Bus length
125 kbit/s	500 m
250 kbit/s	250 m

Default setup values

- Baud rate 125 kbit/s
- Node-ID number 1
- MPDO 0x201

SDO transfer

SDO transfers are used to access all objects implemented in the XL unit regarding CANopen, e.g. read Identity Object index 0x1018 sub 1,2,3 and 4.

- Read Vendor ID index 0x1018 sub-index 0x01, Node-ID 1.

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Don't care	Don't care	Don't care	Don't care
0x601	0x40	0x18	0x10	0x01	0x00	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x580 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Data	Data	Data	Data
0x581	0x43	0x18	0x10	0x01	0xB2	0x00	0x00	0x00

Vendor ID = 0xB2

- Read Product code index 0x1018 sub-index 0x02, Node-ID 1.

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Don't care	Don't care	Don't care	Don't care
0x601	0x40	0x18	0x10	0x02	0x00	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x580 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Data	Data	Data	Data
0x581	0x43	0x18	0x10	0x02	0xXX	0xXX	0xXX	0xXX

- Read Revision number index 0x1018 sub-index 0x03, Node-ID 1.

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Don't care	Don't care	Don't care	Don't care
0x601	0x40	0x18	0x10	0x03	0x00	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x580 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Data	Data	Data	Data
0x581	0x43	0x18	0x10	0x03	0xXX	0xXX	0xXX	0xXX

- Read Serial number index 0x1018 sub-index 0x04, Node-ID 1.

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Don't care	Don't care	Don't care	Don't care
0x601	0x40	0x18	0x10	0x04	0x00	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x580 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Data	Data	Data	Data
0x581	0x43	0x18	0x10	0x04	0xXX	0xXX	0xXX	0xXX

The serial number is unique for each XL node.

- Write XLPointer number index 0x6411 sub-index 0x01, Node-ID 1.

Ex.: XL pointer position = 15000

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Data	Data	Don't care	Don't care
0x601	0x2B	0x11	0x64	0x01	0x98	0x3A	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x580 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Don't care	Don't care	Don't care	Don't care
0x581	0x60	0x11	0x64	0x01	0x00	0x00	0x00	0x00

- Write XLDimmer number index 0x6411 sub-index 0x02, Node-ID 1.

Ex.: XL dimmer level = 255

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Data	Don't care	Don't care	Don't care
0x601	0x2B	0x11	0x64	0x02	0xFF	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x580 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	Don't care	Don't care	Don't care	Don't care
0x581	0x60	0x11	0x64	0x02	0x00	0x00	0x00	0x00

Heartbeat protocol

- Produce Heartbeat

  Time base - 1 ms
  A Heartbeat Producer transmits a Heartbeat message cyclically. To activate Heartbeat protocol, write to address index 0x1017 and sub-index 0x00.

  Non-volatile memory is supported, and node has default value zero.

Ex.: Heartbeat interval = 1000 ms, Node-ID 1.

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	$Data_{lsb}$	Data	Data	$Data_{msb}$
0x601	0x2b	0x17	0x10	0x00	0xE8	0x03	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x580 + Node ID	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	$Data_{lsb}$	Data	Data	$Data_{msb}$
0x581	0x60	0x17	0x10	0x00	0x00	0x00	0x00	0x00

- Consume Heartbeat

  Time base - 1 ms
  Consume Heartbeat is a way of a node to identify the network and is e.g. used for redundant CANopen nodes to decide which CAN line is active and when to switch CAN line.

  Nodes to be monitored are configured in object 0x1016.
  E.g. in units with Node-ID 1 to be monitored with 1000 ms, use index 0x1016 and sub-index 0x01 with SDO transfer.

  Non-volatile memory is supported, and nodes have default value zero.

Request

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	$Data_{lsb}$	Data	Data	$Data_{msb}$
0x601	0x2b	0x16	0x10	0x01	0xE8	0x03	0x02	0x00

Response

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	$Index_{lsb}$	$Index_{msb}$	Sub-index	$Data_{lsb}$	Data	Data	$Data_{msb}$
0x581	0x2b	0x16	0x10	0x01	0xE8	0x03	0x02	0x00

Objects implemented for instrument functionality

MPDO is supported for transfer of instrument objects.

- Pointer control
  Index 0x6411 sub-index 0x01 type unsigned integer (2 bytes).

  E.g. pointer 0 deg. MPDO transfer, Node-ID 1.

Messages from external unit to XL unit

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COB-ID = 0x200 + Node-ID	Addr	$Index_{lsb}$	$Index_{msb}$	Sub-index	$Data_{lsb}$	$Data_{msb}$	Data	Data
0x201	0x81	0x11	0x64	0x01	0x00	0x80	Not used	Not used

E.g. pointer 45 deg. MPDO transfer, Node-ID 1.

Messages from external unit to XL unit

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COB-ID = 0x200 + Node-ID	Addr	$Index_{lsb}$	$Index_{msb}$	Sub-index	$Data_{lsb}$	$Data_{msb}$	Data	Data
0x201	0x81	0x11	0x64	0x01	0xFF	0x1F	Not used	Not used

- Dimmer control
  Index 0x6411 sub-index 0x02 type unsigned integer (2 bytes).

  257 levels: Level 0 (zero light) to level 256 (max. light).
  Parameters from 0x0000 to 0x0100.

  E.g. level 256. MPDO transfer, Node-ID 1.

Messages from external unit to XL unit

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COB-ID = 0x200 + Node-ID	Addr	$Index_{lsb}$	$Index_{msb}$	Sub-index	$Data_{lsb}$	$Data_{msb}$	Data	Data
0x201	0x81	0x11	0x64	0x02	0xFF	0x00	Not used	Not used

Changing Baud rate or Node-ID using LLS protocol

Implemented according to Layer Setting Services and Protocol CiA Draft Standard Proposal 305 Version 1.1.1.

Short description:
It is possible to change Baud rate or Node-ID in NMT states stop, pre-operational and operational. In e.g. NMT state pre-operational, a configuration tool (e.g. Master) has to change the sub-state of the Layer Setting Services protocol to configuration state.

- **Switch Mode Global**
  All slave units which support Layer Setting Services protocol change state to LSS configuration state.
  This **Switch Mode Global can only be used for change of Baud rate and to request information about Node XL units to be used for LLS protocol.**

- **Switch Mode Selective**
  A single Node is switched to configuration mode.
  This **Switch Mode Selective is used for change of Node-ID in a single XL unit.**

- **Service request in LSS configuration mode**
  The following services can be used to request information about an XL Node to be used for later process.

Switch Mode Global (from operation mode to configuration mode)

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Switches to configuration mode	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E5	0x04	0x01	0x00	0x00	0x00	0x00	0x00	0x00

a) Request Vendor-ID from XL Node

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E5	0x5A	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Vendor-id_{lsb}	Vendor-id	Vendor-id	Vendor-id_{msb}	Reserved	Reserved	Reserved
0x7E4	0x5A	0xB2	0x00	0x00	0x00	0x00	0x00	0x00

b) Request Product-Code Protocol

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E5	0x5B	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Product-id_{lsb}	Product-id	Product-id	Product-id_{msb}	Reserved	Reserved	Reserved
0x7E4	0x5B	0xXX	0xXX	0xXX	0xXX	0x00	0x00	0x00

c) Request Revision-Number Protocol

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E5	0x5C	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Revision-no_{lsb}	Revision-no	Revision-no	Revision-no_{msb}	Reserved	Reserved	Reserved
0x7E4	0x5C	0xXX	0xXX	0xXX	0xXX	0x00	0x00	0x00

d) Request Serial-Number Protocol

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E5	0x5D	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Serial-no_{lsb}	Serial-no	Serial-no	Serial-no_{msb}	Reserved	Reserved	Reserved
0x7E4	0x5D	0xXX	0xXX	0xXX	0xXX	0x00	0x00	0x00

e) Request Node-ID Protocol

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E5	0x5E	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Node-id(NID)	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E4	0x5E	0xXX	0x00	0x00	0x00	0x00	0x00	0x00

Together, this information identifies a single unique XL Node in any CANopen network.

- E.g. change Baud rate using Switch Mode Selective Node-ID 2, Vendor-ID: 0xB2, Product-Code: 0x00, Revision-Number: 0xFF and Serial-Number: 0x10

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Vendor-id$_{(lsb)}$	Vendor-id	Vendor-id	Vendor-id$_{(msb)}$	Reserved	Reserved	Reserved
0x7E5	0x40	0xB2	0x00	0x00	0x00	0x00	0x00	0x00

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Product-code $_{(lsb)}$	Product-code $_{(lsb)}$	Product-code $_{(lsb)}$	Product-code $_{(msb)}$	Reserved	Reserved	Reserved
0x7E5	0x41	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Revision-no$_{lsb}$	Revision-no	Revision-no	Revision-no$_{msb}$	Reserved	Reserved	Reserved
0x7E5	0x42	0xFF	0x00	0x00	0x00	0x00	0x00	0x00

Request

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Serial-no$_{lsb}$	Serial-no	Serial-no	Serial-no$_{msb}$	Reserved	Reserved	Reserved
0x7E5	0x43	0x10	0x00	0x00	0x00	0x00	0x00	0x00

Response

CAN-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E4	0x44	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Node 2 is in LSS configuration mode.

Configure Bit Timing Parameters Protocol.
XL-Node can be configured to 250 kbit/s (Index = 0x03) or 125 kbit/s (Index = 0x04).

- E.g. change Baud rate to 250 kbit/s

Request

Node-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Table selector	Table index(0x00 CANopen)	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E5	0x13	0x00	0x03	0x00	0x00	0x00	0x00	0x00

Response

Node-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Error code(0x00 protocol successfully completed)	Spec. error(not used)	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E4	0x13	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Activate Bit Timing Parameters Protocol.

- E.g. implement the Bit Timing Parameters
  Switch delay time is set to 1000 ms (time unit of switch delay is 1 ms)

Request

Node-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	$Switch\_delay_{lsb}$	$Switch\_delay_{msb}$	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E5	0x15	0xE8	0x03	0x00	0x00	0x00	0x00	0x00

After approximately 1000 ms, when receiving this message, the Baud rate is changed to 250 kbit/s.

To store the parameter for 250 kbit/s Baud rate, the following message can be used.

Request

Node-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E5	0x17	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Response

Node-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command	Error code(0x00 protocol successfully completed)	Spec. error(not used)	Reserved	Reserved	Reserved	Reserved	Reserved
0x7E4	0x17	0x00	0x00	0x00	0x00	0x00	0x00	0x00

The Baud rate is now changed to 250 kbit/s and saved in non-volatile memory.

Redundant communication

It is based on a communication using two separate physical CAN wires. One line is called the *Default-CAN line* and the other the *Redundant-CAN line*. This is marked on the instrument. The communication starts on the *Default-CAN line*. If the line is disturbed or fails, the communication switches to the *Redundant-CAN line*. The *Default-CAN line* is determined and becomes active, when 3 heartbeats are received from all redundant nodes on the default line.

At cold boot and after a *Reset Communication,* a timer is started. If the *Default-CAN line* is not determined within the period and the timer expires, then the *Redundant-CAN line* will become the active line.

All messages are processed for both CAN lines.

Non-volatile memory

Object 0x1010 sub-index 0x01 is supported.
Parameters supported:
Self-starting device: Object 0x2000 sub-index 0x00.
Producer Heartbeat: Object 0x1017 sub-index 0x00.
Consumer Heartbeat: Object 0x1016 sub-index 0x00 to 0x014.

Restore non-volatile memory

Object 0x1011 sub-index 0x01 is supported.
The last stored parameters can be restored.

LED

If both communication lines are down, the LED will be flashing and the pointer will move to illegal position.

Self-starting device

Self-starting device is the ability to change to the operational state without an NMT command from a master. Time period on index 0x2000 sub 0x00 is used to check for NMT command, if the time period has expired without an NMT command. For changes of states, the device will automatically enter operational state.

If the value of index 0x2000 sub-index 0x00 is zero, state self-starting device is not active.

Non-volatile memory supports this object.

2. sCAN specification

When DEIF single CAN protocol is ordered, XL acts as a listen only device on CANopen. It will receive and present data from TPDO1 with the CAN–ID: 0x180 + “Source node-ID”. F. ex. with Source Node-ID=5, XL will receive the TPDO with CAN-ID: 0x185

Order specifications:
The below 3 tables show the configurations for the different main groups. When ordering, you will need only to state desired CAN Source Node-ID number (1-127), application (General, Azimuth, Pitch, RPM) and input type (12-bit, 16-bit, Absolute):

12-bit encoder

Input	Source NODE ID	Input MIN	Input CENTER	Input MAX
Old version*	1(1-127)	0	2047	4095
Azimuth	1(1-127)	0	2047	4095
Pitch	2(1-127)	0	2047	4095
General	5(1-127)	0	2047	4095

-12-bit encoder uses only positive values.
-All input types can be used with positive values from 12-15 bit.

16-bit encoder

Input	Source NODE ID	Input MIN	Input CENTER	Input MAX
Old version*	1(1-127)	-32768	0	32767
Azimuth	1(1-127)	0	32767	65535
Pitch	2(1-127)	-32768	0	32767
General	5(1-127)	-32768	0	32767

Absolute input

Input	Source NODE ID	Input MIN	Input CENTER	Input MAX
Old version*	1(1-127)	-32768	0	32767
Azimuth	1(1-127)	0	32767	65535
Pitch	2(1-127)	-32768	0	32767
General	5(1-127)	-32768	0	32767

*Old version is only for spare parts for systems delivered before DEIF sCAN was released late 2011!

XL with absolute CAN input can be ordered with any default setting in the span indicated above, however for scales < 360 degree, the MIN value < Center value < MAX value.

Examples – Absolute input:

1) Azimuth 0 - 359.9 degrees. CAN data: 0 to 3559 – Order MIN: 0, Center: 1800, MAX: 3599
2) Pitch +/- 40 degrees. CAN data +/- 400 (0.1 degree resolution)- Order MIN: -400, Center:0, MAX:400
3) RPM 0 - 400 RPM. CAN data: 0 to 400 – Order MIN: 0, Center: 200 MAX: 400

Input types

Input	Function	Note
Old version	Supports old zero adjustment, CW and CCW on 360 degree scales. Supports MIN, CENTER and MAX adjustment on <360 degree scales.	Not recommended for new construction, because old zero adjustment is used. **Should only be used as spare part!**
Azimuth	Supports zero adjustment, CW and CCW on 360 degree scales. Support remote config from MI172	
Pitch General	Supports MIN, CENTER and MAX adjustment on <360 degree scales. Support remote configuration from MI172.	

Manual calibration via CAN2

Azimuth (360 degree scale)

-Zero set – 5 sec
-CCW – 10 sec
-CW – 15 sec

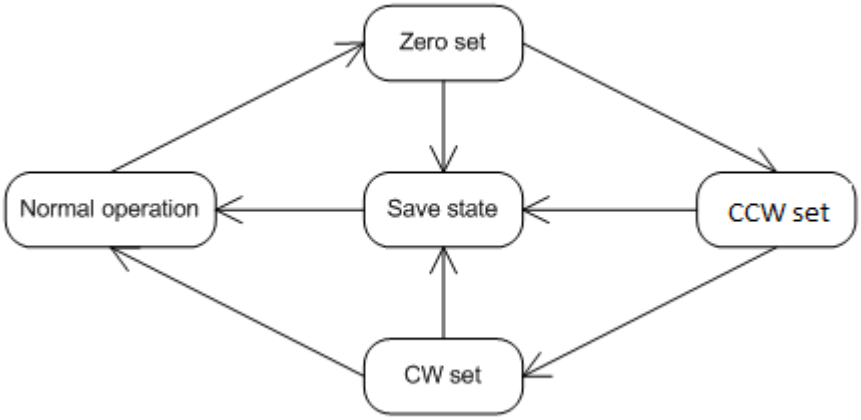


Figure 1 - Azimuth indicator calibration state chart

General indicator (<360 degree scale)

-Zero set – 5 sec
-Max ahead – 10 sec
-Max astern – 15 sec

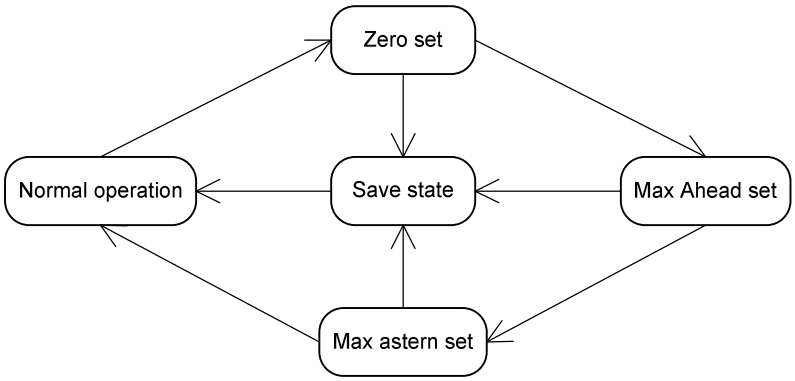


Figure 2 – General indicator calibration state chart

DEIF A/S reserves the right to change any of the above.