CANopen Module for B-sensor Calibration

reference manual

v2.0

7 Jan 2004

Henk Boterenbrood
(boterenbrood@nikhef.nl)
NIKHEF, Amsterdam, NL

Table of Contents

Version History		
Version	Date	Comments
2.0	7 Jan 2004	• Describes firmware version "MB30", minor "0001". • Added support of B-sensor modules with microcontroller. • Added info on timing of conversions on micro-B-sensor using the Simulator.
1.3	3 Dec 2003	• Describes firmware version "MB23", minor "0004". • ADC for NTC by default not present. • Timing of conversions confirmed using Simulator.
1.2	5 Nov 2003	• Describes firmware version "MB23". • Added this document change record. • Added object to set the number of the single input channel of coils to read. • Changed index of 'SYNC generation' object.
1.1	1 May 2003	• Describes firmware version "MB22". • Added description of continuous read-out; update of Object Dictionary tables.
1.0.1	31 Mar 2003	First version.

Table 1. Document change record.

1 B-sensor/Coil-ADC Measurement Cycle

The B-sensor/Coil measurement cycle is a special synchronized read-out scan cycle of both the (multi) ADC-based systems of B-sensors and Coils, used in the NIKHEF ATLAS B-sensor calibration. It is implemented on an ELMB, based on an existing version of the MDT-DCS *CANopen* application firmware for the ELMB (this explains why there are some objects in this application's Object Dictionary (*OD*) left that are meaningless in the B-sensor calibration environment; the objects in question are greyed out in the *OD* tables in section 5).

The new measurement cycle is in addition to the 'standard' ways of readout of the individual ADC-based systems, as described in sections below (and in the document "MDT-DCS CANopen Module, user manual and reference"). The MDT-DCS *CANopen* application has been extended and modified to support the Coil-ADC system and the new measurement cycle.

The synchronized B-sensor/Coil measurement cycle can be performed in two modes: *single-shot* or *continuous*. Continuous mode only works with at least one Coil-ADC present.

A *single-shot* B-sensor/Coil-ADC measurement cycle consists of a sequence of conversion commands and channel read-out operations, as follows:

1. Start a conversion of H1 on all B-sensor ADCs (in quick succession)
2. Start a conversion on all Coil-ADCs (in quick succession)
3. Wait for Coil-ADC conversions to finish
4. Send a SYNC message (optional)
5. Read out all Coil channels (upto 3; wait more if necessary), and send
6. Wait for B-sensor ADC conversions to complete (if necessary)

7. Read out all B-sensor module H1 Hall-sensor channels (upto 5; wait more if necessary), and send
8. Repeat steps 1-7 for the H2 Hall-sensor channels
9. Repeat steps 1-7 for the H3 Hall-sensor channels
10. Repeat steps 1-7 for the T-sensor channels

The delay between the start of step 1 and the start of step 2 is in the order of a few hundred μs (*measured*[1]*: one B-sensor ADC conversion start takes 77 μs, so for 5 B-sensors the delay is about 400 μs; for one micro-B-sensor a conversion start takes 2.5 ms, if all are micro-B-sensors the conversion-start message is broadcast*).

Read-out and sending of the data of one channel takes in the order of 1 ms (*measured: ca. 850 μs, or ca. 950 μs with SYNC message generation*).

The number of B-sensor modules (up to 5) and Coils (up to 3) to be read out, can be configured by writing to the appropriate objects in the Object Dictionary.

Thus the maximum total number of read-out messages/data generated by the ELMB in one measurement cycle is 4 * (3+5) = 32, in the following order:

1. Coil channels (ch 0,1,2) + B-sensor module H1 Hall-sensor channels (ch 0,4,8,12,16)
2. Coil channels (ch 0,1,2) + B-sensor module H2 Hall-sensor channels (ch 1,5,9,13,17)
3. Coil channels (ch 0,1,2) + B-sensor module H3 Hall-sensor channels (ch 2,6,10,14,18)
4. Coil channels (ch 0,1,2) + B-sensor module T-sensor channels (ch 3,7,11,15,19).

One full single-shot cycle roughly takes $T = 4*(P+8)$ ms, with P the time for one conversion to complete on the ADC set to the lowest conversion word rate, so e.g. at 15 Hz conversion wordrate $P = 67$ ms and $T = 4*(67+8) = 300$ ms.

A <u>*continuous*</u> B-sensor/Coil-ADC measurement consists of a sequence of conversion commands and channel read-out operations, as follows:

1. Start a conversion of H1 on all B-sensor ADCs (in quick succession)
2. Start continuous conversions on all Coil-ADCs (in quick succession)
3. Wait for Coil-ADC conversions to finish
4. Send a SYNC message (optional)
5. Read out all Coil channels (upto 3; wait more if necessary), and send
6. Wait for B-sensor ADC conversions to complete (if necessary)
7. Read out all B-sensor module Hx/T Hall-sensor channels (upto 5; wait more if necessary), and send
8. Wait for Coil-ADC conversions to finish
9. Start a conversion of Hx/T on all B-sensor ADCs (in quick succession)
10. Send a SYNC message (optional)
11. Read out all Coil channels (upto 3; wait more if necessary), and send
12. Goto step 3 and continue.

The delay between step 8 and 9 (important to place Hall-sensor data relative to Coil-ADC data, in time) is less than 100 μs (*measured: 85 μs or less*), and the time to start the B-sensor conversions (5x) is about 400 μs (or 2.5 ms in case all B-sensors have a micro, see above), so

[1] using *AVR Studio* Simulator with ATmega128 at 4MHz, 3 Dec 2003.

it is safe to say that all B-sensor conversions are started within 500 μs (or 2.6 ms) of the Coil-ADC conversion-ready.

Data is generated by the ELMB in the following order:

1. Coil channels (ch 0,1,2) + B-sensor module H1 Hall-sensor channels (ch 0,4,8,12,16)
2. Coil channels (ch 0,1,2)
3. Coil channels (ch 0,1,2) + B-sensor module H2 Hall-sensor channels (ch 1,5,9,13,17)
4. Coil channels (ch 0,1,2)
5. Coil channels (ch 0,1,2) + B-sensor module H3 Hall-sensor channels (ch 2,6,10,14,18)
6. Coil channels (ch 0,1,2)
7. Coil channels (ch 0,1,2) + B-sensor module T-sensor channels (ch 3,7,11,15,19).
8. Coil channels (ch 0,1,2)
9. repeat 1-8, etc, etc.

So there is always an extra set of Coil-ADC data between subsequent B-sensor ADC read-out; this extra period is used to initiate the next B-sensor ADC conversions, which are then read out after the *next* conversion of the Coil-ADC completes.

To work properly and guarantee proper timing, the B-sensor ADC conversion wordrate must be equal or higher than the Coil-ADC conversion wordrate, so that B-sensor ADC readout can start immediately after the Coil-ADC conversion completes.

Readout of 3 Coil- and 5 B-channels in total takes in the order of 10 ms, so the conversion wordrate of the Coil-ADCs cannot be higher than 100 Hz (this is for five B-sensor modules in the setup; with less B-sensor modules we can in principle go to even higher rates).

Suppose we let the Coil-ADCs run at 60 Hz, then every B-sensor would be sampled at a rate of 30 Hz (note that in this case the conversion word rate of the B-sensor still needs to be at least 60 Hz !), which means that every individual Hall-sensor is sampled at 30/4 = 7.5 Hz.

In a more noise-tolerant configuration the Coil-ADCs and B-sensor ADCs would probably be set to run at a conversion wordrate of 15 Hz, and then every inidividual Hall-sensor is sampled at a rate of 7.5/4=1.875 Hz.

Messages with ADC data are in **PDO** format, as described in the sections below.

The ELMB has to be put in *Operational* state before any measurement cycles can be triggered.

To trigger a single-shot or continuous measurement cycles send the following message to the ELMB:

Host → MDT-DCS node

COB-ID	DataByte 0
200h + *NodeID*	1

Whether the message triggers a single-shot or continuous measurement cycles depends on the setting of Object 4C00h in the Object Dictionary (use an **SDO** message to change its value).

To abort a single-shot or continuous measurement cycles send the following message to the ELMB:

Host → MDT-DCS node

COB-ID	DataByte 0
200h + *NodeID*	0

(Alternatively, taking the ELMB out of *Operational* state automatically aborts any measurement cycle, single-shot or continuous.)

2 B-sensor Read-out

By writing to *OD* index 2800h, up to five B-sensors can be selected. The default is: no B-sensor present, i.e. Object 2800h has value 0 (zero).

As of version 3.0 of the firmware, B-sensor modules with microcontroller (enabling multiple B-sensor modules on the same SPI-plus-one-chipselect-line bus) are supported. Either module #0 or module #1 to #4 are equipped with a microcontroller, or all are. This can be configured in *OD* index 2801h and 2802h or 2803h.

B-sensor modules equipped with microcontroller do not have an ID-chip. In this case, reading *OD* index 2900h to 2904h results in the 1-byte-significant "SPI address" of the module.

The ELMB module sends one PDO message containing 5 bytes for each B-sensor input and per B-sensor module 4 inputs are read: Hall sensors H1, H2 and H3 and the temperature sensor. The CAN-identifier used for this PDO is the socalled *4th-transmit-PDO* (**TPDO4**) of the CANopen *Predefined Connection Set*.

Note that one B-sensor read-out *scan* reads out only one channel per B-sensor ADC. The next scan will produce the next channel of each connected B-sensor ADC ! (this is different from how the standard MDT-DCS *CANopen* module responds).

The MDT-DCS module produces the following 5-databyte TPDO4:

MDT-DCS node → Host

COB-ID	DataByte 0	DataByte 1	DataByte 2-4
480h + *NodeID*	Channel number	ADC-config	24-bit ADC value

with:

ADC value: signed/unsigned 24-bits ADC value, LSB in byte 2, MSB in byte 4.
Note: Hall sensors: 24-bit signed value; T-sensor: 24-bit unsigned value,

Channel number: Number between 0 and 19.
Chan 0-3: Hall sensor H1, H2, H3 and T-sensor resp. of Bsensor #0,
Chan 4-7: Hall sensor H1, H2, H3 and T-sensor resp. of Bsensor #1.
Chan 8-11: Hall sensor H1, H2, H3 and T-sensor resp. of Bsensor #2.
Chan 12-15: Hall sensor H1, H2, H3 and T-sensor resp. of Bsensor #3.
Chan 16-19: Hall sensor H1, H2, H3 and T-sensor resp. of Bsensor #4.

ADC-config: **bit 7**: Conversion status: 1=ERROR (overflow or oscillation occurred during ADC conversion), 0=OKAY.
bits 6-0: ADC configuration: conversion wordrate (bits W0, W1 and W2), gain range (bits G0, G1 and G2) and unipolar or bipolar (bit U/B); see below.
For definitions see *OD* index 2500h/2501h, sub 2,3,4,5,6 and 7.

BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Meaning	*Error*	W2	W1	W0	G2	G1	G0	U/B

The method by which the B-sensor module inputs can now be read out depends on the *transmission-type* of TPDO4, which can be set in *OD* index 1803h, subindex 2 of the MDT-DCS module. The method options are identical to what has been described for the read-out of the T-sensors in document "MDT-DCS CANopen Module, user manual and reference".

Individual B-sensor module channels (there are 7 per module) can be read out using CANopen **SDO** messages by reading from *OD* index 4200h to 4204h (see *OD* tables for a description of the individual channels).

For additional details about the B-sensor module read-out see above-mentioned document.

3 Coil-ADC Read-out

By writing to *OD* index 4E00h up to three Coil-ADCs can be selected. The default is: no Coil-ADCs, i.e. Object 4E00h has value 0 (zero).

The ELMB module sends one PDO message containing 5 bytes for each Coil-ADC input and per Coil-ADC one input is read (hardcoded for the time being: the ADC's first physical input channel *AIN1*). The CAN-identifier used for this PDO is called *5th-transmit-PDO* (**TPDO5**) and here has a fixed COB-ID of 500h+NodeID.

The MDT-DCS module produces the following 5-databyte TPDO5:

MDT-DCS node → Host

COB-ID	DataByte 0	DataByte 1	DataByte 2-4
500h + *NodeID*	Channel number	ADC-config	24-bit ADC value

with:

ADC value: signed/unsigned (?) 24-bits ADC value, LSB in byte 2, MSB in byte 4.

Channel number: Number between 0 and 2.
Chan 0: Coil-ADC #0,
Chan 1: Coil-ADC #1,
Chan 2: Coil-ADC #2.

ADC-config: **bit 7**: Conversion status: 1=ERROR (overflow occurred during ADC conversion), 0=OKAY.
bits 6-0: ADC configuration: conversion wordrate (bits W0, W1, W2 and W3) and gain range (bits G0, G1 and G2); see below; unipolar or bipolar mode bit is left out (read it from OD if necessary).
For definitions see *OD* index 4F00h/4F01h/4F02h, sub 2,3,4,5,6 and 7.

BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Meaning	*Error*	G2	G1	G0	W3	W2	W1	W0

The method by which all three Coil-ADC module inputs can now be read out depends on the *transmission-type* of TPDO5, which can be set in *OD* index 1804h, subindex 2 of the ELMB module. The method options are identical to what has been described for the read-out of the T-sensors in the MDT-DCS user manual and reference.

Individual Coil-ADC channels (there are 4 channels per ADC) can be read out using CANopen **SDO** messages by reading from *OD* index 4400h, 4401h or 4402h.

Note: the ADC clock used is 5.0 MHz instead of the 4.9152 MHz required to get the conversion sample rates mentioned in the ADC datasheet. This means the Coil-ADC sample rates listed have to be multiplied by a factor of 5.0000 / 4.9152.

4 ELMB Interface

This section describes how the ELMB board interfaces hardware-wise to the rest of the B-module calibration system. It is shown here for reference only.
Table 2 shows the I/O-pin configuration of the ATmega128 microcontroller on the ELMB:

- **B_*xxx*** is the SPI interface including five chip-select (**B_CS*x***) lines to up to 5 B-sensor modules in the B-module calibration test setup; **B_CS*0*** selects the calibrated B-module; includes 4 connections to B-module identifier chips (**B_IDx**) on the B-modules being calibrated.
- **COIL_*xxx*** is the SPI interface to the coil ADCs of which there are 3; only chip-selects are shown; other SPI-interface lines are shared with B-sensor modules (**B_*xxx***).
- **DIGIO*x*** are digital in- and outputs (*not used in the B-module calibration app code*).

I/O PORT: **Function:**	**A** **In/Out**	**B** **In/Out**	**C** **In/Out**	**D** **In/Out**	**E** **In/Out**	**F** **I/O/ADC**
pin 0	*DIGIO0*	**x**	*B_CS1*	**x**	**x**	*B_ID1*
pin 1	*DIGIO1*	*SCLK*	*B_CS2*	**x**	**x**	*B_ID2*
pin 2	*DIGIO2*	*SDI*	*B_CS3*	**x**	**x**	*B_ID3*
pin 3	*DIGIO3*	*SDO*		*B_SDI*	*B_SCLK*	*B_ID4*
pin 4	*DIGIO4*	**x**	*B_CS4*			
pin 5	*DIGIO5*	**x**	*B_CS0*			
pin 6	*DIGIO6*	**x**	*COIL_CS0*			*COIL_CS2*
pin 7	*DIGIO7*	**x**	*COIL_CS1*			*B_SDO*

Table 2. I/O-pin configuration of the ELMB microcontroller (ATmega128) on the B-sensor calibration module:
x = *NOT* available externally (used internally by ELMB).
SCLK/SDI/SDO = lines carrying **SPI**-protocol for the on-board CAN-controller.
(see text for explanation of other signals).

5 ELMB B-sensor Calibration Object Dictionary

The values of objects marked with ∗ in the *Index* column can be stored permanently in EEPROM. They are retrieved from EEPROM at reset and power-up.

Communication Profile Area (ELMB *B-calibration*)						
Index (hex)	**Sub Index**	**Description**	**Data/ Object**	**Attr**	**Default**	**Comment**
1000	-	Device type	U32	RO	00070191h	Meaning: DSP-401 device profile, analogue inputs, digital in- and outputs on device
1001	-	Error register	U8	RO	0	
1002	-	Manufacturer status reg	U32	RO	0	[1] (see footnote)
1008	-	Manufacturer device name	VisStr	RO	"ELMB"	= Embedded Local Monitor Board
1009	-	Manufacturer hw version	VisStr	RO	"el40"	= ELMB v4
100A	-	Manufacturer software version	VisStr	RO	"MB30"	***MDT-DCS B-calibration*** application v3.0 (*minor-version* number in 100A,1)
100C	-	Guard time [ms]	U16	RO	1000	= 1 second
100D *	-	Life time factor	U8	RW	0	lifeguarding timeout in seconds; 0 → no lifeguarding timeout
1010		Store parameters	Array			Save stuff in onboard EEPROM
	0	Highest index supported	U8	RO	3	
	1	Save all parameters	U32	RW	1	Read: 1; Write "save": store all
	2	Save communication parameters	U32	RW	1	Read: 1; Write "save": store PDO par's, Life time factor, …
	3	Save application par's	U32	RW	1	Read: 1; Write "save": store ADC config, dig.I/O config, …
1011		Restore default parameters	Array			Invalidate stuff in onboard EEPROM; use defaults
	0	Highest index supported	U8	RO	3	
	1	Restore all parameters	U32	RW	1	Read: 1; Write "load": invalidate all parameters stored
	2	Restore communication parameters	U32	RW	1	Read: 1; Write "load": invalidate stored PDO par's, etc.
	3	Restore application par's	U32	RW	1	Read: 1; Write "load": invalidate stored ADC config, etc.
1017 *	-	Producer Heartbeat Time [1 s]	U16	RO	0	In units of seconds (but <=255 !), (NB: should be in ms according to CANopen!); 0 → Heartbeat is disabled
1018		Identity	Record			Mandatory CANopen object
	0	Number of entries	1..4	RO	1	
	1	Vendor ID	U32	RO	12345678h	*to be ordered from CiA*

[1] Manufacturer Status Register: byte0 = **NTC-ADC**, byte1 = **B1-ADC**, byte2 = **B2-ADC**, byte3 = **CSM-ADC**; status byte: **01**: ADC reset error, **02**: ADC calibration error, **04**: ADC conversion time-out, **FF**: ADC absent / not used.

Communication Profile Area (*ELMB B-calibration*) *(continued…)*						
Index (hex)	**Sub Index**	**Description**	**Data/ Object**	**Attr**	**Default**	**Comment**
1400		1st Receive PDO par's	Record			Data type = PDOCommPar
	0	Number of entries	U8	RO	5	
	1	COB-ID used by PDO	U32	RO	200h + *NodeID*	According to CANopen Predefined Connection Set
	2	Transmission type	U8	RO	255	Only 255 allowed
	3,5	*Not used*		RO	0	
1600		1st Receive PDO mapping	Record			Data type = PDOMapping
	0	Number of entries	U8	RO	1	
	1	Digital outputs 1-8	U32	RO	62000108	OD-index 6200, sub-index 1: Outputs 1-8 (see DSP-401), size = 8 bits
1800		1st Transmit PDO par's	Record			Data type = PDOCommPar
	0	Number of entries	U8	RO	5	
	1	COB-ID used by PDO	U32	RO	180h + *NodeID*	According to CANopen Predefined Connection Set
*	2	Transmission type	U8	RW	255	Only 1 and 255 allowed
	3	Inhibit time [100 µs]	U16	RO	0	*not used*
*	5	Event timer [1 ms]	U16	RW	0	Truncated to multiples of 1000; active if transm-type = 255
1801		2nd Transmit PDO par's	Record			Data type = PDOCommPar
	0	Number of entries	U8	RO	5	
	1	COB-ID used by PDO	U32	RO	280h + *NodeID*	According to CANopen Predefined Connection Set
*	2	Transmission type	U8	RW	1	Only 1 and 255 allowed
	3	Inhibit time [100 µs]	U16	RO	0	*not used*
*	5	Event timer [1 ms]	U16	RW	0	Truncated to multiples of 1000; active if transm-type = 255
1802		3rd Transmit PDO par's	Record			Data type = PDOCommPar
	0	Number of entries	U8	RO	5	
	1	COB-ID used by PDO	U32	RO	380h + *NodeID*	According to CANopen Predefined Connection Set
*	2	Transmission type	U8	RW	1	Only 1 and 255 allowed
	3	Inhibit time [100 µs]	U16	RO	0	*not used*
*	5	Event timer [1 ms]	U16	RW	0	Truncated to multiples of 1000; active if transm-type = 255
1803		4th Transmit PDO par's	Record			Data type = PDOCommPar
	0	Number of entries	U8	RO	5	
	1	COB-ID used by PDO	U32	RO	480h + *NodeID*	According to CANopen Predefined Connection Set
*	2	Transmission type	U8	RW	1	Only 1 and 255 allowed
	3	Inhibit time [100 µs]	U16	RO	0	*not used*
*	5	Event timer [1 ms]	U16	RW	0	Truncated to multiples of 1000; active if transm-type = 255

Communication Profile Area (*ELMB B-calibration*) *(continued…)*						
Index (hex)	**Sub Index**	**Description**	**Data/ Object**	**Attr**	**Default (*hex*)**	**Comment**
1804		5th Transmit PDO par's	Record			Data type = PDOCommPar
	0	Number of entries	U8	RO	5	
	1	COB-ID used by PDO	U32	RO	500h + *NodeID*	
*	2	Transmission type	U8	RW	1	Only 1 and 255 allowed
	3	Inhibit time [100 µs]	U16	RO	0	*not used*
*	5	Event timer [1 ms]	U16	RW	0	Truncated to multiples of 1000; active if transm-type = 255
1A00		1st Transmit PDO mapping	Record			Data type = PDOMapping
	0	Number of entries	U8	RO	1	
	1	Digital inputs 1-8	U32	RO	60000108	OD-index 6000, sub-index 1: Inputs 1-8 (see DSP-401), size = 8 bits
1A01		2nd Transmit PDO mapping	Record			Data type = PDOMapping
	0	Number of entries	U8	RO	2	*should be 255 for MuxPDO, but this is not a CANopen MPDO…*
	1	NTC number	U32	RO	40000008	*actually not allowed, but…*
	2	24-bit analogue input + stat	U32	RO	40000*x*18	OD-index 4000, sub-index *x*: Analogue inputs, multiplexed, size = 24 bits
1A02		3rd Transmit PDO mapping	Record			Data type = PDOMapping
	0	Number of entries	U8	RO	2	*should be 255 for MuxPDO, but this is not a CANopen MPDO…*
	1	CSM ADC channel no	U32	RO	41000008	*actually not allowed, but…*
	2	24-bit analogue input + stat	U32	RO	41000*x*18	OD-index 4100, sub-index *x*: Analogue inputs, multiplexed, size = 24 bits
1A03		4th Transmit PDO mapping	Record			Data type = PDOMapping
	0	Number of entries	U8	RO	2	*should be 255 for MuxPDO, but this is not a CANopen MPDO…*
	1	B-sensor ADC channel number	U32	RO	42000008	*actually not allowed, but…*
	2	24-bit analogue input	U32	RO	420*x*0*x*24	OD-index 4200/4201,subindex *x*, Analogue inputs, multiplexed, size = 32 bits
1A04		5th Transmit PDO mapping	Record			Data type = PDOMapping
	0	Number of entries	U8	RO	2	*should be 255 for MuxPDO, but this is not a CANopen MPDO…*
	1	Coil ADC channel number	U32	RO	44000008	*actually not allowed, but…*
	2	24-bit analogue input	U32	RO	440*x*0*x*24	OD-index 4400/4401,subindex *x*, Analogue inputs, multiplexed, size = 32 bits

Manufacturer-specific Profile Area (*ELMB B-calibration*) *(continued…)*						
Index (hex)	**Sub Index**	**Description**	**Data/ Object**	**Attr**	**Default**	**Comment**
2500		B-sensor #0 ADC-config	Record			CRYSTAL CS5524 24-bit ADC [1]
	0	Number of entries	U8	RO	21	
	1	Number of input channels	U8	RO	7	
*	2	Conversion Word Rate Hall/Temp	U8	RW	0	3-bit code [2] (same as Object 2500, sub 5)
*	3	Input Voltage Range Hall	U8	RW	0	3-bit code [3]
*	4	Unipolar/Bipolar Measurement Mode Hall	U8	RW	0	0 = bipolar, 1 = unipolar
*	5	Conversion Word Rate Hall/Temp	U8	RW	0	3-bit code [2] (same as Object 2500, sub 2)
*	6	Input Voltage Range Temp	U8	RW	5	3-bit code [3]
*	7	Unipolar/Bipolar Measurement Mode Temp	U8	RW	1	0 = bipolar, 1 = unipolar
	8	Power Save Mode	Bool	WO		1 = power save
	9	Configuration Register	U32	RW		CS5523 Config Register
	10	Offset Register #1	U32	RW		CS5523 physical channel AIN1
	11	Gain Register #1	U32	RW		CS5523 physical channel AIN1
	12	Offset Register #2	U32	RW		CS5523 physical channel AIN2
	13	Gain Register #2	U32	RW		CS5523 physical channel AIN2
	14	Offset Register #3	U32	RW		CS5523 physical channel AIN3
	15	Gain Register #3	U32	RW		CS5523 physical channel AIN3
	16	Offset Register #4	U32	RW		CS5523 physical channel AIN4
	17	Gain Register #4	U32	RW		CS5523 physical channel AIN4
	18	Channel-Setup Register #1	U32	RW		LC 1 (12-bits) in lower 2 bytes, LC 2 (12-bits) in upper 2 bytes
	19	Channel-Setup Register #2	U32	RW		LC 3 (12-bits) in lower 2 bytes, LC 4 (12-bits) in upper 2 bytes
	20	Channel-Setup Register #3	U32	RW		LC 5 (12-bits) in lower 2 bytes, LC 6 (12-bits) in upper 2 bytes
	21	Channel-Setup Register #4	U32	RW		LC 7 (12-bits) in lower 2 bytes, LC 8 (12-bits) in upper 2 bytes
2501		B-sensor #1 ADC-config	Record			
2502		B-sensor #2 ADC-config	Record			
2503		B-sensor #3 ADC-config	Record			
2504		B-sensor #4 ADC-config	Record			

[1] Subindex 2-7 are common to all B-sensor modules ! (and subindex 2 and 5 are identical).

[2] **000**: 15.0 Hz, **001**: 30.0 Hz, **010**: 61.6 Hz, **011**: 84.5 Hz,
100: 101.1 Hz, **101**: 1.88Hz, **110**: 3.76 Hz, **111**: 7.51 Hz

[3] **000**: 100 mV, **001**: 55 mV, **010**: 25 mV, **011**: 1 V, **100**: 5 V, **101**: 2.5 V

Manufacturer-specific Profile Area (*ELMB B-calibration*) *(continued…)*						
Index (hex)	**Sub Index**	**Description**	**Data/ Object**	**Attr**	**Default**	**Comment**
2600	-	ADC-reset-and-calibrate B-sensor #0	U8	WO		Writing any value triggers a reset and calibration sequence on B-sensor #0 with its current ADC settings
2601	-	ADC-reset-and-calibrate B-sensor #1	U8	WO		idem for B-sensor #1
2602	-	ADC-reset-and-calibrate B-sensor #2	U8	WO		idem for B-sensor #1
2603	-	ADC-reset-and-calibrate B-sensor #3	U8	WO		idem for B-sensor #1
2604	-	ADC-reset-and-calibrate B-sensor #4	U8	WO		idem for B-sensor #1
2700 *	-	ADC-reset-and-calibrate before each channel scan all B-sensors	U8	RW	0	If =1 a reset/calibration sequence is performed before every B-sensor ADC input channel scan
2800 *	-	Number of B-sensors present	U8	RW	0	Must be <= 5; if >0 B-sensors must be installed consecutively
2801 *	-	B-sensor #0 ('calibrated') microcontroller	Bool	RW	0	B-sensor module #0 is of the microcontroller type
2802 *	-	B-sensor #1 to #4 microcontroller	Bool	RW	0	B-sensor module #1 to #4 are of the microcontroller type
2803	-	B-sensor microcontroller	Bool	WO		set all B-sensor modules to the same type (with/without micro)
2900		B-sensor #0 identification	Record			DS2401 Identification chip: unique 8-byte serial number (on B-sensor modules without microcontroller); If B-sensor is of microcontroller type: the SPI address (1 byte significant) is given by reading subindex 1
	0	Number of entries	U8	RO	2	
	1	First 4 bytes	U32	RO		Read this one always first
	2	Second 4 bytes	U32	RO		Read this one only after having read first 4 bytes
2901		B-sensor #1 identification	Record			
2902		B-sensor #2 identification	Record			
2903		B-sensor #3 identification	Record			
2904		B-sensor #4 identification	Record			

Manufacturer-specific Profile Area (*ELMB B-calibration*) *(continued…)*						
Index (hex)	**Sub Index**	**Description**	**Data/ Object**	**Attr**	**Default**	**Comment**
2E00 *	-	Digital Input debounce timer	U8	RW	10	In units of ca. 400 µs (set to 0 there is ca. 400 µs between consecutive input polls).
2F00 *	-	Digital Output Init High	U8	RW	0	After a hard reset: if set to 0 Digital Outputs will be initialised to all low; if set to 1 Digital Outputs will be initialised to all high
3000		Calculate 16-bit CRC	Array			
	0	Number of entries	U8	RO	2	
	1	CRC of program code in FLASH memory	U16	RO	0	SDO reply unequal to zero means there is a checksum error; absence of CRC results in SDO *Abort* with *Error Code* 1; error while accessing FLASH results in SDO *Abort* with *Error Code* 6.
3100	-	ELMB Serial Number	U32	RW		Number or 4-byte string uniquely identifying an ELMB, given during production.
3101	-	Enable ELMB Serial Number write operation	U8	WO	**EXPERT ONLY**	Writing 0x5A enables one write operation on the Serial Number (Object 3100).
3200		CAN-controller settings	Array			
	0	Number of entries	U8	RO	1	
*	1	Disable Remote Frames	U16	RO	0	[1]

[1] Due to the way the ELMB's CAN-controller handles Remote Frames, it is recommended to disable Remote Frames permanently if not needed (for PDO read-out). A special provision in the software has been made to ensure that the Node Guard Remote Frame is still handled properly.

Manufacturer-specific Profile Area (*MDT-DCS B-calibration*) *(continued…)*						
Index (hex)	**Sub Index**	**Description**	**Data/ Object**	**Attr**	**Default**	**Comment**
4200		Read analogue input B-sensor #0	Record			8 bits status, 24 bits analogue value (B-sensor #0)
	0	Number of entries	U8	RO	7	Fixed value (see OD-index 2400, subindex 1)
	1	Input 1 (B-sensor ADC #0)	I24	RO		1st analog input:24-bit (Hall H1)
	2	Input 2 (B-sensor ADC #0)	I24	RO		2nd " " " (Hall H2)
	3	Input 3 (B-sensor ADC #0)	I24	RO		3rd " " " (Hall H3)
	4	Input 4 (B-sensor ADC #0)	I24	RO		4th " " (fullscale Hall)
	5	Input 5 (B-sensor ADC #0)	I24	RO		5th " " " (NTC)
	6	Input 6 (B-sensor ADC #0)	I24	RO		6th " " " (0ºC calib)
	7	Input 7 (B-sensor ADC #0)	I24	RO		7th " " " (100ºC calib)
4201		Read analogue input B-sensor #1	Record			8 bits status, 24 bits analogue value (B-sensor #1)
4202		Read analogue input B-sensor #2	Record			8 bits status, 24 bits analogue value (B-sensor #2)
4203		Read analogue input B-sensor #3	Record			8 bits status, 24 bits analogue value (B-sensor #3)
4204		Read analogue input B-sensor #4	Record			8 bits status, 24 bits analogue value (B-sensor #4)
4C00 *	-	Continuous B-sensor/Coil-ADC measurement cycles	U8	RW	0	If =1 measurement cycles are performed continuously after the command to start has been received, until the command to stop is received (see text for details)
4D00		Read analogue input COIL-ADC #0	Record			8 bits status, 24 bits analogue value (Coil #0)
	0	Number of entries	U8	RO	4	Fixed value (see OD-index 4F00, subindex 1)
	1	Input 1 (Coil ADC #0)	I32	RO		1st analog input:32-bit
	2	Input 2 (Coil ADC #0)	I32	RO		2nd " " "
	3	Input 3 (Coil ADC #0)	I32	RO		3rd " " "
	4	Input 4 (Coil ADC #0)	I32	RO		4th " " "
4D01		Read analogue input Coil-ADC #1	Record			8 bits status, 24 bits analogue value (Coil #1)
4D02		Read analogue input Coil-ADC #2	Record			8 bits status, 24 bits analogue value (Coil #2)

Manufacturer-specific Profile Area (*MDT-DCS B-calibration*) *(continued…)*						
Index (hex)	**Sub Index**	**Description**	**Data/ Object**	**Attr**	**Default**	**Comment**
4E00 *	-	Number of Coil-ADCs present	U8	RW	0	Must be <= 3; if >0 Coil-ADCs must be installed consecutively
4E01 *	-	Generate SYNC in Coil-ADC readout loop	Bool	RW	1	If =1 the ELMB generates/sends a SYNC message immediately after 'conversion-ready' is detected of the first Coil-ADC (after a next conversion of the B-sensor ADCs is initiated (in continuous mode) and before the Coil-ADC data is read out)
4E02	-	Physical input channel number of Coil-ADCs in readout loop	U8	RW	1	1 <= value <= 4 Goes back to default at every reset
4F00		ADC-configuration Coil #0	Record			CRYSTAL CS5534 24-bit ADC [1]
	0	Number of entries	U8	RO	18	
	1	Number of input channels	U8	RO	4	
*	2	Conversion Word Rate	U8	RW	4	3-bit code [2]
*	3	Input Gain	U8	RW	0	3-bit code [3]
*	4	Unipolar/Bipolar Measurement Mode	U8	RW	0	0 = bipolar, 1 = unipolar
	5	Power Save Mode	Bool	WO		1 = power save
	6	Configuration Register	U32	RW		CS5534 Config Register
	7	Offset Register #1	U32	RW		CS5534 physical channel AIN1
	8	Gain Register #1	U32	RW		CS5534 physical channel AIN1
	9	Offset Register #2	U32	RW		CS5534 physical channel AIN2
	10	Gain Register #2	U32	RW		CS5534 physical channel AIN2
	11	Offset Register #3	U32	RW		CS5534 physical channel AIN3
	12	Gain Register #3	U32	RW		CS5534 physical channel AIN3
	13	Offset Register #4	U32	RW		CS5534 physical channel AIN4
	14	Gain Register #4	U32	RW		CS5534 physical channel AIN4
	15	Channel-Setup Register #1	U32	RW		LC 1 (16-bits) in lower 2 bytes, LC 2 (16-bits) in upper 2 bytes
	16	Channel-Setup Register #2	U32	RW		LC 3 (16-bits) in lower 2 bytes, LC 4 (16-bits) in upper 2 bytes
	17	Channel-Setup Register #3	U32	RW		LC 5 (16-bits) in lower 2 bytes, LC 6 (16-bits) in upper 2 bytes
	18	Channel-Setup Register #4	U32	RW		LC 7 (16-bits) in lower 2 bytes, LC 8 (16-bits) in upper 2 bytes
4F01		ADC-configuration Coil #1	Record			CRYSTAL CS5534 24-bit ADC
4F02		ADC-configuration Coil #2	Record			CRYSTAL CS5534 24-bit ADC

[1] Subindex 2-7 are common for all Coil-ADCs !

[2] **0000**: 120 Sps, **0001**: 60 Sps, **0010**: 30 Sps, **0011**: 15 Sps, **0100**: 7.5 Sps, **1000**: 3840 Sps, **1001**: 1920 Sps, **1010**: 960 Sps, **1011**: 480 Sps, **0100**: 240 Sps

[3] **000**: 1x, **001**: 2x, **010**: 4x, **011**: 8x, **100**: 16x, **101**: 32x, **110**: 64x

Manufacturer-specific Profile Area (*MDT-DCS B-calibration*) *(continued…)*						
Index (hex)	**Sub Index**	**Description**	**Data/ Object**	**Attr**	**Default**	**Comment**
5C00	-	Compile Options	U32	RO		Bitmask denoting which compile options were used when the application was generated (see table below for details)
5E00	-	Transfer control to Bootloader	U8	WO		

Object 5C00: Compile Options		
Bit	***Compile Option***	***Comment***
0	–	–
1	–	–
2	–	–
3	–	–
4	–	–
5	7BIT_NODEID	Only DIP-switch 1 used for CAN baudrate (125 or 250 kbaud); other 7 switches used for setting the Node-ID: 1-127 (when this option is not set a 6-bit Node-ID is used and 2 bits are used for selecting a baudrate)
6	RS232	Include stuff to be able to use 'printf()' and such; requires the *Programmer* or other RS232 adapter to be connected to the ELMB programmer connector
7	ELMB103	The ELMB is an ELMB103 type (with ATmega103 processor); by default an ELMB128 (with ATmega128 processor) is assumed
8	VARS_IN_EEPROM	Store/retrieve working copies of configuration parameters in/from EEPROM
9	–	–
10	–	–
11	–	–
12	CAN_REFRESH	Refresh CAN-controller descriptor register (at each buffer write/read)
13	2313_SLAVE_PRESENT	there is (probably) a Slave processor (usually when using an ELMB103, so in combination with compile option *ELMB103* shown above); this includes the code that deals with the Slave processor
14	SPI_SLOWER	Signal hold time at least ca. 0.5 µs longer (extra on top of ca. 0.5 µs hold time)
15	–	–

Table 3. Optional compiler macro defines (individual options are preceeded and ended by a double underscore '__').

Standardised Device Profile Area (*MDT-DCS B-calibration*)						
Index (hex)	**Sub Index**	**Description**	**Data/ Object**	**Attr**	**Default**	**Comment**
6000		Read state 8 input lines	Array			
	0	Number of 8-bit inputs	U8	RO	1	
	1	Read inputs 1-8	U8	RO		ELMB ATmega128 PORTA (Port shared with Object 6200,1)
6005 *	-	Global Digital Input Interrupt Enable	Bool	RW	0	Enables/disables change-of-state TPDO1 transmission
6006		Interrupt Mask Any Change 8 input lines	Array			
	0	Number of 8-bit inputs	U8	RO	1	
	1	Interrupt Mask Inputs 1-8	U8	RW	FFh	
6200		Write state 8 output lines	Array			
	0	Number of 8-bit outputs	U8	RO	1	
	1	Write outputs 1-8	U8	RW		ELMB ATmega128 PORTA (Port shared with Object 6000,1)
6208		Filter mask 8 output lines	Array			
	0	Number of 8-bit masks	U8	RO	1	
*	1	Filter mask outputs 1-8	U8	RW	F0h	maskbit=1: output; pins not defined as outputs are inputs, to be accessed thru Object 6000, 1

6 Emergency Objects

CANopen *Emergency* messages are triggered by the occurrence of an internal (fatal) error situation. An *Emergency* CAN-message has the following general syntax:

MDT-DCS → Host

COB-ID	Byte 0-1	Byte 2	Byte 3-7
080h + *NodeID*	Emergency Error Code	Error Register (Object 0x1001)	Manufacturer specific error field

The following Emergency messages can be generated by the **MDT-DCS** application:

Error Description	Emergency Error Code (byte 0-1; hex)	Manufacturer-specific Error Field (byte 3-7)
CAN communication	8100	Byte 3: 81C91 Interrupt Register content Byte 4: 81C91 Mode/Status Register content Byte 5: error counter
Life Guarding	8130	(CAN-controller has been reinitialized)
RPDO: too few bytes	8210	Byte 3: minimum DLC (Data Length Code)
CRC error	5000	Byte 3: 30 Byte 4: 1 (program FLASH)
EEPROM: write error	5000	Byte 3: 41 Byte 4: Parameter block index [1] Byte 5: 0 : writing block info > 0: size of parameter block to write
EEPROM: read error	5000	Byte 3: 42 Byte 4: Parameter block index [1] Byte 5: Error id (1=CRC, 2=length, 4=infoblock)
B-sensor ADC: conversion timeout	5000	Byte 3: 51 Byte 4: B-sensor number (0..4) Byte 5: ADC channel number (0..7)
B-sensor ADC: reset failed	5000	Byte 3: 52 Byte 4: B-sensor number (0..4) Byte 5: Error id [2]
B-sensor ADC: offset calibration failed	5000	Byte 3: 53 Byte 4: B-sensor number (0..4)
B-sensor ADC: gain calibration failed	5000	Byte 3: 54 Byte 4: B-sensor number (0..4)
B-sensor ADC problem(s) during initialisation (check OD 1002)	5000	Byte 3: 55 Byte 4: ADC 0/1 status (see OD index 1002) Byte 5: ADC 2/3 status (see OD index 1002) Byte 6: ADC 4 status (see OD index 1002)

…table continues on the next page…

[1] **0**: PDO communication parameters, **1**: Guarding parameters, **2**: Digital I/O configuration, **3**: NTC ADC configuration, **4**: B-sensor ADC configuration, **5**: CSM ADC configuration, **6**: CAN configuration parameters, **7**: COIL-ADC configuration, **0xFE**: Calibration constant(s), **0xFF**: ELMB Serial Number.

[2] **01**: Reset-Valid bit not set, **02**: Reset-Valid bit not reset, **04**: error in Offset Register value, **08**: error in Gain Register value

Error Description	Emergency Error Code (byte 0-1; hex)	Manufacturer-specific Error Field (byte 3-7)
COIL-ADC: conversion timeout	5000	Byte 3: 71 Byte 4: COIL-ADC number (0..2) Byte 5: ADC channel number (0..3)
COIL_ADC: reset failed	5000	Byte 3: 72 Byte 4: COIL-ADC number (0..2) Byte 5: Error id [1]
COIL-ADC: offset calibration failed	5000	Byte 3: 73 Byte 4: COIL-ADC number (0..2)
COIL-ADC: gain calibration failed	5000	Byte 3: 74 Byte 4: COIL-ADC number (0..2)
COIL-ADC problem(s) during initialisation	5000	Byte 3: 75 Byte 4: ADC 0 status Byte 5: ADC 1 status Byte 6: ADC 2 status
Irregular reset (Watchdog, Brown-out or JTAG)	5000	Byte 3: F0 Byte 4: microcontroller MCUCSR register contents [2]
Bootloader: not present	5000	Byte 3: F1
Bootloader is now in control [3]	5000	Byte 3: FE Byte 4: 01 Byte 5: 28 Byte 6: microcontroller MCUCSR register contents [2] Byte 7: 00
Bootloader cannot jump to application: invalid [3]	6000	Byte 3: FE Byte 4: AA Byte 5: AA

Byte 2 of the *Emergency* message contains the value of the socalled *Error Register* (Object Dictionary index 0x1001, a mandatory CANopen object). One or more bits of the 8-bit Error Register can be set to 1, depending on the node's history of errors since the last reset. The table below gives a description of the different bits.

Error Register (Object 0x1001) bits	
Bit	**Error type**
0	generic
1	current
2	voltage
3	temperature
4	communication
5	device profile specific
6	*reserved (=0)*
7	manufacturer specific

[1] **01**: Reset-Valid bit not set, **02**: Reset-Valid bit not reset, **04**: error in Offset Register value, **08**: error in Gain Register value

[2] ATmega128 *MCUCSR* register bits: **0x01**: Power-On Reset, **0x02**: External Reset, **0x04**: Brown-Out Reset, **0x08**: Watchdog Reset, **0x10**: JTAG Reset, **0x80**: JTAG Interface Disable

[3] This Emergency message is sent by the Bootloader.