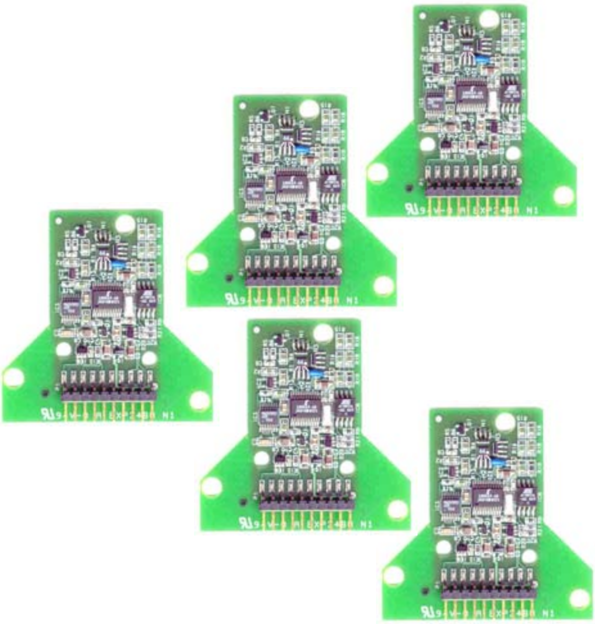

BsCAN

CANopen B-sensor readout system

user manual, version 0.4

29 May 2001

Henk Boterenbrood
(*boterenbrood@nikhef.nl*)

NIKHEF

Table of Contents

1 Introduction

The ***BsCAN*** *CANopen* application is a custom-made solution for reading out multiple 'addressable B-sensor modules' (described in [1]), using the CAN-bus and the *CANopen* protocol ([2]) for communication with a host system.

BsCAN is implemented on a **CRYSTAL-CAN-V2** box (which has been slightly modified for this application), a general-purpose microcontroller module with CAN-interface and eight **SPI** (*Serial Peripheral Interface*) connectors for linking up to external I/O.

The **CRYSTAL-CAN-V2** module is built around a 16-MHz Philips **80C592** 8-bit microcontroller with on-chip CAN controller. It provides 48 kByte of user program memory and 63.5 kByte of user RAM [1]. Program code (in standard Intel Hex format) can be downloaded directly via the RS232 port (which is accessible only after opening the box).

The ***BsCAN*** box can have up to 8 ***strings*** of B-sensor modules connected to its SPI connectors. A *string* is one (10-wire flat-)cable with one or more B-sensor modules. A special adapter module is needed between the B-sensor cable and the CRYSTAL-CAN's SPI-connector. It provides the proper connector conversion and signal buffering.
The number of B-sensor modules connected to one string and to one ***BsCAN*** box is only limited by electrical conditions and in practice means that up to about 10 B-sensor modules can be connected to one string, and a total of up to 30 to 40 B-sensor modules can be connected to one ***BsCAN*** box. A string can have a length of up to about 15 meter (*preliminary information*; tested and working is a cable of 10 meters with 10 B-sensor modules).
See Figure 1 for an example of a ***BsCAN*** system configuration.

A requirement is that the B-sensor modules connected to one ***BsCAN*** box must have a unique address. The address of a B-sensor is printed on a sticker on the board and is a number that lies between 0 and 127. This address is stored on-board in non-volatile memory (for details see [1]).

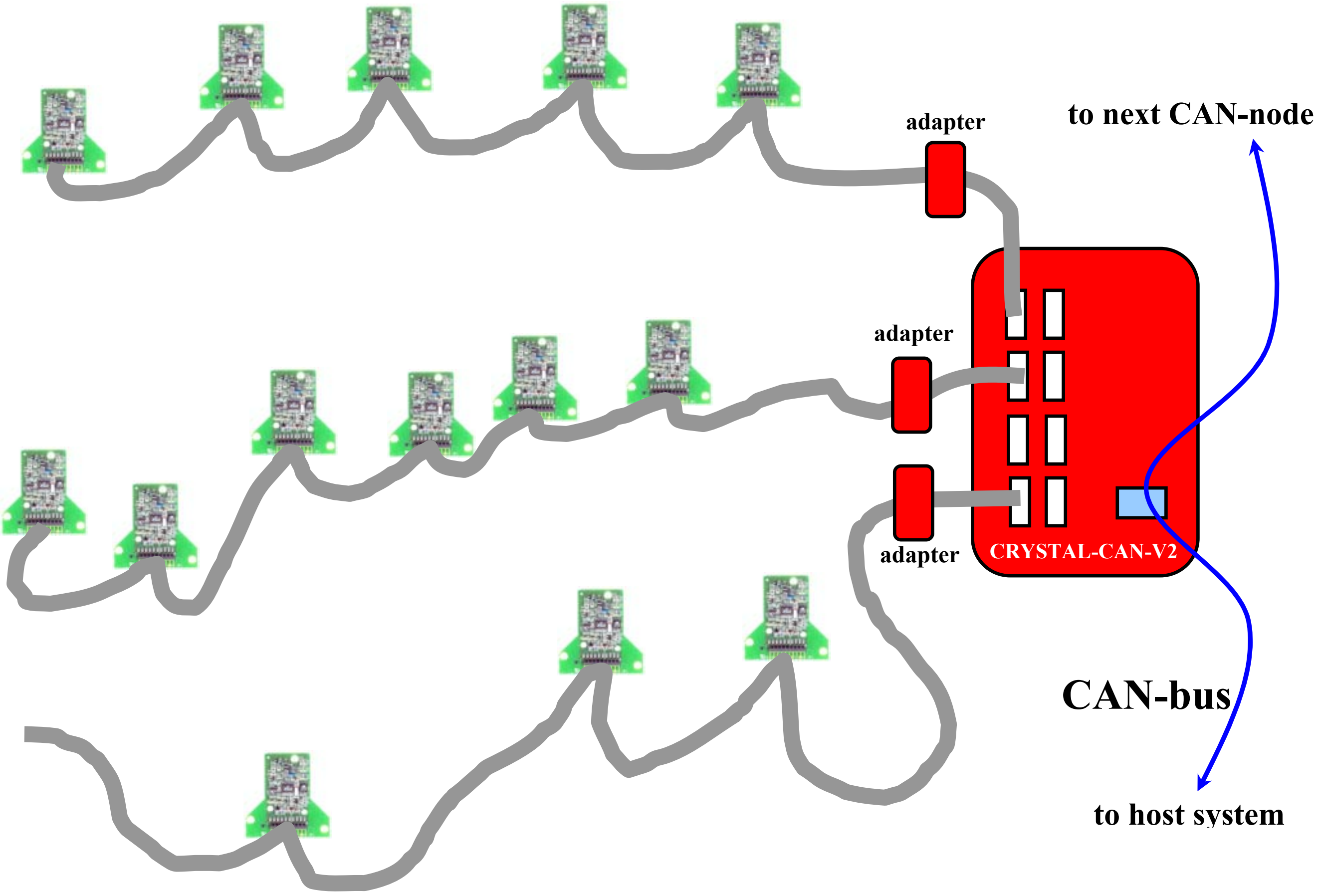

Figure 1. Schematic picture of a ***BsCAN*** system consisting of 3 *strings* and a number of B-sensor modules.

A B-sensor module can be added to the system onto any string and a string can be connected to any of the 8 CRYSTAL-CAN-V2 SPI connectors. However, after adding B-sensors or strings to a ***BsCAN*** box (or moving a B-sensor module from one string to another) it is necessary to initiate a *configure* operation on the ***BsCAN*** box. During this operation the ***BsCAN*** box will determine automatically where and which B-sensor modules are connected to it, subsequently store the new configuration in its onboard non-volatile memory and finally reset and calibrate all B-sensor modules (for details see [1]). At every subsequent power-up or reset the system assumes this configuration is the current configuration (and repeat the reset and calibrate operation on the B-sensor modules). The user will be informed about any B-sensor modules missing (or not functioning) or having changed string position. Added B-sensors are ignored and are not read out until after the next *configure* operation.

As seen from a controlling host system, B-sensor modules are accessed on the basis of their unique address and CAN-node module-id only, their location on a string is irrelevant.
All information about a ***BsCAN***'s current configuration can be read from a number of entries in its Object Dictionary (see section 3).

2 Operation

2.1 Initialisation

Important:
- When powering up the system it should be switched off and immediately after switched on again: due to a ***hardware bug*** in the B-sensor microcontroller (Atmel AT90LS2343) the program in the microcontroller does not start up at the first power-up: it needs to be powered off and on for the on-chip RC oscillator to start running !

After power-up, watchdog reset, manual reset or *CANopen* initiated reset actions a *CANopen* node sends a so-called ***Bootup*** message (a *N*etwork ***M****anagemen****T*** (NMT) message defined in the *CANopen* standard) as soon as it has finished initialising (which takes a couple of seconds); this is a CAN-message with the following syntax:

BsCAN **(NMT-Slave) ⟶ Host (NMT-Master)**

COB-ID	Byte 0
0x700 + *NodeID*	0

The *COB-ID* is the *Communication Object Identifier*, the 11-bit CAN-message identifier defining one unit of transportation in a CAN network.
The *NodeID* is the CAN node identifier set by means of the CRYSTAL-CAN-V2 switches (accessible on the frontpanel), which must be in the range between 1 and 127.

To simplify matters, after power-up or reset the ***BsCAN*** node automatically goes into *Operational* state, and the host application can immediately start reading out the B-sensor modules, by sending *CANopen* ***SYNC*** messages (see section 2.3 for details).

Note that the *CANopen* standard defines that after power-up/reset a node goes into *Preoperational* state. Before any input channels can be read using the ***PDO*** mechanism, the connected *CANopen*-nodes have to be set into the *Operational* state. If for any reason the ***BsCAN*** node has been set into another state (see [2]) other than *Operational*, it can be set into the *Operational* state using the 2-databyte ***NMT*** message shown below. There is no reply from the node to this CAN-message.

Host (NMT-Master) ⟶ ***BsCAN*** **(NMT-Slave)**

COB-ID	Byte 0	Byte 1
0x000	1 (*Start_Remote_Node*)	*Node-ID* or 0 (all nodes on the bus)

2.2 Configuration

To trigger the ***BsCAN*** node to probe its *strings* for B-sensor modules and store the newly found configuration to non-volatile memory, a *CANopen* **SDO** (*Service Data Object*) message must be send to read *Object 5000h, subindex 0* (see the Object Dictionary in section 3). All B-sensors found are subsequently reset and calibrated. The reply message sent has one significant databyte, containing the total number of B-sensor modules found.

Here is a description of these CAN-messages, the **SDO**-client (request from host) and **SDO**-server (reply from ***BsCAN***) CAN-message:

Host → BsCAN

COB-ID	**Byte**						
	0	**1**	**2**	**3**	**4**	**5**	**6-7**
0x600+ *NodeID*	0x40	0x00	0x50	0x00	–	–	–

Assuming everything went well and the number of B-sensor modules found is 30 (2Dh), ***BsCAN*** replies with the following message:

BsCAN → Host

COB-ID	**Byte**						
	0	**1**	**2**	**3**	**4**	**5**	**6-7**
0x580+ *NodeID*	0x4F	0x00	0x50	0x00	0x2D	–	–

A full probing operation (scanning 8 strings, each for addresses 0 to 127) takes approximately 10 seconds to complete, also depending on the number of modules found.

If for any reason an **SDO**-client request from the host fails ***BsCAN*** replies with a so-called **SDO** *Abort Transfer* message; this message has the following syntax:

BsCAN → Host

COB-ID	**Byte**				
	0	**1**	**2**	**3**	**4-7**
0x580+ *NodeID*	0x80	*index* *(LSB)*	*index* *(MSB)*	*subindex*	*Abort Code*

See Table 1 below for a description of some *Abort Codes* (see also [2]).

Abort Code	Description
0503 0000	Toggle bit not alternated
0504 0000	SDO protocol timed out
0601 0000	Unsupported access to an object
0602 0000	Object does not exist in the Object Dictionary
0606 0000	Object access failed due to a hardware error
0609 0011	Sub-index does not exist
0800 0000	General error

Table 1. SDO *Abort Domain Transfer*: descriptions of some *Abort Codes* (in byte 4-7).

2.3 Read-out

To trigger the read-out of all connected B-modules (their 3 Hall-sensors and 1 T-sensor) a socalled **SYNC** message must be send.

The **SYNC** message is a CAN-message with a fixed COB-ID and no data bytes:

Host → all (SYNC-)slave nodes

COB-ID
0x080

After receiving this message the ***BsCAN*** node starts up a sequence of AD-conversions on all B-sensor modules simultaneously to convert the H1, H2 and H3 Hall-sensors and the T-sensor, and subsequently reads out the converted analog inputs and sends them one-by-one in a message on the CAN-bus.

Strings are scanned for data from ***BsCAN*** string connector 0 to connector 7, and B-sensor modules on one string are scanned from the lowest to the highest B-sensor module address. Per B-sensor module the messages arrive in the order H1, H2, H3 and T respectively.

In practice this means that after a conversion sequence is started (a **SYNC** message is received by the ***BsCAN*** node) it takes about 4*80=320 ms for the conversion sequence to complete and the first message arrives (for an AD-conversion wordrate of 15 Hz). The 4 messages from one B-sensor module are about 1.5 ms apart. The time between the last message of one B-sensor module and the first message of the next B-sensor module is about 5 ms.

For example: read-out of 32 B-sensor modules on 4 strings by one ***BsCAN*** box takes about 4*80 + 4*5 + 32*1.5 ms, which is roughly about 0.4 seconds from start to finish…

The message containing an ADC value is called a **TPDO** (***T****ransmit* ***P****rocess* ***D****ata* ***O****bject*) in *CANopen* jargon, which is a message without any further *CANopen* protocol overhead. The data bytes in the message contain application data only.

BsCAN will produce –per channel– the following 6-databyte **TPDO** message containing the data for one ADC channel conversion:

***BsCAN* → Host**

TPDO COB-ID	Byte 0	Byte 1	Byte 2	Byte 3-5
0x280+*NodeID*	Address	Chan no	ADC-config	ADC value

with:

Address: B-sensor module address (between 0 and 127).
Chan no: 0 = Hall H1, 1 = Hall H2, 2 = Hall H3, 3 = T-sensor.
ADC value: 24-bits value, LSB in byte 3, MSB in byte 5.
ADC-config: **bit 7**: not used.
bits 6-0: ADC configuration: conversion word rate (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 3000h-307Fh, sub 2,3,4,5,6 and 7.

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

(*NB*: ***BsCAN*** supports other modes of readout for this **PDO** (e.g. timer-triggered), but this is not further described in detail here).

Example messages with ADC data:

***BsCAN* → Host**

TPDO COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x290	0x13	0x02	0x00	0xB0	0xD6	0xFF

This is a message from ***BsCAN*** node 16 (*NodeID*=0x290-0x280=0x10, see COB-ID):

- B-sensor 19 (=0x13, Byte 0)
- Hall-sensor H2 (channel number 2, Byte 1)
- Gain range = 100 mV bipolar, conversion word rate = 15.0 Hz (Byte 2)
- 24-bit ADC-value -10576 (=0xFFD6B0, Byte 3+4+5)

Note: a Hall-sensor conversion value is a 24-bit *signed* number ! (note the negative value above)

***BsCAN* → Host**

TPDO COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x290	0x13	0x03	0x0B	0x26	0x6B	0x98

This is a message from ***BsCAN*** node 16 (*NodeID*=0x290-0x280=0x10, see COB-ID):

- B-sensor 19 (=0x13, Byte 0)
- T-sensor (channel number 3, Byte 1)
- Gain range = 2.5 V unipolar, conversion word rate = 15.0 Hz (Byte 2)
- 24-bit ADC-value 624306 (=0x986B26, Byte 3+4+5)

Note: a T-sensor conversion value is a 24-bit *unsigned* number !

3 BsCAN Object Dictionary

The values of objects marked with ∗ in the *Index* column can be saved in EEPROM for permanent storage in non-volatile memory. They are retrieved from EEPROM at reset and power-up (The B-sensor mapping, Object 4300, is automatically stored).

Communication Profile Area (*BsCAN*)

Index (hex)	Sub Index	Name	Data/ Object	Attr	Default	Comment
1000	-	Device type	U32	RO	00040191h	Meaning: DSP-401 device profile, analogue inputs on device
1001	-	Error register	U8	RO	0	
1002	-	Manufacturer status reg	U32	RO	0	[1] (see footnote)
1008	-	Manufacturer device name	VisStr	RO	"CRYS"	= CRYSTAL-CAN
1009	-	Manufacturer hw version	VisStr	RO	"CRY2"	= CRYSTAL-CAN-V2
100A	-	Manufacturer software version	VisStr	RO	"BC10"	***BsCAN*** application v1.0
100C	-	Guard time [ms]	U16	RO	1000	= 1 second
100D ∗	-	Life time factor	U8	RW	0	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, …
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 ms]	U16	RO	0	Truncated to multiples of 1000; = 30 s, if Heartbeat enabled
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: status for B-sensor modules #0 to #31;
1 bit per module: 0=OKAY, 1=Error/Absent.

Communication Profile Area (*BsCAN*) *(continued…)*

Index (hex)	Sub Index	Name	Data/ Object	Attr	Default	Comment
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

Index (hex)	Sub Index	Name	Data/ Object	Attr	Default	Comment
1A01		2nd Transmit PDO mapping	Record			Data type = PDOMapping
	0	Number of entries	U8	RO	3	*should be 255 for MuxPDO, but this is not a CANopen MPDO…*
	1	B-sensor/ADC address	U32	RO	40000008	*actually not allowed, but…*
	2	B-sensor channel number	U32	RO	30000108	OD-index 3000, sub-index 1: size = 8 bits
	3	24-bit analogue input	U32	RO	*40xx0y18*	OD-index 40xx, sub-index *y*: Analogue inputs, multiplexed, size = 24 bits

Manufacturer-specific Profile Area (*BsCAN*)

Index (hex)	Sub Index	Name	Data/ Object	Attr	Default	Comment
3000		B-sensor #0 ADC-config	Record			CRYSTAL CS5524 24-bit ADC
	0	Number of entries	U8	RO	21	
	1	Number of input channels	U8	RO	7	
*	2	Conversion Word Rate Hall-sensors	U8	RW	0	3-bit code [1] (Global setting!)
*	3	Input Voltage Range Hall	U8	RW	0	3-bit code [2] (Global setting!)
*	4	Unipolar/Bipolar Measurement Mode Hall	U8	RW	0	0 = bipolar, 1 = unipolar (Global setting!)
*	5	Conversion Word Rate Temperature sensor	U8	RW	0	3-bit code [1] (Global setting!)
*	6	Input Voltage Range Temp	U8	RW	0	3-bit code [2] (Global setting!)
*	7	Unipolar/Bipolar Measurement Mode Temp	U8	RW	0	0 = bipolar, 1 = unipolar (Global setting!)
	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
3001		B-sensor #1 ADC-config	Record			**NB**: some settings are global for all B-sensor modules !
	etc	…	…	…	…	…
3002		B-sensor #2 ADC-config	Record			**NB**: some settings are global for all B-sensor modules !
	etc	…	…	…	…	…
…	…	…	…	…		…
307F		B-sensor #127 ADC-config	Record			**NB**: some settings are global for all B-sensor modules !
	etc	…	…	…	…	…

[1] **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

[2] **000**: 100 mV, **001**: 55 mV, **010**: 25 mV, **011**: 1 V, **100**: 5 V, **101**: 2.5 V

Manufacturer-specific Profile Area (*BsCAN*)

Index (hex)	Sub Index	Name	Data/ Object	Attr	Default	Comment
3100		B-sensor module status	Record			Error status, one bit per B-sensor 0=OKAY, 1=Error or Absent
	0	Number of entries	U8	RO	4	
	1	status B-sensors #0-#31	U32	RO	0xfffffff	
	2	status B-sensors #31-#63	U32	RO	0xfffffff	
	3	status B-sensors #63-#95	U32	RO	0xfffffff	
	4	status B-sensors #96-#127	U32	RO	0xfffffff	
3200		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
3201		ADC-reset-and-calibrate B-sensor #1	U8	WO		reset+calib of B-sensor #1 ADC
3202		ADC-reset-and-calibrate B-sensor #2	U8	WO		reset+calib of B-sensor #2 ADC
…	…	…	…	…		…
…	…	…	…	…		…
327F		ADC-reset-and-calibrate B-sensor #127	U8	WO		reset+calib of B-sensor #127 ADC
3280		ADC-reset-and-calibrate **all** B-sensors in config	U8	WO		
4000		Read analogue input B-sensor #0	Record			24 bits analogue value (B-sensor address #0)
	0	Number of entries	U8	RO	7	
	1	Input 1: Hall-sensor H1	I24	RO		1st analog input:24-bit
	2	Input 2: Hall-sensor H2	I24	RO		2nd " " "
	3	Input 3: Hall-sensor H3	I24	RO		3rd " " "
	4	Input 4: current sense	I24	RO		4th " " "
	5	Input 5: NTC T-sensor	I24	RO		5th " " "
	6	Input 6: 0ºC calib input	I24	RO		6th " " "
	7	Input 7: 100ºC calib input	I24	RO		7th " " "
4001		Read analogue input B-sensor #1	Record			24 bits analogue value (B-sensor #1)
	0	Number of entries	U8	RO	7	
	1	Input 1	I24	RO		1st analog input:24-bit
	…	…	…	…		…
	7	Input 7	I24	RO		7th " " "
…	…	…	…	…		…
…	…	…	…	…		…
407F		Read analogue input B-sensor #127	Record			24 bits analogue value (B-sensor #127)
	0	Number of entries	U8	RO	7	
	1	Input 1	I24	RO		1st analog input:24-bit
	…	…	…	…		…
	7	Input 7	I24	RO		7th " " "

Manufacturer-specific Profile Area (*BsCAN*)

Index (hex)	Sub Index	Name	Data/ Object	Attr	Default	Comment
4100		B-sensor address list				
	0	Total number of B-sensors	U8	RO		= number of modules found when *probing* (see Object 5000)
	1	Address of 1st B-sensor	U8	RO		
	…	…	…	…		…
	n	Address of n th B-sensor	U8	RO		
4200		Number of B-sensors per string	Array			
	0	Total number of strings	U8	RO	8	
	1	Number of B-sensors on string #0	U8	RO		
	2	Number of B-sensors on string #1	U8	RO		
	3	Number of B-sensors on string #2	U8	RO		
	4	Number of B-sensors on string #3	U8	RO		
	5	Number of B-sensors on string #4	U8	RO		
	6	Number of B-sensors on string #5	U8	RO		
	7	Number of B-sensors on string #6	U8	RO		
	8	Number of B-sensors on string #7	U8	RO		
4300		B-sensor-to-string mapping	Array			
*	0	string# with B-sensor #0	U8	RO		value 0xFF means: this B-sensor not found in any of the strings
*	1	string# with B-sensor #1	U8	RO		"
*	2	string# with B-sensor #2	U8	RO		"
*	…	…	…	…		…
*	127	string# with B-sensor #127	U8	RO		"
5000	-	Probe strings for B-sensors		RO		stores newly found mapping (Object 4300) in EEPROM; resets and calibrates all B-sensors found; returns Object 4100, sub 0; takes about 10 s to complete.

4 Emergency Objects

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

***BsCAN* → 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 ***BsCAN*** application:

Error Description	Emergency Error Code (byte 0-1)	Error Register bit (Object 1001H) (byte 2)	Manufacturer-specific Error Field (byte 3-7)
CAN communication	8100	10	Byte 3: 81C91 Interrupt Register content Byte 4: 81C91 Mode/Status Register content Byte 5: error counter
Life Guarding	8130	10	(CAN-controller has been reinitialized)
EEPROM: write failed	5000	80	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	80	Byte 3: 42 Byte 4: Parameter block index [2] Byte 5: Error id (1=CRC, 2=length, 4=infoblock)
B-sensor ADC: conversion timeout	5000	80	Byte 3: 51 Byte 4: B-sensor address (0..127) Byte 5: ADC channel number (0..3)
B-sensor ADC: reset failed	5000	80	Byte 3: 52 Byte 4: B-sensor address (0..127) Byte 5: Error id [3]
B-sensor ADC: offset calibration failed	5000	80	Byte 3: 53 Byte 4: B-sensor address (0..127)
B-sensor ADC: gain calibration failed	5000	80	Byte 3: 54 Byte 4: B-sensor address (0..127)
B-sensor ADC problem(s) during initialisation	5000	80	Byte 3: 55

Note that the Error Register (Object Dictionary index 0x1001) can have one or more of the bits shown above set, depending on the node's history of errors since the last reset. The table shows the Error Register bit that gets additionally set when the corresponding error occurs.

[1] **0**: PDO communication parameters, **1**: Guarding parameters, **2**: ADC configuration, **3**: Bsensor-to-string mapping

[2] *idem*

[3] **01**: Reset-Valid bit not set, **02**: Reset-Valid bit not reset, **04**: error in Offset Register value,
08: error in Gain Register value

References

[1] H.Boterenbrood,
B-sensor with addressable Serial Peripheral Interface,
Version 1.3, NIKHEF, Amsterdam, 27 March 2001.
(http://www.nikhef.nl/pub/departments/ct/po/html/Bsensor/Bsensor.pdf

[2] H.Boterenbrood,
CANopen, high-level protocol for CAN-bus,
Version 3.0, NIKHEF, Amsterdam, 20 March 2000.
(http://www.nikhef.nl/pub/departments/ct/po/doc/CANopen30.pdf).

[3] **CRYSTAL CS5521/22/23/24/28, 16-Bit or 24-bit, 2/4/8-Channel ADCs with PGIA,** CIRRUS LOGIC product datasheet, DS317F2, May 2000,
(http://www.cirrus.com).