FIELDBUS APPENDIX
ANYBUS-S CANOPEN

DOC. NO ABS-COP-1.92

HMS INDUSTRIAL NETWORKS AB
PIELEFELTSGATAN 93 - 95
S - 302 50 HALMSTAD
SWEDEN

PHONE: +46 35 17 29 00
FAX: +46 35 17 29 09
e-mail: info@hms-networks.com
web: www.hms-networks.com

Revision notes

Index	Date	Chapter	Author	Revision	Notes
1	1999-01-04	-	MiM	1.0	Created
2	1999-02-07	-	KaJ	1.1	Layout updated
3	1999-05-04	-	MiM	1.2	Minor modifications
4	1999-05-31	-	MiM	1.3	Modified for SCI
5	1999-12-01	-	MiM	1.4	Pre-release of ver 1.5
6	2000-01-02	-	MiM	1.5	Modified for conformance test
7	2000-	All	TTh	1.90	Document transferred to FrameMaker
8	2002-07-17	3.1	TTh	1.91	Updated the bus connector pin descriptions
9	2003-03-11	2.2	TTh	1.92	Added missing switch information

Preface

The data and illustrations found in this manual are not binding. We reserve the right to modify our products in line with our policy of continuous product development. The information in this manual is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks AB.

HMS Industrial Networks AB assumes no responsibility for any errors that may appear in this document.

The product and technology described in this document is patented or patent pending in the following countries: USA, Canada, Japan, Belgium, Denmark, Finland, France, Greece, Ireland, Italy, Luxemburg, Monaco, Netherlands, Portugal, Switzerland, Lichtenstein, Spain, United Kingdom, Sweden, Germany, Austria and others.

Table of Contents

1 Fieldbus introduction

This section provides information about the CANopen organization and network.

1.1 Introduction to CANopen

CANopen has an international user organisation called CAN in Automation (CiA). For further information regarding CANopen matters not associated with AnyBus-S, please contact CiA on e-mail: headquarters@can-cia.de.

General CANopen information is available on Internet: www.can-cia.de

CANopen is used for industrial automation, normally for the control of valves, sensors and I/O units.

1.2 Network overview

The media for the fieldbus is a shielded copper cable consisting of one twisted pair and two optional cables for the external power supply. As standard, the CANopen AnyBus-S module does not use external power supply. The power for the bus is galvanically isolated with a DC/DC converter mounted on the card. The baudrate can be changed from 10 kbits/s up to 1 Mbit/s. This can be done in two different ways, either by the DIP switches mounted on the card or with the 'FIELDBUS_SPECIFIC_INIT' command sent at start-up. Both master-slave and slave-slave communication is possible.

The maximum amount of data that can be handled by the AnyBus-S CANopen module is 2 kbyte of input data and 2 kbyte of output data.

1.3 Technical features for CANopen

Table 1: Technical features for CANopen

Technical features for CANopen
• CANopen specific cable (twisted pair) • Access to intelligence present in low-level devices • Master/Slave and Peer-to-Peer capabilities • Selectable baudrates from 10kbit / s - 1Mbit/s • Max distance 5000m • Support for up to 127 nodes • Node removal without severing the network • Use of sealed or open-style connectors • Provisions for the typical request/response oriented network communications • Provisions for the efficient movement of I/O data • Fragmentation for moving larger bodies of information

1.4 Related documents

Table 2: Documents related to this manual

Name	Description	Document number
AnyBus-S Parallel Design Guide	Main documentation for the parallel interface version of the AnyBus-S	ABS-DGP-1.13
AnyBus-S Serial Design Guide	Main documentation for the serial interface version of the AnyBus-S	ABS-DGS-1.02

2 Module overview

This section provides an overview of the AnyBus-S CANopen module.

2.1 Functional overview

The AnyBus-S module for CANopen is a slave node that can be read and written to from a CANopen master/slave. Via the host connector, the AnyBus-S is connected to an application that gets an instant connection to a CANopen network.

2.2 Mechanical overview

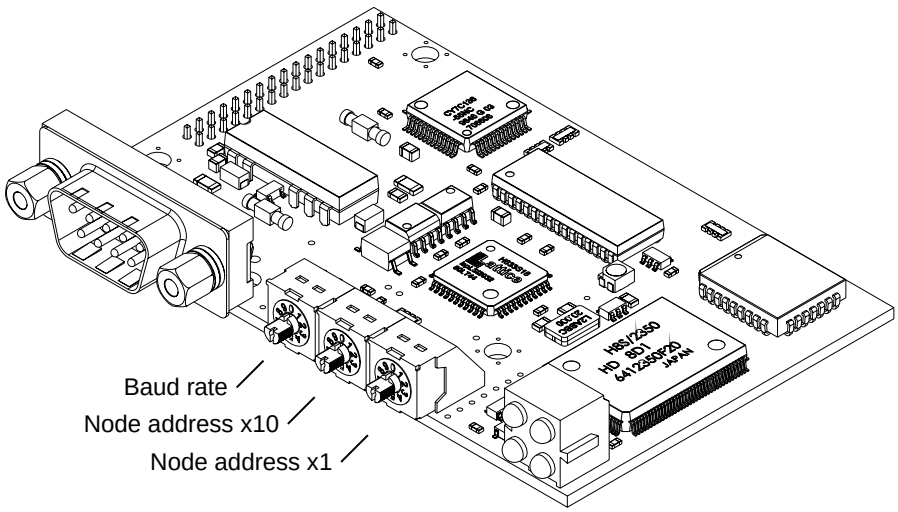


Figure 1: Mechanical overview

2.3 Application interface

This section describes the two access methods available from the application side between the AnyBus-S CANopen module and the application.

- Parallel: Through a Dual Port RAM memory (DPRAM).
- Serial: Through an asynchronous interface

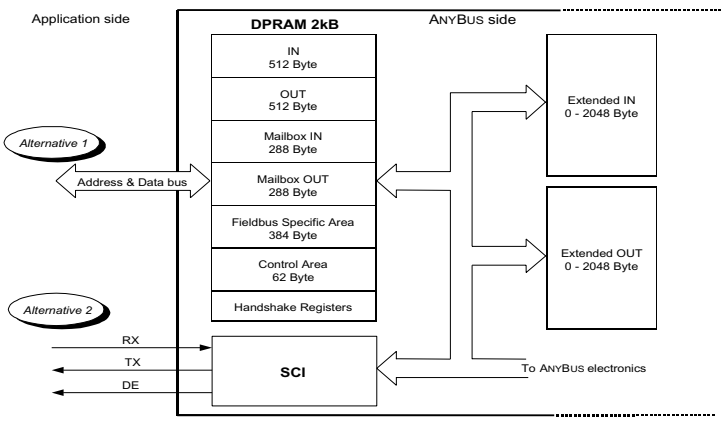


Figure 2: AnyBus-S access methods

2.3.1 Dual-Port RAM

The easiest way to use the AnyBus-S module is to integrate it into the system microprocessor bus. This is achieved using the parallel DPRAM interface. All relevant access and handshaking procedures are achieved through this interface. The handshake procedure is used to ensure that there will always be consistent data available on the host application side as well as on the fieldbus side. In the handshaking procedure, it will generally be the host side that initiates the start of the handshaking and the AnyBus-S module will respond to it.

For more information about designing with the Parallel interface, please consult the AnyBus-S Design Guide for parallel interface.

2.3.2 Asynchronous interface

The asynchronous serial interface allows the AnyBus-S module to exchange data asynchronously with external microprocessors. Some applications cannot access the Dual Port RAM via address and data bus, but have a free serial link. It is therefore possible to operate through the serial interface. The initialisation procedure will be held in a similar way to the Dual Port RAM solution, except the addresses are different and the telegrams are now sent on a serial link.

For more information about designing with the serial interface, please consult the AnyBus-S Design Guide for serial interface.

2.3.3 AnyBus Memory Map

When using Alternative 1 access method (DPRAM), the memory area is a linear data area of 2kByte. When using version 2 (Serial Interface), the information is located in the same structure but with other addresses and it must be accessed via serial telegrams. The AnyBus-S CANopen Memory Map consists of Process Data Objects (fast data) and Service Data Objects (slow data). The data on the CANopen bus is updated when a change is made or on request from another node.

Note: since different hardware is required for the different interfaces, only one of the above mentioned access methods is possible to use in one module.

3 Installation and configuration

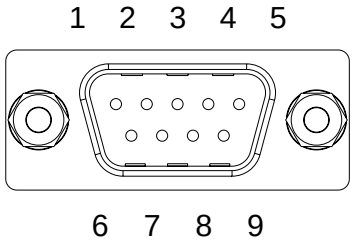
3.1 Fieldbus connector

Note: The standard CANOpen hardware cannot use an external power supply for the bus interface electronics, it will always take the bus power from pin 1 and 2 in the application connector. Contact HMS If you need a module that use an external bus power supply instead of the internal one.

3.1.1 D-SUB connector

Table 3: 9-pin D-SUB fieldbus connector

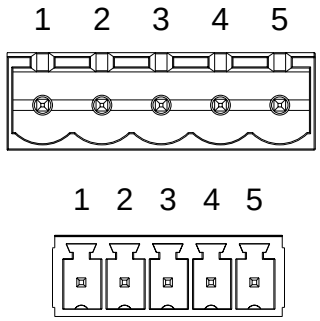
Connector pin	Signal	Description
1	-	Reserved
2	CAN_L	CAN_L Bus line (dominant low)
3	CAN_V-	External bus power minus (optional)
4	-	Reserved
5	CAN_SHLD	CAN cable shield
6	CAN_V-	External bus power minus (optional)
7	CAN_H	CAN_H Bus line (dominant high)
8	-	Reserved
9	CAN_V+	External bus power plus (optional)
Case	CAN_SHLD	CAN cable shield



3.1.2 Screw terminals

Table 4: 5-pin plugable/non-plugable screw terminal fieldbus connector

Connector pin	Signal	Description
1	CAN_V-	External bus power minus (optional)
2	CAN_L	CAN_L bus line (dominant low)
3	CAN_SHLD	CAN cable shield
4	CAN_H	CAN_H bus line (dominant high)
5	CAN_V+	External bus power plus (optional)



3.1.3 2 mm connector

Table 5: 10-pin 2 mm fieldbus connector

Connector pin	Signal	Description
1	CAN _SHLD	Optional CAN shield
2	-	Reserved
3	CAN_L	CAN_L bus line (dominant low)
4	CAN_V-	External bus power minus (optional)
5	CAN_H	CAN_H bus line (dominant high)
6	CAN_V-	External bus power minus (optional)
7	-	Reserved
8	CAN_V+	External bus power plus (optional)
9	-	Reserved
10	CAN_SHLD	CAN cable shield

Pin 1 is the pin closest to the corner of the card. It has a square-shaped soldering pad on the circuit board, where pins 2-10 have circular soldering pads. See Figure 10 for more information.

3.2 Configuration

3.2.1 Node address

The network node address is set with two rotary switches on the option board (ADDRESS_HIGH and ADDRESS_LOW), see Figure 1, or with the 'FIELDBUS_SPECIFIC_INIT' telegram during startup of the module. Possible node address if you use the dip switches are between 1 - 99 in decimal format.

The node address is calculated in the following way:
Node address = (ADDRESS_HIGH * 10) + (ADDRESS_LOW * 1)

Note: Pdo 3 - 8 only have default cobid's if the node address is less than 64.
Note: The node address cannot be changed during operation.

3.2.2 Baudrate

The baudrate is configured with one decimal rotary switch, or with the 'FIELDBUS_SPECIFIC_INIT' telegram during start-up. See table below for supported baudrates. The baudrate switch is located on the ABS module according to Figure 1.

Table 6: Baudrate switch settings

Switch setting	Baudrate
0	Not available
1	10 kbit/s
2	20 kbit/s
3	50 kbit/s
4	125 kbit/s
5	250 kbit/s
6	500 kbit/s
7	800 kbit/s
8	1 Mbit/s
9	Not available

Note: The baudrate can not be changed during operation

3.2.3 Termination

CANopen uses standard CAN termination on the first and last node on the network. The termination resistor should be 120 ohm. This should be connected between CAN_H and CAN_L on the bus. Note that the termination is only used when the AnyBus-S module is the first or last node on the bus.

3.3 LED indicators

The module is equipped with four bi-color status and indication LED's mounted at the front of the module. During start-up a LED test will be performed to make sure the LED's are working. Test sequence is: Red - Green - Off.

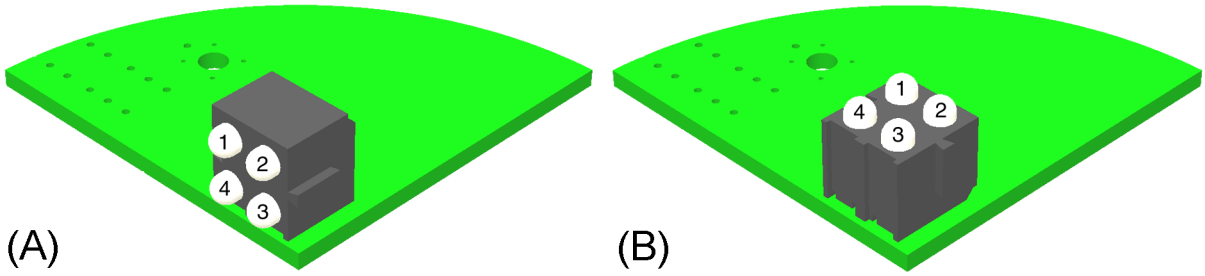


Figure 3: AnyBus-S LED's, with 90° angle mounting (A), and 180° straight mounting (B).

There is also one additional bi-color Watchdog LED on the AnyBus-S module. The function of the watchdog LED is the same for all modules, and information about it can be found in the AnyBus-S Design Guide.

LED 1 is not used on the CANopen module.

Table 7: LED 2 - State indication

Color	Frequency	Description
Green	1 Hz	Module in 'Pre-Operational' state
Green	2 Hz	Module in 'Prepared' state
Green	Steady on	Module in 'Operational' state
Red	1 Hz	Bus initialisation failed

Table 8: LED 3 - Bus indication

Color	Frequency	Description
Green	1 Hz	Bus off / error passive
Green	Steady on	Bus running
Red	1 Hz	Other error
Off	-	Power off or module not initialised

Table 9: LED 4 - Power

Color	Frequency	Description
Green	Steady on	Module has power

3.4 EDS file

This file is used for configuring up the network. If standard configuration is used this file is not required. If a standard CANopen configurator is used an EDS file is required. The latest version of the EDS file can be downloaded from the HMS Industrial Networks web page, www.anybus.com.

4 Functions and operations

4.1 CANopen Functional Description

The memory structure of the CANopen module is shown in chapter 4.2. There are three areas supported by the Any-Bus-S CANopen:

1. High speed data area: this area is transferred on change of value
2. Slow speed data area: this area is transferred on request from another bus node.
3. Mailbox area. In this area a standard mailbox interface is used that includes command, fragmentation of data etc.

4.2 Memory structure for the AnyBus-S CANopen module

Table 10: Memory structure for the AnyBus-S CANopen module

Address for Parallel interface	Address for Serial interface	Contents	Access
000h-00Fh	0000h-0007h	Default Input Fast data area (Area 1)	R/W
010h-1FFh	0008h-03FFh	Default Input Slow data area (Area 2)	R/W
200h-20Fh	1000h-1007h	Default Output Fast data area (Area 1)	R/O
210h-3FFh	1008h-13FFh	Default Output Slow data area (Area 2)	R/O
400h-51Fh	2000h-208Fh	Mailbox commands (Area 3)	R/W
520h-63Fh	2090h-211Fh	Mailbox responses (Area 3)	R/O
640h-7BFh	2120h-21DFh	Fieldbus specific area	Depends on register
7C0h-7FDh	21E0h-21FEh	Control area	Depends on register

4.2.1 IN/OUT area - High Speed Data Area

Writing to any byte of the input area of the module with the proper accessing method, will result in an immediate write to the bus. When new data appear on the bus this is immediately copied to the output area.

4.2.2 IN/OUT area - Slow Speed Data Area

Writing to any byte of the input area of the module with the proper accessing method, will result in a buffering of the data until a request is made on the bus to read the data. A write from the bus will result in a transfer to the output area.

4.2.3 Mailbox area

This area uses a standard communication protocol. The description of this protocol can be found in the AnyBus-S design guide. The mailbox commands described in this section is CANopen specific and only includes the commands available under Message type FIELDBUS SPECIFIC MESSAGES. The function of these messages is to enable the user to access CANopen specific functionality. Below the functionality of these commands are described in detail as well as error codes not included in the design guide.

The following fieldbus specific commands are available for the CANopen ABS module.

Table 11: Fieldbus specific commands

Command ID	Description of command	Section
0x01	Fieldbus Specific init	5.1.1
0x10	CANopen Object Read	5.1.2
0x20	CANopen Object Write	5.1.3
0x30	Emergency message indication	5.1.4
0x40	Fieldbus Remap Command	5.1.5

4.2.4 Fieldbus specific area

This area contains data that is necessary to use for fieldbus specific features.

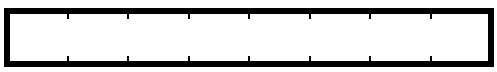
When reading an 8-bit register with the serial interface, the 8-bit register is loaded into the low 8 bits of the 16-bit word, and the high 8 bits is filled with zeroes.

Table 12: Registers in the fieldbus specific area

Address (parallel)	Address (serial)	Size (bytes)	Name	Description
640h	2120h	1	Node address	Contains the nodes current address
641h	2121h	1	Baudrate	Contains the baudrate currently used
643h	2123h	1	Bus state indicator	see object index 2200h in section 6.2.1 - 6.2.3
644h	2124h	1	Module state indicator	see object index 2205h in section 6.2.1 - 6.2.3

Node address	**R/O**

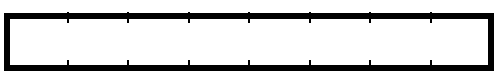
Table 13: Node address register

Address (parallel)	Address (serial)	b7 b6 b5 b4 b3 b2 b1 b0	Description
640h	2120h		Node address

Note: The bits in this register have the same configuration as the rotary configuration switches on the module.

Baudrate	**R/W**

Table 14: Baudrate register

Address (parallel)	Address (serial)	b7 b6 b5 b4 b3 b2 b1 b0	Description
641h	2121h		Baudrate

Note: The bits in this register have the same configuration as the rotary configuration switches on the module.

Bus state indicator	R/O

Table 15: Bus state indicator register

Address (parallel)	Address (serial)	b7 b6 b5 b4 b3 b2 b1 b0	Description
643h	2123h		Bus state indicator

Note: Also see object index 2200h in section 6.2.1 - 6.2.3

Module state indicator	R/O

Table 16: Module state indicator register

Address (parallel)	Address (serial)	b7 b6 b5 b4 b3 b2 b1 b0	Description
644h	2124h		Module state indicator

Note: Also see object index 2205h in section 6.2.1 - 6.2.3

4.2.5 Control area

These registers contain information about the status and control environment, both for the fieldbus and the module itself, e.g. fieldbus type, fieldbus state, firmware version, configuration, etc. The following registers are specific for the CANopen module. For further information, please see AnyBus-S Design Guide.

Table 17: Registers in the control area

Address (parallel)	Address (serial)	Size (bytes)	Name	Description
7CCh	21E6h	2	Fieldbus type	Code describing the fieldbus type
7CEh	21E7h	2	Module software version	Version number of the modules software

Fieldbus Type	R/O

Table 18: Fieldbus Type register

Address (parallel)	Address (serial)	b7 b6 b5 b4 b3 b2 b1 b0	Description
7CCh	21E6h		Fieldbus type - high byte
7CDh	-		Fieldbus type - low byte

For a CANopen module, this value should be 0x0020.

Note: Since the serial interface is word-oriented, both high-byte and low-byte are read when reading address 21E6h.

Module Software version	**R/O**

Table 19: Module Software Verison register

Address (parallel)	Address (serial)	b7 b6 b5 b4 b3 b2 b1 b0	Description
7CEh	21E7h		Module software version - high byte
7CDh	-		Module software version - low byte

Note: Since the serial interface is word-oriented, both bytes are read when reading address 21E6h.

4.3 Initialization

- Install the module in the application.
- Configure Baudrate with baudrate switch
- Configure Node Address with address switches
- Connect the CANopen network cable
- Connect a CANopen master to the system
- Initialise the module according to standard AnyBus-S initialisation.

5 Mailbox interface

5.1 Fieldbus specific mailbox messages

5.1.1 Fieldbus Specific Init

This telegram is used to make a fieldbus specific initialisation of the AnyBus-S module. The command includes fieldbus specific parameters, as well as standard ANYBUS parameters. When you are going to send a command, the following command response is expected. There are two options available for this command: The first option is to write the ANYBUS set-up variables such as IN/OUT lengths in the fieldbus telegram and not send any ANYBUS init command. The second option is to send a separate ANYBUS init command and then send the fieldbus specific variables such as Node address and baudrate in the fieldbus init command.

Table 20: Mailbox message "FIELDBUS_SPECIFIC_INIT" summary

Parameter	Description
Command initiator	Application
Message Name	FIELDBUS_SPECIFIC_INIT
Message type	0x02
Command number	0x0001h
Fragmented	No
Extended header data	
Message data	
Response message	

Option 1 for parallel interface - init without AnyBus init command sent for parallel interface (ANYBUS variables sent in this telegram).

Command and response layout:

Table 21: Mailbox message "FIELDBUS_SPECIFIC_INIT" memory layout for parallel interface

Register Name	*Command*	*Expected response*	
Message ID	Increased every mess.	Increased every mess.	
Message information	0x4002	0x0002	*Fieldbus Specific command*
Command	0x0001	0x0001	*Fieldbus Specific init*
Data size	0x0016	0x0016	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data*
Extended word 2	-	-	"
Extended word 3	-	-	"
Extended word 4	-	-	"
Extended word 5	-	-	"
Extended word 6	-	-	"
Extended word 7	0x0000	Fieldbus Specific Fault handling	
Extended word 8	0x0000	Standard ANYBUS Fault handling	
Message data word 1	IN cyclic I/O length	IN cyclic I/O length	*Message data*
Message data word 2	IN DPRAM length	IN DPRAM length	"
Message data word 3	IN total length	IN total length	"
Message data word 4	OUT cyclic I/O length	OUT cyclic I/O length	"
Message data word 5	OUT DPRAM length	OUT DPRAM length	"
Message data word 6	OUT total length	OUT total length	"
Message data word 7	Module Status	Module Status	"
Message data word 8	Interrupt Notification	Interrupt Notification	"
Message data word 9	Watchdog counter	Watchdog counter	"
Message data word 10	Node address	Node address	"
Message data word 11	Baudrate	Baudrate	"

The first 9 message data words are specified in the AnyBus Design Guide for parallel interface. The last 2 message data words are specified in Table 24.

The Fieldbus Specific Fault Handling is specified in extended word 7, the Standard AnyBus fault handling is specified in extended word 8.

Standard AnyBus Fault handling is specified in the AnyBus-S design guide and the Fieldbus Specific Fault Handling is specified inTable 25.

Option 1 for serial interface - init without AnyBus init command sent for serial interface (ANYBUS variables sent in this telegram).

Command and response layout:

Table 22: Mailbox message "FIELDBUS_SPECIFIC_INIT" memory layout for serial interface

Register Name	*Command*	*Expected response*	
Message ID	Increased every mess.	Increased every mess.	
Message information	0x4002	0x0002	*Fieldbus Specific command*
Command	0x0001	0x0001	*Fieldbus Specific init*
Data size	0x0016	0x0016	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data*
Extended word 2	-	-	"
Extended word 3	-	-	"
Extended word 4	-	-	"
Extended word 5	-	-	"
Extended word 6	-	-	"
Extended word 7	0x0000	Fieldbus Specific Fault handling	
Extended word 8	0x0000	Standard ANYBUS Fault handling	
Message data word 1	IN cyclic I/O length	IN cyclic I/O length	*Message data*
Message data word 2	IN total length	IN total length	"
Message data word 3	OUT cyclic I/O length	OUT cyclic I/O length	"
Message data word 4	OUT total length	OUT total length	"
Message data word 5	Module Status	Module Status	"
Message data word 6	ComMode	ComMode	"
Message data word 7	Slow Data Priority	Slow Data Priority	"
Message data word 8	Heartbeat Min Time	Heartbeat Min Time	"
Message data word 9	Reserved	Reserved	"
Message data word 10	Node address	Node address	"
Message data word 11	Baudrate	Baudrate	"

The first 8 message data words are specified in the AnyBus-S Design guide for Serial interface, the 9th message data word is reserved, and the last 2 message data words are specified in Table 24.

The Fieldbus Specific Fault Handling is specified in extended word 7, The Standard AnyBus fault handling is specified in extended word 8.

The standard AnyBus error codes is specified in the AnyBus-S design guide and the fieldbus specific error codes is specified inTable 25.

Option 2 for parallel/serial interface - init with ANYBUS init command sent before for both Serial and Parallel Interface (ANYBUS variables sent in the command before this one):

Table 23: Mailbox message "FIELDBUS_SPECIFIC_INIT" memory layout for parallel/serial interface

Register Name	*Command*	*Expected response*	
Message ID	Increased every mess.	Increased every mess.	
Message information	0x4002	0x0002	*Fieldbus Specific command*
Command	0x0001	0x0001	*Fieldbus Specific init*
Data size	0x0004	0x0004	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data*
Extended word 2	-	-	"
Extended word 3	-	-	"
Extended word 4	-	-	"
Extended word 5	-	-	"
Extended word 6	-	-	"
Extended word 7	-	-	"
Extended word 8	0x0000	Fieldbus Specific Fault handling	
Message data word 1	Node address	Node address	*Message data*
Message data word 11	Baudrate	Baudrate	"

The two message data words are specified in Table 24 below.

Fieldbus Specific Fault Handling is specified in extended word 8.

The fieldbus specific error codes is specified inTable 25.

Table 24: Possible values for Node address and Baudrate in mailbox message "FIELDBUS_SPECIFIC_INIT"

Variable	Description	Default value	Possible values
Node address	Node address of the CANopen node	Set on DIP switch	1 - 127
Baudrate	Baudrate	Set on DIP switch	1 - 8

Table 25: Possible error codes for mailbox message "FIELDBUS_SPECIFIC_INIT"

Variable	Description	Possible values
Node address	Node address error	Bit 1 set
Baudrate	Baudrate error	Bit 2 set

5.1.2 Object Read

This section describes the 'OBJECT_READ' mailbox telegram. This telegram is used to read a specific entry from the CANopen specific objects. When you are going to send a command, the following command response is expected.

Table 26: Mailbox message "OBJECT_READ" summary

Parameter	Description
Command initiator	Application
Message Name	OBJECT_READ
Message type	0x02
Command number	0x0010
Fragmented	No
Extended header data	
Message data	
Response message	

The possible objects to access are listed in standard CANopen communication profile (DS301) and in object description, section 6.1.3, 6.1.4 and 6.2.

Command and response layout:

Table 27: Mailbox message "OBJECT_READ" memory layout

Register Name	*Command*	*Expected response*	
Message ID	Increased every mess.	Increased every mess.	
Message information	0x4002	0x0002	*Fieldbus Specific command*
Command	0x0010	0x0010	*Object Read*
Data size	0x0000	0x0000	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	Index	Index	
Extended word 2	Subindex	Subindex	
Extended word 3	0x0000	Length	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault handling	
Message data word	-	The data bytes of the object read are returned here	

The error codes possible are specified in Table 28 below. The information about errors are returned in extended word 8.

Table 28: Possible error codes for mailbox message "OBJECT_READ"

Description	Possible values
Object does not exist	Bit 1 set
Subindex does not exist	Bit 2 set
No read permission	Bit 3 set
The number of bytes returned is to high	Bit 4 set
Other fieldbus Specific error	Bit 16 set

5.1.3 Object Write

This section describes the 'OBJECT_WRITE' mailbox telegram. This telegram is used for writing to a specific entry. When you are going to send a command, the following command response is expected.

Table 29: Mailbox message "OBJECT_WRITE" summary

Parameter	Description
Command initiator	Application
Message Name	OBJECT_WRITE
Message type	0x02
Command number	0x0020
Fragmented	No
Extended header data	
Message data	
Response message	

Command and response layout:

Table 30: Mailbox message "OBJECT_WRITE" memory layout

Register Name	*Command*	*Expected response*	
Message ID	Increased every mess.	Increased every mess.	
Message information	0x4002	0x0002	*Fieldbus Specific command*
Command	0x0020	0x0020	*Object Write command*
Data size	Number of data bytes to write	Number of data bytes that has been written	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	Index	Index	
Extended word 2	Subindex	Subindex	
Extended word 3	Number of data bytes to write	Number of data bytes that has been written	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault handling	
Message data word	The data bytes of the message is entered here	The data bytes of the message is returned here	

The possible objects to access are listed in standard CANopen communication profile (DS301) and in object description, section 6.1.3, 6.1.4 and 6.2.

The error codes possible are specified in Table 31 below. The information about errors are returned in extended word 8.

Table 31: Possible error codes for mailbox message "OBJECT_WRITE"

Description	Possible values
Object does not exist	Bit 1 set
Subindex does not exist	Bit 2 set
No Write permission	Bit 3 set
Wrong byte count error	Bit 4 set
Value to High	Bit 5 set
Value to Low	Bit 6 set
Mapping error	Bit 7 set
No Read Permission	Bit 8 set
Other fieldbus Specific error	Bit 16 set

5.1.4 Emergency Message Indication

This section describes the 'EMERGENCY MESSAGE TELEGRAM' mailbox telegram. This telegram is used for sending a CANopen specific emergency telegram onto the bus.

Table 32: Mailbox message "EMERGENCY_MESSAGE" summary

Parameter	Description
Command initiator	Application
Message Name	EMERGENCY_MESSAGE
Message type	0x02
Command number	0x0030
Fragmented	No
Extended header data	
Message data	
Response message	

Command and response layout:

Table 33: Mailbox message "EMERGENCY_MESSAGE" memory layout

Register Name	*Command*	*Expected response*	
Message ID	Increased every mess.	Increased every mess.	
Message information	0x4002	0x0002	*Fieldbus Specific command*
Command	0x0030	0x0030	*Emergency message command*
Data size	0x0007	0x0007	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault handling	
Message data word 1	Emergency Error Code 1	Emergency Error Code 1	*Emergency Error Code*
Message data word 2	Emergency Error Code 2	Emergency Error Code 2	*Emergency Error Code*
Message data word 3	Emergency Error Code 3	Emergency Error Code 3	*Error Register (h'1001)*
Message data word 4	Emergency Error Code 4	Emergency Error Code 4	*Manufacturer Specific Error*
Message data word 5	Emergency Error Code 5	Emergency Error Code 5	*Manufacturer Specific Error*
Message data word 6	Emergency Error Code 6	Emergency Error Code 6	*Manufacturer Specific Error*
Message data word 7	Emergency Error Code 7	Emergency Error Code 7	*Manufacturer Specific Error*

5.1.5 Object Mapping

This section describes the 'OBJECT_REMAPPING' mailbox telegram. This telegram is used to get an easy way of mapping the Process Data Objects onto the DPRAM area or the SCI internal buffer area. Send down if the mapping is a receive or a transmit object and the number of the object. The data should contain the address in the DPRAM or the address in the SCI internal buffer area where you want to map the object.

Table 34: Mailbox message "OBJECT_REMAPPING" summary

Parameter	Description
Command initiator	Application
Message Name	OBJECT_REMAPPING
Message type	0x02
Command number	0x0040
Fragmented	No
Extended header data	
Message data	
Response message	

Command and response layout:

Table 35: Mailbox message "OBJECT_REMAPPING" memory layout

Register Name	*Command*	*Expected response*	
Message ID	Increased every mess.	Increased every mess.	
Message information	0x4002	0x0002	*Fieldbus Specific command*
Command	0x0040	0x0040	*Fieldbus Specific init*
Data size	Number of data bytes to be included in the message	Number of data bytes to be included in the message	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	Transmit = 0x0100 Receive = 0x0200	Transmit = 0x0100 Receive = 0x0200	
Extended word 2	Number of the object	Number of the object	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault handling	
Message data word 1	The DPRAM address or SCI address of the first byte in the object	The DPRAM address or SCI address of the first byte in the object	*The address is defined from zero*
Message data word 2	The DPRAM address or SCI address of the second byte in the object	The DPRAM address or SCI address of the second byte in the object	*The address is defined from zero*
Message data word 3	The DPRAM address or SCI address of the third byte in the object	The DPRAM address or SCI address of the third byte in the object	*The address is defined from zero*
Message data word 4	The DPRAM address or SCI address of the fourth byte in the object	The DPRAM address or SCI address of the fourth byte in the object	*The address is defined from zero*
Message data word 5	The DPRAM address or SCI address of the fifth byte in the object	The DPRAM address or SCI address of the fifth byte in the object	*The address is defined from zero*
Message data word 6	The DPRAM address or SCI address of the sixth byte in the object	The DPRAM address or SCI address of the sixth byte in the object	*The address is defined from zero*
Message data word 7	The DPRAM address or SCI address of the seventh byte in the object	The DPRAM address or SCI address of the seventh byte in the object	*The address is defined from zero*
Message data word 8	The DPRAM address or SCI address of the eighth byte in the object	The DPRAM address or SCI address of the eighth byte in the object	*The address is defined from zero*

The error codes possible are specified in Table 36 below. The information about errors are returned in extended word 8.

Table 36: Possible error codes for mailbox message "OBJECT_REMAPPING"

Description	Possible values
PDO number error	Bit 1 set
Direction error	Bit 2 set
Data error	Bit 3 set
Other fieldbus Specific error	Bit 4 set

6 Fieldbus specific lists and tables

6.1 Object Description for the CANopen Module

6.1.1 Default Fast Input Data to the CANopen Bus

Table 37: Default Fast Input Data to the CANopen Bus

Input data (High Speed)	PDO mapping	Default COB-ID	Object index	Object subindex	Default State
Input data byte 1 - 8	TPDO1	384+ node address	2000h	1 - 8	Enabled
Input data byte 9 - 16	TPDO2	640 + node address	2000h	9 - 16	Enabled
Input data byte 17 - 24	TPDO3	448 + node address	2000h	17 - 24	Disabled
Input data byte 25 - 32	TPDO4	704 + node address	2000h	25 - 32	Disabled
Input data byte 33 - 40	TPDO5	960 + node address	2000h	33 - 40	Disabled
Input data byte 41 - 48	TPDO6	1088 + node address	2000h	41 - 48	Disabled
Input data byte 49 - 56	TPDO7	1216 + node address	2000h	49 - 56	Disabled
Input data byte 57 - 64	TPDO8	1344 + node address	2000h	57 - 64	Disabled

6.1.2 Default Fast Output Data from the CANopen Bus

Table 38: Default Fast Output Data from the CANopen Bus

Input data (High Speed)	PDO mapping	Default COB-ID	Object index	Object subindex	Default State
Output data byte 1 - 8	RPDO1	512 + node address	2100h	1 - 8	Enabled
Output data byte 9 - 16	RPDO2	768 + node address	2100h	9 - 16	Enabled
Output data byte 17 - 24	RPDO3	576 + node address	2100h	17 - 24	Disabled
Output data byte 25 - 32	RPDO4	832 + node address	2100h	25 - 32	Disabled
Output data byte 33 - 40	RPDO5	896 + node address	2100h	33 - 40	Disabled
Output data byte 41 - 48	RPDO6	1024 + node address	2100h	41 - 48	Disabled
Output data byte 49 - 56	RPDO7	1152 + node address	2100h	49 - 56	Disabled
Output data byte 57 - 64	RPDO8	1280 + node address	2100h	57 - 64	Disabled

Note: The default COB-ID for PDO3 - PDO8 is valid only if the node address is lower than 64. If the node address is greater than 63 the COB-ID has to be configured at start-up.

6.1.3 Total Input Data to the CANopen Bus

Table 39: Total Input Data to the CANopen Bus

Input data (High Speed)	Object Index	Object Subin-dex
Input data byte 1 - 128	2000h	1 - 128
Input data byte 129 - 256	2001h	1 - 128
Input data byte 257 - 384	2002h	1 - 128
Input data byte 385 - 512	2003h	1 - 128
Input data byte 513 - 640	2004h	1 - 128
Input data byte 641 - 768	2005h	1 - 128
Input data byte 769 - 896	2006h	1 - 128
Input data byte 897 - 1024	2007h	1 - 128
Input data byte 1025 - 1152	2008h	1 - 128
Input data byte 1153 - 1280	2009h	1 - 128
Input data byte 1281 - 1408	200Ah	1 - 128
Input data byte 1409 - 1536	200Bh	1 - 128
Input data byte 1537 - 1664	200Ch	1 - 128
Input data byte 1665 - 1792	200Dh	1 - 128
Input data byte 1793 - 1920	200Eh	1 - 128
Input data byte 1921 - 2048	200Fh	1 - 128

6.1.4 Total Output Data from the CANopen Bus

Table 40: Total Output Data from the CANopen Bus

Output data (High Speed)	Object Index	Object Subin-dex
Output data byte 1 - 128	2100h	1 - 128
Output data byte 129 - 256	2101h	1 - 128
Output data byte 257 - 384	2102h	1 - 128
Output data byte 385 - 512	2103h	1 - 128
Output data byte 513 - 640	2104h	1 - 128
Output data byte 641 - 768	2105h	1 - 128
Output data byte 769 - 896	2106h	1 - 128
Output data byte 897 - 1024	2107h	1 - 128
Output data byte 1025 - 1152	2108h	1 - 128
Output data byte 1153 - 1280	2109h	1 - 128
Output data byte 1281 - 1408	210Ah	1 - 128
Output data byte 1409 - 1536	210Bh	1 - 128
Output data byte 1537 - 1664	210Ch	1 - 128
Output data byte 1665 - 1792	210Dh	1 - 128
Output data byte 1793 - 1920	210Eh	1 - 128
Output data byte 1921 - 2048	210Fh	1 - 128

6.2 Other Parameters

6.2.1 Parameters Combined Parallel and Serial Interface

Table 41: Parameters Combined Parallel and Serial Interface

Other parameters	PDO map-ping	Object Index, Subindex	Indication	Type
Bus state indicator	OK	2200h , 00	1 = Bus running 2 = Bus error	UNSIGNED8
Module state indicator	OK	2205h , 00	1 = Init error 2 = Prepared 3 = Pre operational 4 = Operational	UNSIGNED8
Module serial number	RTR only	2210h , 00	Unique module serial number	UNSIGNED32
Vendor ID	RTR only	2211h , 00	Manufacturer ID	UNSIGNED16
Module status	RTR only	2212h , 00	Indicates status set by the application during init	UNSIGNED16
IN cyclic I/O length	RTR only	2240h , 00	Fast in length	UNSIGNED16
IN Total length	RTR only	2242h , 00	Total in length	UNSIGNED16
OUT cyclic I/O length	RTR only	2243h , 00	Fast out length	UNSIGNED16
OUT Total length	RTR only	2245h , 00	Total out length	UNSIGNED16
LED status	OK	2260h - 2263h, 00	Status of the LED’s on the card	UNSIGNED8
Bus-off timeout	RTR only	2800h , 00	Indicates the number of ms needed before the node reinitiates and enters pre-operational	UNSIGNED16

6.2.2 Parameters Parallel Interface

Table 42: Parameters Parallel Interface

Other parameters	PDO map-ping	Object Index, Subindex	Indication	Type
Interrupt count	OK	2220h , 00	Interrupt from the application to the AnyBus-S module	UNSIGNED16
Interrupt cause	OK	2221h , 00	Indicates the current interrupt source	UNSIGNED16
Interrupt notification	OK	2222h , 00	Indicates what source can generate an interrupt	UNSIGNED16
Watchdog counter IN	OK	2230h , 00	Watchdog in from the application	UNSIGNED16
Watchdog counter OUT	OK	2231h , 00	Watchdog out to the application	UNSIGNED16
IN DPRAM length	RTR only	2241h , 00	Total in length in DPRAM	UNSIGNED16
OUT DPRAM length	RTR only	2244h , 00	Total out length in DPRAM	UNSIGNED16

6.2.3 Parameters Serial Interface

Table 43: Parameters Serial Interface

Other parameters	PDO mapping	Object Index, Subindex	Indication	Type
Com Mode	RTR only	2220h , 00	Indicates What Communication Mode The Module Is In	UNSIGNED16
Slow Data Priority	RTR only	2221h , 00	Number of times the fast data is transferred before one slow data package is sent	UNSIGNED16
Heartbeat Min Time	RTR only	2222h , 00	The minimum time between two heartbeat messages in ms.	UNSIGNED16
Bootloader Version	RTR only	2230h , 00	Watchdog in from the application	UNSIGNED16

7 Electrical specifications

7.1 Current consumption

Both the module electronics and the fieldbus interface should be supplied with regulated 5V DC. For more information regarding the power supply, consult the AnyBus-S Design Guide.

Table 44: AnyBus-S CANopen current consumption

Symbol	Description	Min.	Typ.	Max.	Unit
I_{INMOD}	Supply current, module electronics	-	TBD	TBD	mA
I_{INBUS}	Supply current, bus interface	-	TBD	TBD	mA

8 Mechanical specifications

8.1 Overview

This section contains mechanical drawings for the AnyBus-S module for CANopen. The fieldbus interface can be either side or top mounted (reverse mounting is not allowed on the 9 pin D-SUB connector). Several other connectors, including plugable screw terminal, are also available for the CANopen AnyBus-S module. For more information regarding other connectors please contact HMS Fieldbus Systems marketing department.

The mechanical drawings describe the standard configuration of the AnyBus-S CANopen module.
Article number for parallel interface: AB4003
Article number for serial interface: AB4025

For further information regarding the AnyBus-S module, we refer to the AnyBus-S Design Guide.

8.2 Mechanical drawings in this manual

These drawings are contained in this section:

- AnyBus-S CANopen module, angled configuration, 3D view
- AnyBus-S CANopen module, angled configuration, top view
- AnyBus-S CANopen module, angled configuration, front and side view
- AnyBus-S CANopen module, straight configuration, 3D view
- AnyBus-S CANopen module, straight configuration, top view
- AnyBus-S CANopen module, straight configuration, front and side view
- AnyBus-S CANopen module, PCB connection points

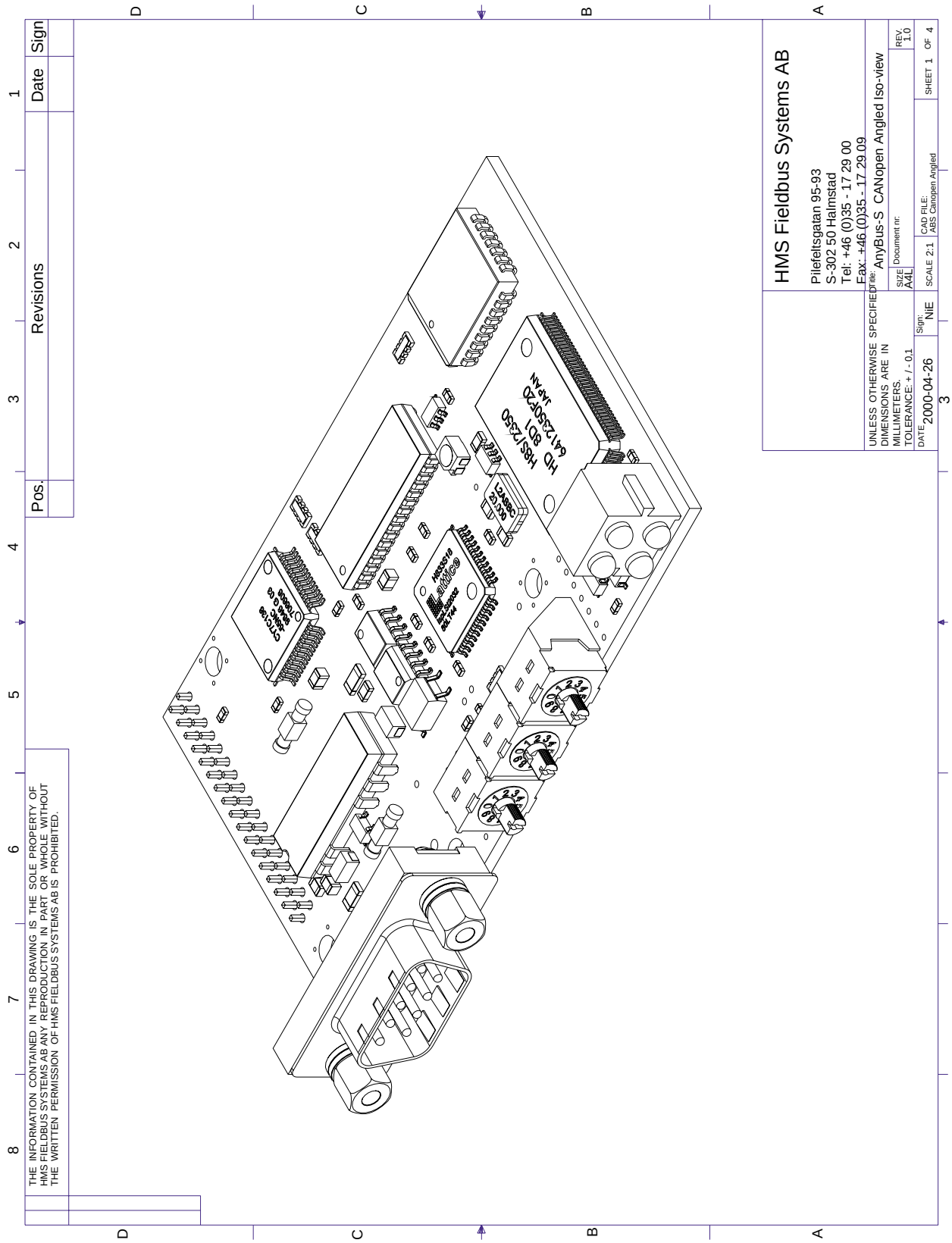


Figure 4: AnyBus-S CANopen module, angled configuration, 3D view

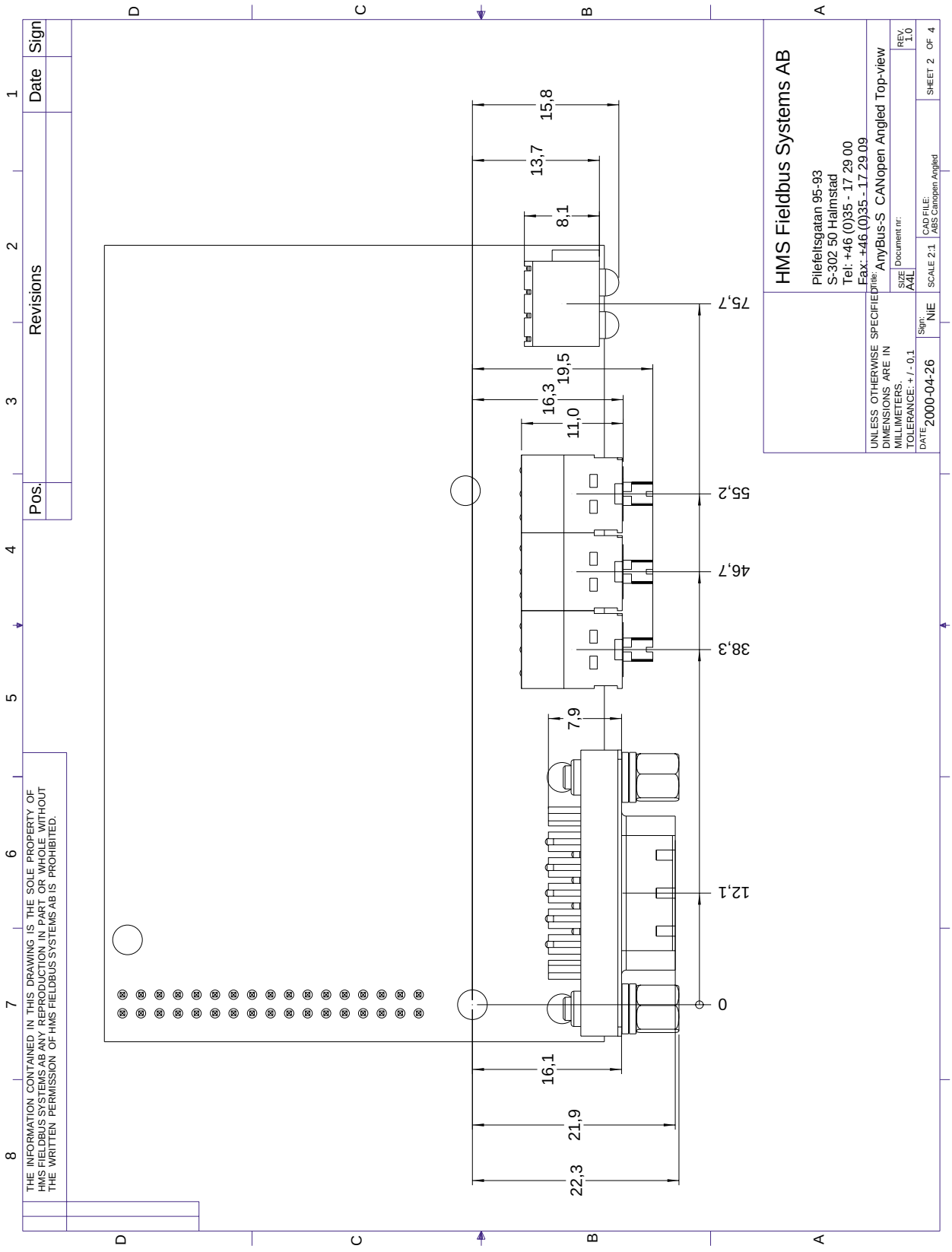


Figure 5: AnyBus-S CANopen module, angled configuration, top view

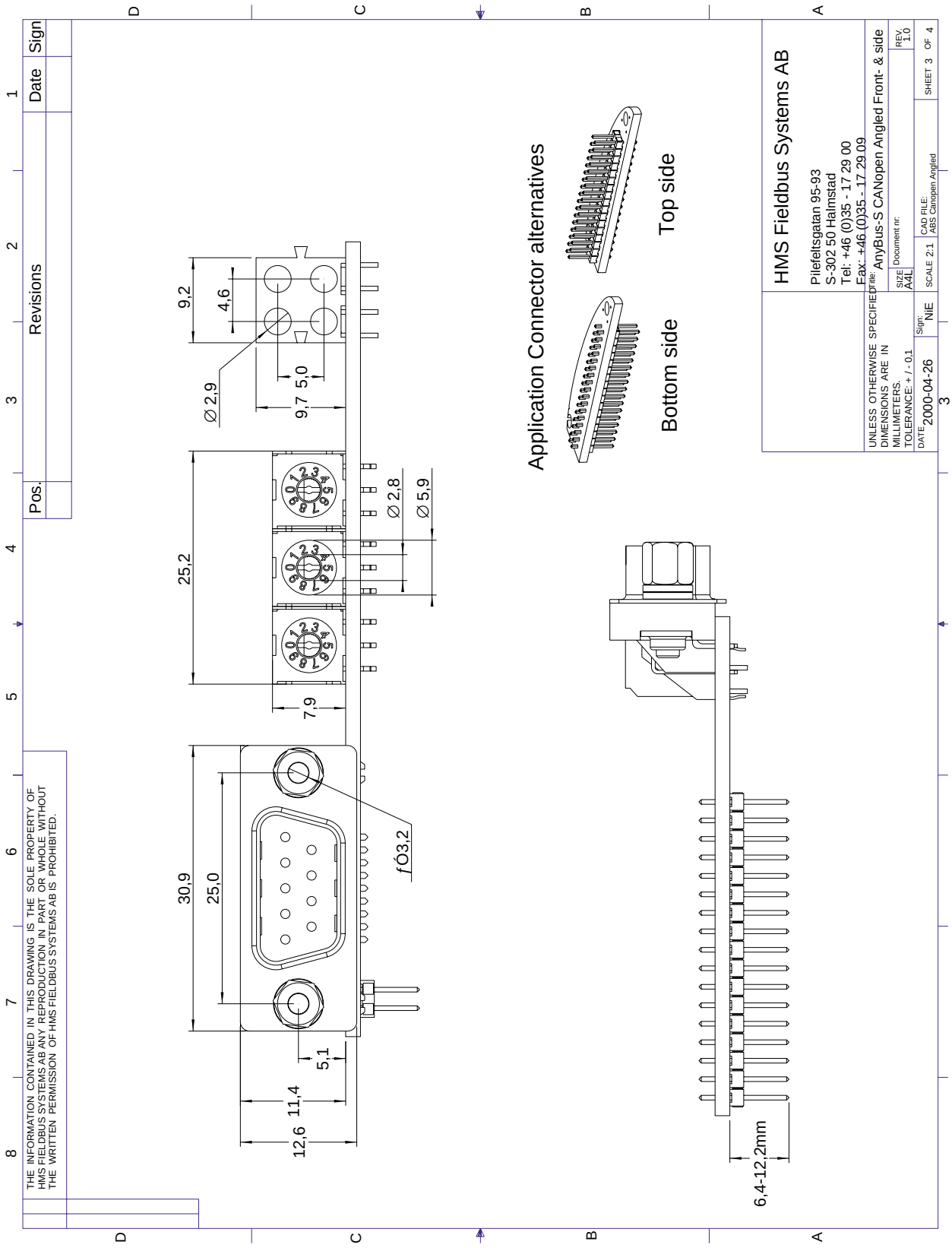


Figure 6: AnyBus-S CANopen module, angled configuration, front and side view

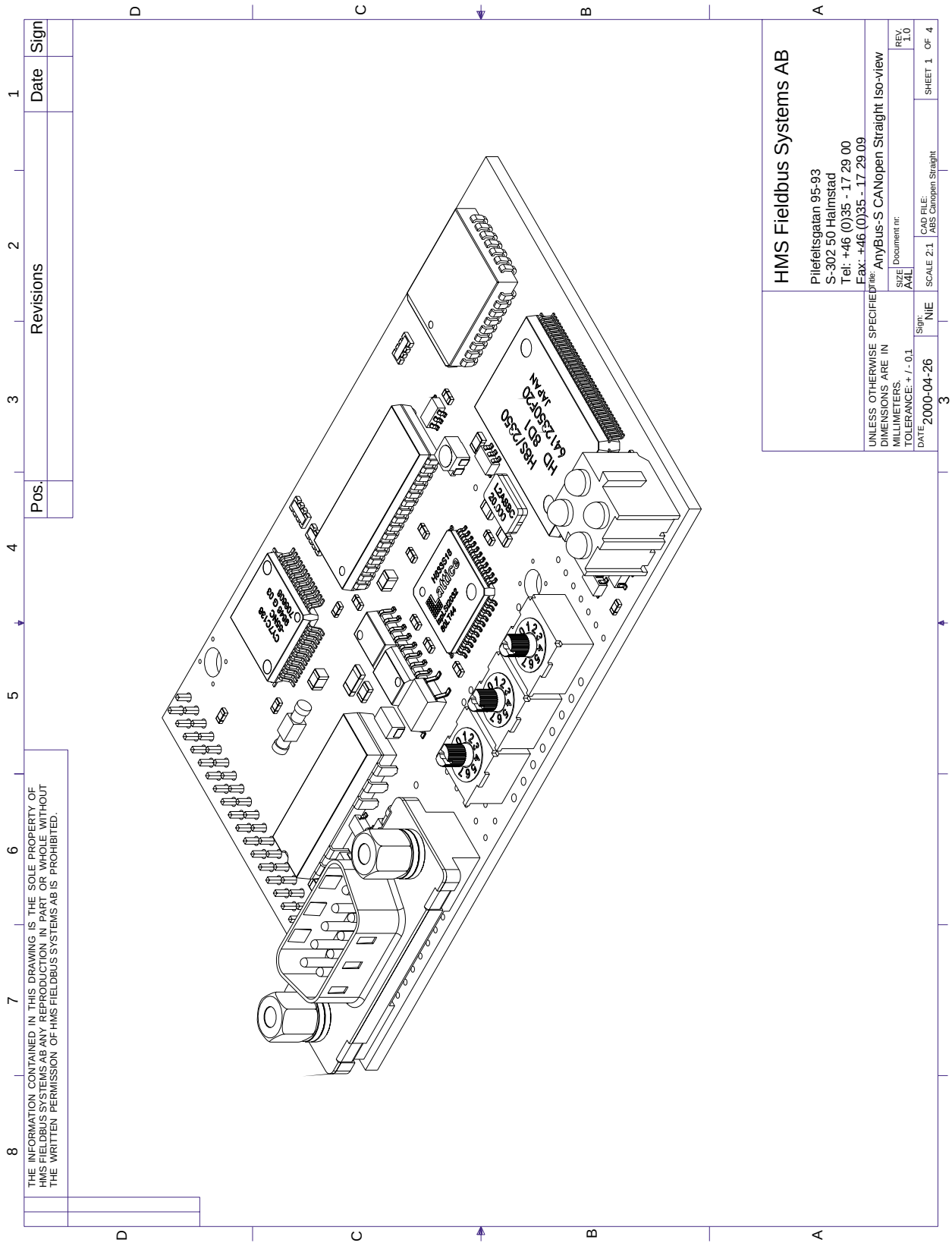


Figure 7: AnyBus-S CANopen module, straight configuration, 3D view

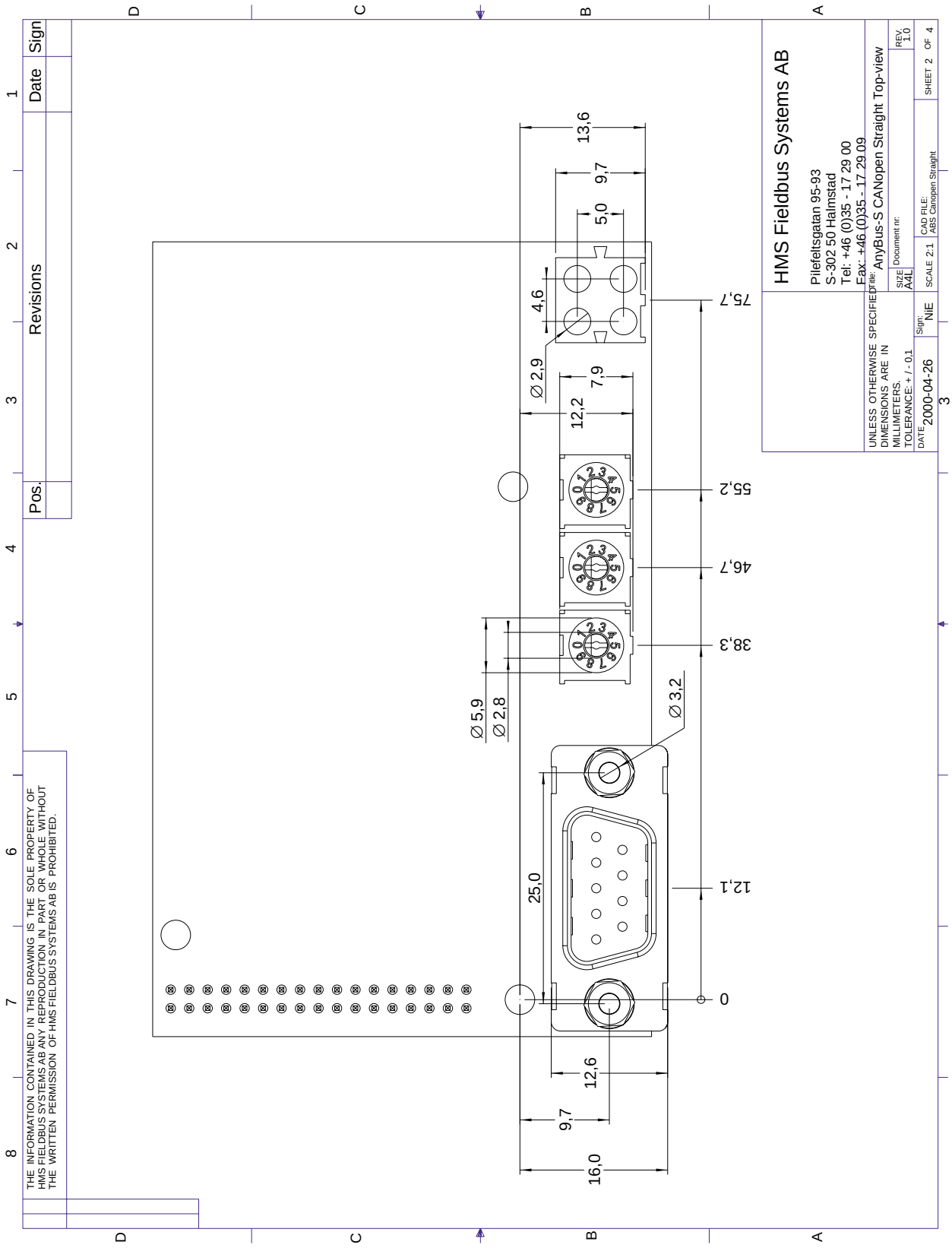


Figure 8: AnyBus-S CANopen module, straight configuration, top view

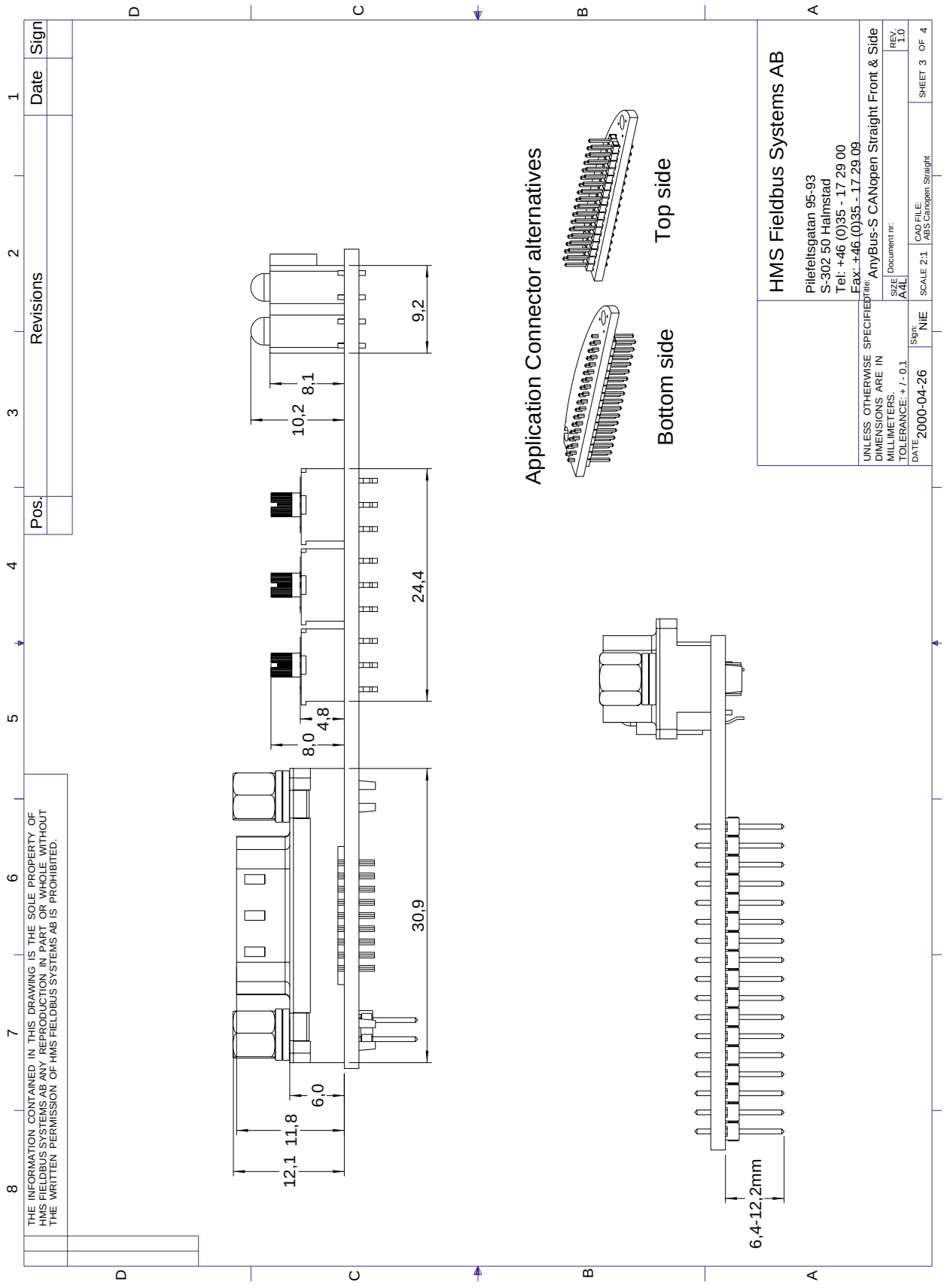


Figure 9: AnyBus-S CANopen module, straight configuration, front and side view

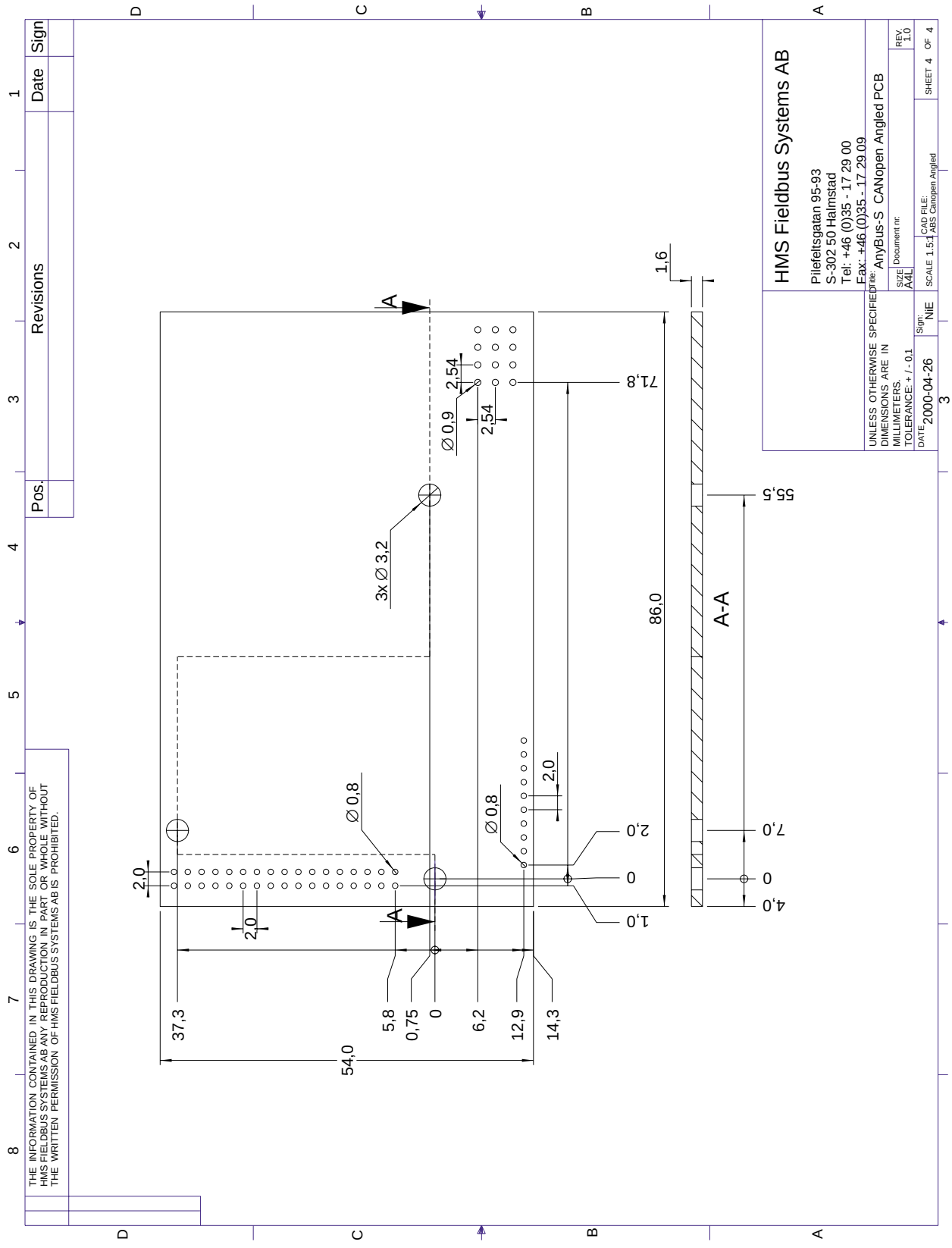


Figure 10: AnyBus-S CANopen module, PCB connection points

9 List of figures

10 List of tables