H1 Very Forward Proton Spectrometer

CAN Heidenhain Interface V2

Preliminary !

J. Schütt, R. van Staa
Universität Hamburg

Institut für Experimentalphysik
May 2003

Content

Introduction

The CAN Heidenhain Interface provides an easy and cheap computer access to absolute Heidenhain encoders by means of the widely used popular CAN field bus. It is restricted to encoders, which are equipped with the Heidenhain EnDat® interface[1)]. EnDat (**En**coder **Dat**a) is a serial bi-directional interface, which outputs absolute position values and is able to provide or update additional informations, stored in the encoder.
The CAN Heidenhain Interface has been developed for the H1 VFPS experiment, where Heidenhain linear encoders of type LC481 are used for measurement of the absolute Roman Pot positions.

[1)] EnDat® Schnittstelle V2.1, Heidenhain Dokument Nr. D297403-01-A-01, 1999
EnDat® is registered trademark of Dr.Johannes Heidenhain GmbH
Dr.-Johannes-Heidenhain-Straße 5
83301 Traunreut, Germany

Block Diagram

Fig.1 shows a block diagram of the CAN Heidenhain Interface. Main Part of the circuit is a micro controller of type PIC18F448 (Microchip) with integrated CAN interface. It is connected to the CAN bus by a dedicated driver chip PCA82C250 (Philips).

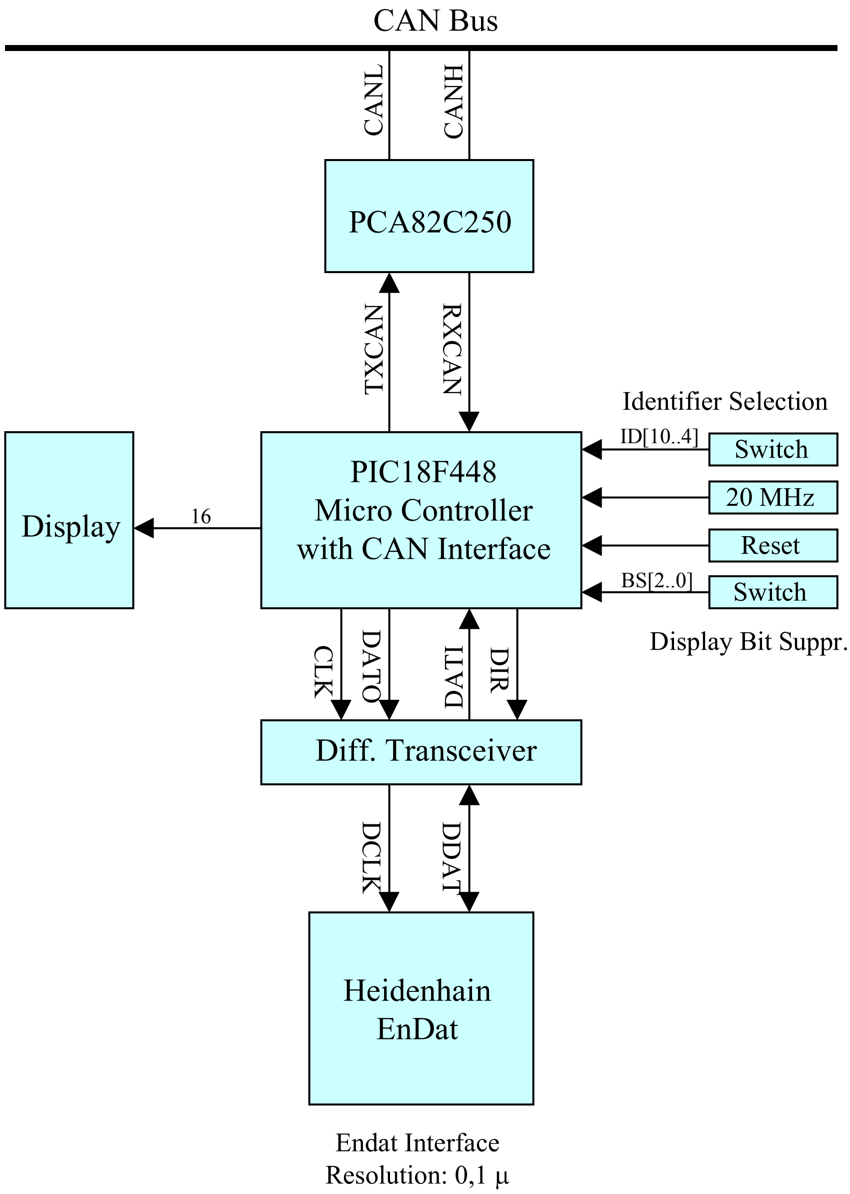


Fig. 1: CAN Heidenhain Interface Block Diagram

The connection to the encoder consists of only two signals: the clock, which is provided by the controller, and a bi-directional data line, on which commands and parameters are transmitted to the encoder and position informations are received. After power-on or external reset the controller reads some useful parameters from the encoder like serial number, resolution and data format. So it automatically adapts to encoders of different configuration.

For example the linear encoder LC481 with a resolution of 0,1 µ sends a string of 29 bits, consisting of one Start Bit, one Alarm Bit, 22 position bits and five check bits.

The controller stores the position information in 3 bytes. So it is able to handle up to 24 position bits. If the Alarm Bit is set, the controller additionally reads an Alarm Byte and a Warning Byte, which give informations about problems of the encoder.

The actual position is displayed on the front panel by 16 LED's grouped to four hexadecimal digits. A 3 bit rotary switch on the board determines the number of least significant bits, which are suppressed in the display. In the appendix a table is given, which shows the different display options for the encoder LC481.
Also after power-on or an external reset the controller reads a 7 bit switch, which defines the Identifier for CAN messages.
The definition of Request Message and Response Message is described in the following chapter.

CAN Higher Level Protocol

For computer access the widely used CAN bus interface has been implemented. Devices connected to a CAN bus have to follow a so called CAN Higher Level Protocol, which includes the Baud rate and the Identifier definition. For the CAN Heidenhain Interface Module the following parameters are fixed by firmware:

- Baud Rate: 125 kHz
- Identifier: 11 Bit Standard: ID10..ID0
- Identifier Definition: Device Type: ID10..ID8
  Device Instance: ID7..ID4
  Device Function: ID3..ID0

The Device Type (3 bits) and the Instance Number (4 bits) are selected on the board by means of two rotary switches.
Type Number and Instance Number defined for the H1 Forward Proton Spectrometer are listed in the Appendix.
Three functions are defined:

- Function F1: Read Heidenhain Parameter
- Function F2: Read CAN Parameter
- Function F3: Transparent Access to EnDat

All messages are data messages (RTR Bit = 0) with at least one data byte (argument).
The first two Functions have one argument each, which defines the type of parameters to be read. Function Nr. 3 needs four arguments and has been implemented for experts, which have experience with the EnDat interface and know the EnDat manual.
The controller responds to the requests by sending a data message with an identifier, containing the same Type Number and Instance Number, and the Function Number incremented by 8.
In the following the functions and the controllers response are described in more detail.

Function F1

Identifier	RTR	DLC	Byte 1	Function
$TI1	0	1	1	Read Position
$TI1	0	1	2	Read Serial Number
$TI1	0	1	3	Read Resolution
$TI1	0	1	4	Read Data Format

T: Type Number (3 Bit)
I: Instance Number (4 Bit)

Depending on the argument, function F1 generates the following response:

F1(1) Read Position:

Identifier	RTR	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
$TI9	0	7	1	**PosH**	**PosM**	**PosL**	**AL1**	**AL2**	**Warn.**

Pos. High:	Position Bit P23..P16	
Pos. Med:	Position Bit P15..P08	
Pos. Low:	Position Bit P07..P00	
Alarm AL1:	A0:	Time out: EnDat does not respond to a request
	A1:	Time out: EnDat does not reset data line after transfer
	A2:	Parameter read after reset failed
	A3:	Check Sum Error
	A7:	EnDat Alarm Flag
Alarm AL2:	A0:	Light Failure
	A1:	Signal Amplitude too low
	A2:	Position Failure
	A3:	Supply Voltage too high
	A4:	Supply Voltage too low
	A5:	Supply Current too high
	A6:	Battery Change necessary
Warn:	A0:	Frequency Failure
	A1:	Temperature too high
	A2:	Light Regulation Failure
	A3:	Battery Charge too low
	A4:	Reference Point not reached

If one of Bits A0..A2 of **AL1** is set, position read-out does not work, and the bytes **PosH**, **PosM** and **PosL** are set to $FF.
If Bit A3 is set, the check sum, sent by the EnDat interface, differs from the check sum, calculated for the received parameter or position. So the received informations could be wrong.
If Bit A7 of **AL1** is set, an internal encoder problem has been detected, and at least one bit of **AL2** or **Warn** should be set, indicating the type of the problem. Not all alarms of **AL2** and all warnings of **Warn** are supported by all encoders. Please consult the respective manuals for more details.

F1(2) Read Serial Number:

Identifier	RTR	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
$TI9	0	7	2	**SN5**	**SN4**	**SN3**	**SN2**	**SN1**	**SN0**

SN5:	ASCII Code of leading literal
SN4:	Bits S31..S24 of Serial Number
SN3:	Bits S23..S16 of Serial Number
SN4:	Bits S15..S08 of Serial Number
SN1:	Bits S07..S00 of Serial Number
SN0:	ASCII Code of trailing literal

F1(3) Read Resolution:

Identifier	RTR	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
$TI9	0	5	3	**RS3**	**RS2**	**RS1**	**RS0**

RS3:	Bits R31..R24 of Resolution in nm
RS2:	Bits R23..R16 of Resolution in nm
RS1:	Bits R15..R08 of Resolution in nm
RS0:	Bits R07..R00 of Resolution in nm

F1(4) Read Data Format:

Identifier	RTR	DLC	Byte 1	Byte 2	Byte 3
$TI9	0	3	4	**DF1**	**DF0**

DF1: Bit D15=1, Bits D14..D08 of number of Bits for position value
DF0: Bits D07..D00 of number of Bits for position value

Function F2: Read CAN Parameter

Identifier	RTR	DLC	Byte 1	Function
$TI2	0	1	1	Read CAN Error
$TI2	0	1	2	Read Firmware Version Number

T: Type Number (3 Bit)
I: Instance Number (4 Bit)

Function **F2** generates the following response:

F2(1) Read CAN Error:

Identifier	RTR	DLC	Byte 1	Byte 2	Byte 3	Byte 4
$TIA	0	4	1	**CEB**	**TEC**	**REC**

CEB:	C0: Error Warning Bit	(C1 or C2 are set)
	C1: Receiver Warning Bit	(95 < REC < 128)
	C2: Transmitter Warning Bit	(95 < TEC < 128)
	C3: Receiver Bus Passive Bit	(127 < REC)
	C4: Transmitter Bus Passive Bit	(127 < TEC)
	C5: Transmitter Bus Off Bit	(255 < TEC)
	C6: Receiver Buffer 1 Overflow Bit	
	C7: Receiver Buffer 0 Overflow Bit	
TEC:	Transmitter Error Counter	
REC:	Receiver Error Counter	

For more details of the CAN error handling please consult the CAN Bus Specification manual (CAN Specification, Version 2.0, Robert Bosch GmbH, 1991).

F2(2) Read Firmware Version Number:

Identifier	RTR	DLC	Byte 1	Byte 2	Byte 3
$TIA	0	3	2	**VNH**	**VNL**

VNH: MS-Byte of Version Number
VNL: LS-Byte of Version Number

After Power-on or after Reset the version number is displayed on the front panel for about two seconds.

Function F3: Transparent Access to EnDat

Identifier	RTR	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Function
$TI3	0	4	**Mode**	**MRS**	**DatH**	**DatL**	Transparent Access

T: Type Number (3 Bit)
I: Instance Number (4 Bit)

Mode: Mode of EnDat Access
MRS: MRS Code or Address of EnDat Access
DatH: Data Bits D15..D08 of EnDat Access
DatL: Data Bits D07..D00 of EnDat Access

Function **F3** generates the following response:

F3 Transparent Access:

Identifier	RTR	DLC	Byte 1	Byte 2	Byte 3	Byte 4
$TIB	0	4	**Mode**	**MRS**	**DatH**	**DatL**

Mode: Mode of EnDat Access
MRS: MRS Code or Address of EnDat Access
DatH: Data Bits D15..D08 of EnDat Access
DatL: Data Bits D07..D00 of EnDat Access

Function F3 enables the experienced user to read all available parameters from the encoder and to write parameters to the encoder. For more details please consult the EnDat Interface manual[1)].

Appendix

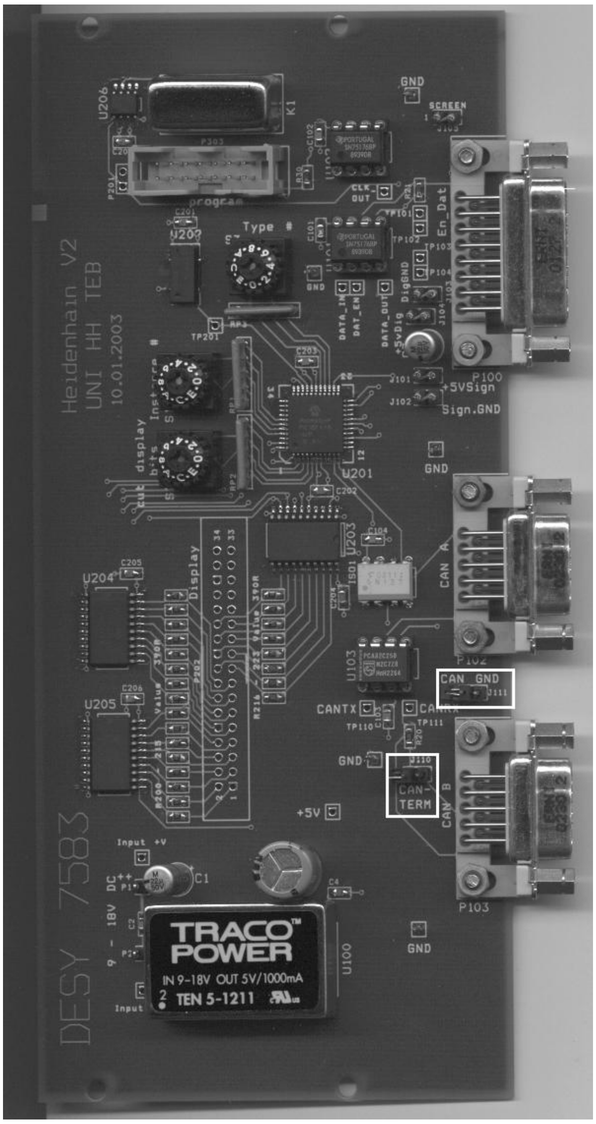
1) Jumpers

J110: CAN Bus Termination
The CAN bus has to be terminated at the last node of the bus. The termination is activated by closing jumper J110.

J111: CAN GND
By means of J111, the ground potential of the module can be connected to the ground line of the CAN bus

The jumper locations are marked by white rectangulars in the following Board Layout

2) Board Layout



3) Rotary Switches

S3: **Type #:** Type Number Selection (3 Bit): Range: 0..7

S1: **Instance #:** Instance Number Selection (4 Bit): Range: 0..15

S2: **Cut Display Bits :** Number of least significant Position Bits suppressed in Display (3 Bit): Range: 0..7

Encoder LC481 provides 22 Position Bits P0..P21 with a resolution of 0,1 μ.
The following table shows displayed Position Bits, resolution and range for different

positions of rotary switch S2:

S2	**Position Bits**	**Resolution [μ]**	**Range [mm]**
0	P0..P15	0,1	6,6
1	P1..P16	0,2	13,1
2	P2..P17	0,4	26,2
3	P3..P18	0,8	52,4
4	P4..P19	1,6	104,9
5	P5..P20	3,2	209,7
6	P6..P21	6,4	419,4
7	P7..P21	12,8	419,4

Changes of the switch position are effective only after pushing the reset button on the front panel or after power-on.

4) Front Panel

a) Connectors

EnDat: 15 pin male DSUB connector for EnDat Interface

CAN: Two 9 pin male DSUB connectors for CAN Bus

b) LED's

Error: LED indicating, that at least one of the Alarm Bits of **AL1** has been set.

Position: 16 LED's, grouped to four hexadecimal digits and showing the actual position according the display option, selected by switch **S2**.

c) Push Button

Reset: This Push Button resets the micro controller and the EnDat interface. If the module is connected to the VFPS Watch Dog by CAN bus, it can also be reset by pressing the Reset Button on the Watch Dog front panel.
Changes of the rotary switches are effective only after reset or after power-on.

5) Type and Instance Numbers in H1 Forward Proton Spectrometer

Type	Instance	Identifier	Detector
3	0	$30x	P64H
3	1	$31x	P80H
3	2	$32x	P80V
3	3	$33x	P90V
3	4	$34x	P220
3	5	$35x	P226

x = Function Number