H1 Very Forward Proton Spectrometer

CAN Heidenhain Interface V2

Preliminary!
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\textsuperscript{1}). EnDat (Encoder Data) 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.

\textsuperscript{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
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 µm 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**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>RTR</th>
<th>DLC</th>
<th>Byte 1</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>STII</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Read Position</td>
</tr>
<tr>
<td>STII</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>Read Serial Number</td>
</tr>
<tr>
<td>STII</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>Read Resolution</td>
</tr>
<tr>
<td>STII</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>Read Data Format</td>
</tr>
</tbody>
</table>

T: Type Number (3 Bit)
I: Instance Number (4 Bit)
Depending on the argument, function F1 generates the following response:

**F1(1) Read Position:**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>RTR</th>
<th>DLC</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Byte 4</th>
<th>Byte 5</th>
<th>Byte 6</th>
<th>Byte 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST19</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>PosH</td>
<td>PosM</td>
<td>PosL</td>
<td>AL1</td>
<td>AL2</td>
<td>Warn.</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>RTR</th>
<th>DLC</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Byte 4</th>
<th>Byte 5</th>
<th>Byte 6</th>
<th>Byte 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST19</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>SN5</td>
<td>SN4</td>
<td>SN3</td>
<td>SN2</td>
<td>SN1</td>
<td>SN0</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>RTR</th>
<th>DLC</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Byte 4</th>
<th>Byte 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST19</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>RS3</td>
<td>RS2</td>
<td>RS1</td>
<td>RS0</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Identifier</th>
<th>RTR</th>
<th>DLC</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TI9</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td></td>
<td>DF1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DF0</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Identifier</th>
<th>RTR</th>
<th>DLC</th>
<th>Byte 1</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TI2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Read CAN Error</td>
</tr>
<tr>
<td>$TI2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>Read Firmware Version Number</td>
</tr>
</tbody>
</table>

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

Function F2 generates the following response:

F2(1) Read CAN Error:

<table>
<thead>
<tr>
<th>Identifier</th>
<th>RTR</th>
<th>DLC</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Byte 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TI2</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td></td>
<td>CEB</td>
<td>TEC</td>
</tr>
</tbody>
</table>

**CEB:**
- C0: Error Warning Bit (C1 or C2 are set)
- C1: Receiver Warning Bit (95 < REC < 128)
- C2: Transmitter Warning Bit (55 < 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


F2(2) Read Firmware Version Number:

<table>
<thead>
<tr>
<th>Identifier</th>
<th>RTR</th>
<th>DLC</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TI2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td></td>
<td>VNH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VNL</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Identifier</th>
<th>RTR</th>
<th>DLC</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Byte 4</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TIB$</td>
<td>0</td>
<td>4</td>
<td>Mode</td>
<td>MRS</td>
<td>DatH</td>
<td>DatL</td>
<td>Transparent Access</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Identifier</th>
<th>RTR</th>
<th>DLC</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Byte 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TIB$</td>
<td>0</td>
<td>4</td>
<td>Mode</td>
<td>MRS</td>
<td>DatH</td>
<td>DatL</td>
</tr>
</tbody>
</table>

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) Jumper

**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:

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

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Instance</th>
<th>Identifier</th>
<th>Detector</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>$30x$</td>
<td>P64H</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>$31x$</td>
<td>P80H</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>$32x$</td>
<td>P80V</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>$33x$</td>
<td>P90V</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>$34x$</td>
<td>P220</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>$35x$</td>
<td>P226</td>
</tr>
</tbody>
</table>

$x =$ Function Number