

 PAUL SCHERRER INSTITUT		Projekt / <i>Project</i> SwissFEL
Titel / <i>Title</i> EPICS Support for EIB74x		Dokument Nummer / <i>Doc. Identification</i> -
Autor / <i>Author</i> Rolf Follath Mitauteur(en) NA <i>Co-Author(s)</i>		Versionsnummer / <i>Review Index</i> 1.1

Zusammenfassung / *Summary*

EPICS support for the Heidenhain EIB74x interface box for incremental encoders. The box handles up to four incremental encoders with voltage (1 V_{ss}) or current ($11\text{ }\mu\text{A}_{ss}$) encoder signals. The signals are interpolated with a depth of 4096 and an signal corrections can be applied. Either the onboard correction or the Heydeman correction algorithm. The encoders can have a single or several distance codes index marks.



Organisation	Anzahl	Verteilername <i>Name of distribution</i>	Überprüft <i>Approved</i>	Verantwortlich <i>Responsible</i>	Datum / Unterschrift <i>Date / Signature</i>
PSI	1				
PSI	1				
PSI	1				
PSI	1				
PSI	1				
PSI	1				
PSI	1				
Release Traceability: -			Datum / Unterschrift <i>Date / Signature</i>		
File Name:	EIB.tex	Unterschrift Autor			
Date:	May 8, 2026	<i>Signature Author</i>			

Directory: /follath_r/progs/eib/eibApp/latex

Contents

1 General	3
1.1 Git	3
2 Startup	3
2.1 Startup procedure	3
2.2 Configuration file EIB.CFG	4
3 Monitoring	4
3.1 Connectivity	4
3.2 Signal loss	5
4 Heydeman correction	5
4.1 Take data	5

1 General

The softioc was created with the standard epics application `makeBaseApp`.

- The installation directory was set to `\ioc\EIB` in `CONFIG_SITE`.
- The `EPICS_HOST_ARCH=linux-x86_64`.
- Only `sls-lc8` and `sls-lc9` have write access to this directory and can compile and deploy the code. All others in the west have only read access.
- `adl-` and `ui-`files are put into folder `adl` and are deployed to `\ioc\EIB\adl`
- the shared library is put to folder `lib64` and is deployed to `\ioc\EIB\lib64`
- Writing the encoder data files may become a problem that may be solved by writing to `\data\EIB`

Heidenhain provided software to communicate with the EIB74x board. Most probably I used the shared library `libeib7_64.so` with size 216573 from version 753013-10-01-00, although the version 677372-07-00-00 has the same size.

1.1 Git

The code for the softioc is in gitea:

```
git remote -v
ssh git@gitea.psi.ch:optics/eib700.git (fetch)
ssh git@gitea.psi.ch:optics/eib700.git (push)
```

Note that a second implementation was made according to PSI-standards on

```
git@git.psi.ch:epics_iocboot_sf/satop11-cpcl-eib700.git
```

2 Startup

2.1 Startup procedure

The display files `Eib.adl` and `EIB.ui` have a button "StartSoftIOC" to set the `LD_LIBRARY_PATH`, `EPICS_HOST_ARCH` and start the softioc.

The softioc itself is started by calling the bash-script `st.cmd` which checks the PV `*:running` whether an other instance is already running. If yes it refuses to start a second time.

If not it starts the softioc which performs several initialization steps:

1. Call procedure `EIB_Start()` which performs a basic initialisation by calling `EIB_Init()`.
2. Search for `EIB.CFG` and `./EIB.CFG` and terminate if it can't be found.
3. Read the keywords from `EIB.CFG` and initialize the corresponding variables.

4. Start the tasks

- (a) `EIB_Start_ConnectTask()` to connect to the EIB74x .
- (b) `EIB_StartMonitorRefTask` to perform and monitor an index search .
- (c) `EIB_Start_PingTask` to check the connection to the EIB74x.

5. Trigger the ConnectTask.

2.2 Configuration file EIB.CFG

Name	Meaning
Hostname	Name of the EIB74x (default is EIB741-639546320.psi.ch)
Ex_Name	Name of the channel (e.g. RON, ROD, X05DA)
Ex_iface	Encoder type, 1 = incremental
Ex_RefMarks	what about index marks: 0=none, 1=one, 2=distance coded
Ex_EncType	0=linear , 1 = rotary
Ex_Lines	How many lines per mm or revolution, 25000 for X05DA, 36000 for RON
Ex_RefSep	Separation of two reference marks if distance coded, 1000 for X05DA
Ex_SigV	must be +1, see manual
Ex_Comp	onboard signal correction: 0=Off, 1=On
Ex_HeyComp	use Heydemann correction table: 0=No, 1= Yes
Ex_HeyFile	Name of file with Heydemann corrections
Ex_Dir	Counting direction: +1=standard, -1=reverse

x denotes the channel number 0,1,2,3.

3 Monitoring**3.1 Connectivity**

- The task `EIB_Start_PingTask` monitors the connection to the EIB74x hardware and writes the status into the PV `EIB:CONNECTED`. If the connection is interrupted it periodically tries to reconnect.
- After reestablishing the connection, the status of the index marks is read and the HOMED flags of the encoders are set accordingly.
- If the EIB was powercycled, the HOMED flags are set to FALSE.

The same procedure applies after a restart of the softioc. By this, the necessity of an new index search is avoided.

3.2 Signal loss

- When an encoder is disconnected while the softioc tries to read it, the EIB sets the flag "signal amplitude error" in the encoder status. The softioc indicates this in the `EIB:CH_x_SIGNALERROR`.
- Unfortunately, this works only when the encoder readout happens during the time when the encoder is disconnected. If the encoder is removed and reconnected between two readouts, the encoder status does not reflect this in the "signal amplitude error" flag.
- What can be seen is that the status reports an "Frequency exceeded" in that case. This could be reported to the user with a recommendation to start an index search.

4 Heydeman correction

The EIB has an onboard signal correction algorithm which works quite well. It is activated during startup by setting the keyword `Ex_Comp` in the configuration file.

Additionally, the Heydeman signal correction algorithm was implemented in the softioc. It is activated with the keyword `Ex_HeyComp`

The Heydeman correction is split into steps:

1. Take data pairs from the encoder signals and write them into a file.
2. Call the python-program `do_hey` which reads the encoder data and generates a correction file.
3. Read the correction file and turn the correction on.

4.1 Take data

Encoder data are read by `EIB_Read()` every time the PV `*:CH_x_VALUE` is processed. The procedure feeds them into a ring buffer (`enc.History`) when the PV `CH_x_ADDPOINTS` (\equiv `enc.Addpoints`) is set.

The sample frequency can be selected by the corresponding SCAN-field `*:CH_x_VALUE.SCAN`.