



Bridgeport Instruments, LLC, 11740 Jollyville Rd., Ste 500, Austin, TX 78759,  
[www.BridgeportInstruments.com](http://www.BridgeportInstruments.com)

By Michael Momayezi  
Phone 512-553-9933  
Fax 512-553-9934  
email momayezi  
@bridgeportinstruments.com

# eMorpho Users Manual

## June, 2020

### Summary

The eMorpho is a high-performance multi-channel analyzers (MCA) for measuring gamma-ray radio-activity with a scintillator.

It includes an high voltage supply and plugs directly onto a photomultiplier.

This document describes how to use the eMorpho from the graphical user interface and from Python scripts. Since all commands are formatted as human readable JSON strings, developers will find they can program the eMorpho in any language they want.

### Feature summary

- Histogram:  $4K \times 32$ ; Loss-less two-bank mode with  $2 \times 2K \times 32$ ;
- More than 3Mcps histogramming rate
- Suitable for all scintillators via programmable integration time
- List mode, pulse shape capture
- On-the-fly pulse shape discrimination
- Host-supported gain stabilization via lookup tables
- USB 2.0 low-speed interface
- FPGA customization possible for application specific product
- Positive and negative operating voltage and different PMT pinouts.
- Power consumption is only 300mW (5V@60mA)

### Popular FPGA firmware options

- Loss-less dual bank list mode buffer with 2047 events per buffer.
- 100ms time slice operation with 20-deep buffer, events and 1k histogram per slice.

### Popular software options

- Stabilizer for gain-stabilization, including using an LED
- Radiation Portal mode with automatic background tracking and programmable alarm

### Table of Contents

1. Supporting Documentation
3. Energy Spectrum
4. Radiation Portal Monitor

5. Analog
6. Calibration
7. Gain Stabilization
8. Mechanical and Pinouts
9. Ordering

## 1. Supporting Documentation

**Open-source software** The software is open source and mostly written in Python. Bridgeport Instruments provides a software installer, which by default will create a C:\BPISoftV3 directory. That folder includes Python 3.7 with three added packages: ZMQ (www.zeromq.net), wxPython and matplotlib v 3.2. ZMQ is used to implement the client-server behavior and it is accessible from more than 40 programming languages. Matplotlib is used for the graphics interface, and wxPython is used to create a traditional user interface with pull down menus.

The wxMCA folder that contains the MCA software can be placed anywhere on the hard disk. Inside is a folder called documentation and the best starting page is wxMCA/documentation/english/introduction/introduction.html which you can open in any web-browser. This set of linked documents contains a description of every control variable and every data field used by the eMorpho.

## 2. Getting started

### 2.1 Using the USB interface

For USB communication the eMorpho uses libusb0.1 (Linux) and libusb\_win32 on Windows 10. The Windows 10 libusb dll is digitally signed. When the MCA hardware is present during the installation, the installer will automatically link the eMorpho to libusb\_win32.

However, it is also possible to simply copy the BPISoftV3 folder onto the C: drive and install the Windows USB-driver manually. This needs to be done only once, and Windows will later recognize any other eMorpho, by its USB vendor ID of 0x1FA4 and the USB product ID of 0x6001.

Open the Windows Device Manager and connect the eMorpho to a USB port. Wait until it appears in the Device Manager as an unknown device of type eMorpho. Other MCA from BPI will all have a USB vendor ID of 0x1FA4 but different product ID; eg 101, 103, 201, or 203.

Open the C:\BPISoftV3 folder and launch zadig-2.4.exe. Use Options/List All Devices to

refresh the device list and select the select the device with the BPI vendor ID of 0x1FA4. To the right of the green arrow, select libusb-win32 (v 1.2.6.0) and click the big 'Install Driver' button below. Note that it may take up to 20 seconds for the screen to update - be patient. Once the 'Driver Installation Successful' message appears you can close the window. In the Device Manager you will now see the eMorpho listed under 'libusb-win32 devices'.

### 2.2 Launching the software

**First launch the MCA Data Server.** Look inside the wxMCA folder. Under Windows, double-click on run\_mds.cmd. Under Linux, launch wxMCA/mds/mds\_server.py making sure you use a python 3.6 or higher installation that also includes ZMQ, wxPython and matplotlib.

If the MDS reports no MCA found, the OS may have been slow in enumerating the MCA on the USB bus. Kill the MDS and launch it again.

**Then launch the User Interface.** Under Windows, double-click on run\_wxMCA.cmd. Under Linux, launch wxMCA/wxGUI/MCA\_Main.py

**The items in the menu bar are mostly self-explanatory.** You can view results and edit instrument settings in a spread-sheet environment. To send the changed values to the instrument, you need to click on File →to MCA.

## 3. Energy Spectrum

**The MCA provides fast and accurate measurements of energy spectrum and count rates.** To acquire an energy histogram, select Display→Histogram from the menu bar. In response, a histogram panel will open. On that panel use "New" to erase the old histogram and count rate data and begin a new acquisition. The panel does not automatically refresh, so click the "Refresh" button to get an updated energy spectrum.

Use "Save" to append the energy histogram and count rate data to a default data data file. The file name is automatically generated and is of the form xyz\_histo\_status.json where xyz is the detector serial number. By default that file is located in wxMCA/user/mca3k/data.

Alternatively, use the Display→Save Histogram As to save the histogram where you want it.

### 3.1 Adjusting operating voltage

**You need to adjust the operating voltage to set the maximum measurable energy.**

**Then adjust the digital gain to achieve the desired MCA calibration in keV per bin**

The maximum measurable energy is independent of the number of histogram bins used. Users need to adjust the operating voltage to change the gain of the PMT. In the table below we recommend maximum measurable energies for different applications.

<i>Max. Energy</i>	<i>Comment</i>
1.0MeV	High gain for measuring low energies down to 20keV. Good for measuring contamination with Cs-137 and Cs-140 (fresh fission product), or Radium isotopes.
2.0MeV	Measure NORM up to K-40 at 1460keV
3.0MeV	Wide range covers all naturally occurring gamma-rays, even to Tl-208 at 2615keV

Table 1: Cs-137 calibration for different tasks; NORM: Naturally occurring radioactive materials

<i>Max. Energy</i>	<i>662keV pulse height</i>
1.0MeV	561mV
2.0MeV	281mV
3.0MeV	187mV

Table : Cs-137 pulse height for different maximum measurable energies.

All the controls for the calibration process can be found in Edit→fpga\_ctrl and in the shorter Edit→fpga\_dsp spreadsheet.

**How to determine the maximum measurable energy.** Set `ha_run` to 1 to measure the pulse height. The maximum measurable pulse height is 900. Start a new histogram and record the peak position (P) of a known isotope; eg E=662keV of Cs-137. The maximum measurable energy is computed as

$$E_{\max} = E \cdot \frac{900}{P}$$

Enter a new operating voltage in the high-voltage field and hit enter. Go back to the `Histogram` panel and click `New` to start a new acquisition. If you change the operating voltage by 1%, the PMT gain will change by about 6%.

**Now adjust the digital gain to achieve the desired MCA calibration in keV per bin.**

Once the operating voltage has been established, set `ha_run` back to 0, and start a new histogram

Determine the measured peak position (P) and use the desired peak position (E) to determine the new *digital\_gain*

$$dg_{\text{new}} = dg \cdot \frac{E}{P}$$

With that the detector has been calibrated. See the calibration section 6 for a much more detailed description.

### 3.2 Count rate measurement

**Observe how the count rate accuracy improves with measurement time.** The MCA Data Server reports count rates together with their statistical errors. This gives users a useful tool. Manually, or programmatically, they can end a measurement precisely when the desired accuracy has been reached, which saves time and money.

The error (in %) is computed from the number of events as  $100 \cdot \frac{1}{\sqrt{N}}$ , where N is the number of events. This is called the statistical 2-σ error. The true count rate lies within this error range with a 95% probability.

Note that the MCA corrects the recognized count rate using the known dead time per event. Hence the reported count rate is greater than the recognized count rate. The reported count rate is also an accurate estimate of the true number of counts per second. In fact the reported count rate should match the true input count rate with about 1% accuracy (systematic error) for input count rates up to 100kcps. The statistical count rate error is computed correctly using the number of recognized events.

### 3.3 Histogramming speed

**The MCA has a non-extendable dead time equal to the hold\_off\_time per recognized event.** For NaI(Tl) between 5°C and 65°C the

eMorpho recommends a fixed hold off time for NaI of  $1.2\mu\text{s}$ . The resulting throughput is shown in the next figure.

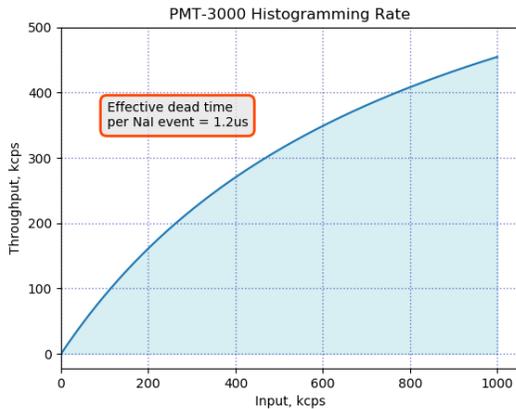


Fig. 1: Expected histogramming rate vs input count rate for a hold\_off\_time =  $1.2\mu\text{s}$ ; (ie typical for NaI(Tl) at room temperature.

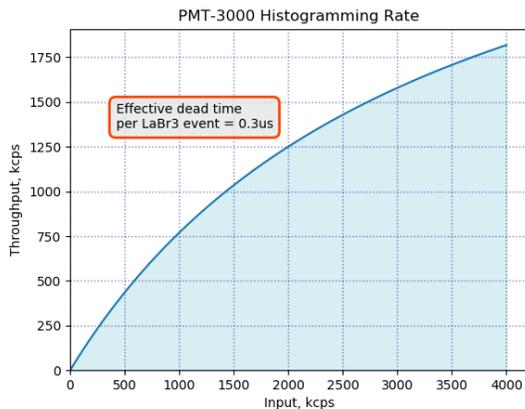


Fig. 2: Expected histogramming rate vs input count rate for a hold\_off\_time =  $0.3\mu\text{s}$ ; (ie typical for LaBr<sub>3</sub>).

### 3.4 High count rates

**There may be a small gain shift at high count rates.** Users should be aware that at input count rates above a few 10kcps there may occur small gain shifts in the energy spectrum. Typically, at 50kcps the gain may fall by about 1%, but the exact gain shift will depend on the PMT that is used on the detector.

## 4. Radiation Portal Monitor

### Non-standard software option

**The MCA can act as a radiation portal monitor with background tracking and a programmable alarm function.** In a Radiation Portal Monitor (RPM) the data acquisition unit must perform a number of tasks:

- Continuous background measuring;
- Deliver an alarm within a few seconds after the time of closest approach by a source;
- Keep a few seconds of alarm history so that a polling host will not miss an alarm;
- Automatically reset the system on a continuous alarm to remain operational;
- Support a programmable alarm threshold based on a false-alarm rate;
- Support an adjustable background averaging time, eg 30s for a big gamma-ray detector vs 5 minutes for a neutron detector,
- Be able to recognize passing sources without the aid of an occupancy detector.

The monitor software implements that functionality by collecting data from the eMorpho in 100ms times slices. The time slices include count rate data and a 1K histogram. A 20-slice deep buffer in the FPGA ensures loss-less data delivery even if the host computer experiences a delay of up to 2s.

### 4.1 Theory of operation

In the discussion below, times are given in units of time slices. The fast processing software is called the mca\_monitor and it uses logger objects to compute the alarms and track the background. There can be multiple logger object per MCA, logging and alarming on the total gamma-ray count rate, external counts (eg from a neutron detector or dosimeter) and from multiple regions of interest. Typically a time slice is 100ms long and the software performs one RPM computation step every time slice.

**At first the software measures the background.** On start, after power on or a reset, the software processor begins to measure backgrounds. Until backgrounds are known with sufficient precision, the ability to alarm is disabled. This wait time is user programmable, cf **wait** in the RPM panel. Typically the wait time is a fraction of the background averaging time, eg 20% to 50%. A shorter wait time yields an active RPM earlier at an elevated risk for a false alarm until a full background averaging time has passed.

As long as there is no alarm, the events counted during one time slice are considered background events. Using the region of interest (**roi\_low**, **roi\_high**) in the RPM panel allows

the user to constrain the attention on a part of the energy spectrum.

The background counts per time slice are averaged using geometric averaging. Using the  $w = 1/Bck\_avg$  from the alarm panel the processor applies the formula:

$B_{n+1} = B_n + w \cdot (N - B_n)$ . The background averages are then stored in a 128-long FIFO. This way the instrument can look back 128 time slices to find an untainted background after an alarm has occurred.

We have  $0 \leq w \leq 1$  for the weight. If the background at time  $t = 0$  changes from  $B_0$  to  $B_1$ , The background average responds to a step function with an exponential function of  $B(t) = (B_1 - B_0) \cdot (1 - \exp(-t/\tau))$  where  $\tau = 1/w$ . The standard deviation of the averaged background, ie its noisiness, improves as if  $1/w = Bck\_avg$  samples had been averaged.

**What happens when a passing source causes an alarm?** The logger continuously computes a moving window sum of the last  $L$  time slices and compares that sum to the number of expected background counts during  $L$  time slices. For every time slice the logger computes the probability that the observed counts could have been caused by the known background, of the mathematics section above. It computes  $P = P(N \geq N_L | N_B)$ , with  $N_B = L \cdot B$ . If that probability is less than **epsilon** on the alarm panel, it will trigger an alarm.

**An alarm is typically raised no later than  $L/2$  time slices after the closest encounter.**

For a very strong signal, that alarm may be raised right at the leading edge of the  $L$  period; for a weak signal the alarm may be raised about  $L/2$  time slices after the source has passed the point of closest approach. In many applications the allowed latency  $L/2$  is about 2 seconds, which allows for a 4s summation time. At a time slice length of 100ms this means  $L = 40$ . In those applications, the time during which the radiation signal is detectable is 4s to 8s and simulations show that 4s to 6s summation times produce the highest sensitivity; ie lowest minimum detectable activity.

**No missed alarms.** The logger keeps a history of the alarm status with a typical maximum depth of 128 time slices. A polling client will

be informed if there was an alarm present in the last **history** time slices. This way a polling host may have a latency of 128 time slices (12.8s) and still will not miss an alarm.

As long as there is an alarm present in the alarm history FIFO, the logger will suspend background updates and use the oldest background average in its memory as the best estimator of the background.

**Automatic logger reset ensures the instrument remains functional when the background suddenly increases.** Consider the case of a radiation detection backpack. The wearer walks into a room where radioactive material is present, and this causes an alarm. After **history** time slices, the logger automatically resets and starts to accept the elevated radiation level as the new background. After a **wait** period (30s typically) the logger will again be ready to alarm if the operator suddenly encounters an even higher radiation level.

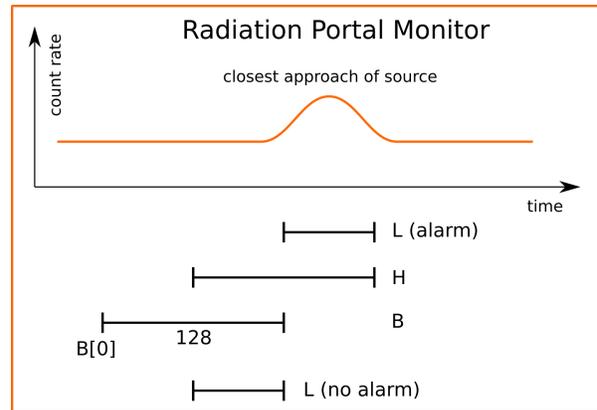


Fig. 3: Programmer's model of the Radiation Portal Monitor software.

## 5. Analog

### 5.1 usbBase

The usbBase combines a PMT operating voltage supply and an MCA data acquisition board with a structure as shown in the figure below.

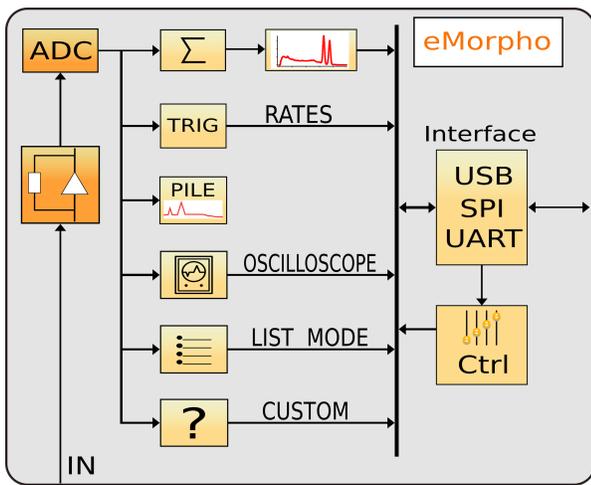


Fig. 4: The components of the eMorpho MCA.

The analog input of the eMorpho uses an I-to-V converter with four programmable gain resistors as shown in the figure below. A gain\_select of 0, 1, 2, 4, and 8 creates a transimpedance of 100Ω, 430Ω, 1100Ω, 3400Ω, and 10100Ω, respectively.

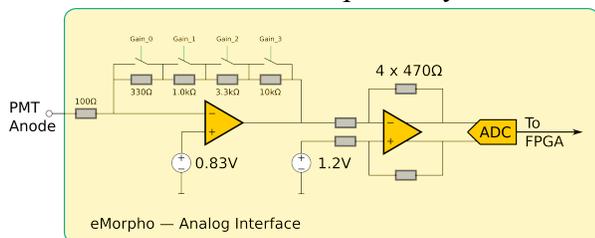


Fig. 5: The input amplifier of the eMorpho.

## 6. Calibration

### 6.1 Theory of operation

The emorpho uses a switched-gain input amplifier stage followed by an ADC with a 1V input range. The ADC samples the amplifier output voltage at a rate of typically 40MSPS. In the eMorpho, the sampling rate can reach 120MHz.

For proper operation, the amplifier output voltage must fall within the 1.0V input range of the ADC, as shown in the two figures below.

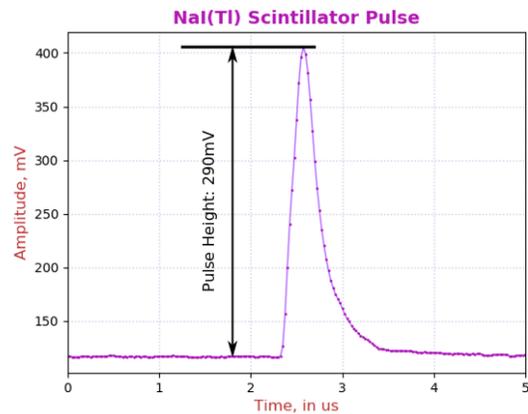


Fig. 6: A regular NaI(Tl) pulse.

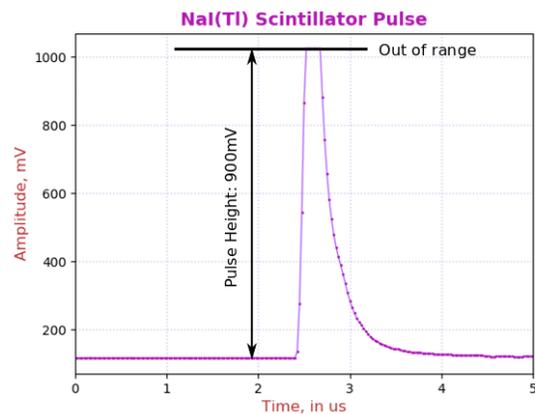


Fig. 7: An out of range NaI(Tl) pulse. Note how it cannot reach beyond 1023mV.

Pulse height is a measure of the difference between the maximum pulse voltage and the baseline from which it rises.

There is a built-in DC-offset of around 128mV and the maximum measurable voltage is shown as 1.023V. Hence the maximum practical signal pulse height range is around 900mV.

The first step of calibration at a chosen electronics gain, is to make sure that the maximum wanted energy can be measured by the ADC. At a fixed electronics gain, the applied operating voltage will determine the maximum measurable energy.

A very common calibration is to set the operating voltage such that the full-energy peak of Cs-137 (661.66keV) corresponds to a pulse height of 187mV. The maximum measurable energy is then  $900\text{mV}/187\text{mV} * 662\text{keV} = 3.18\text{MeV}$ .

After adjusting the high voltage to set the maximum measurable energy, the user can now vary the digital gain to map the entire energy

spectrum into an energy histogram of the desired size and calibration.

For example, we assume 40MHz ADC sampling rate and the recommended integration time of 1.2µs for NaI(Tl). We also assume that the Cs-137 662keV peak has an average pulse height of 187mV. In that case, choose a digital gain of 4700 to place the peak in the energy histogram at 662keV, ie have an MCA calibration of 1.0 keV per bin.

Below we describe the calibration steps in detail.

## 6.2 Calibration file

**Calibration controls are in autocal.json.** This file can be found in the `./rad_config/matplotlib_gui/config/` folder. Its contents are explained in the table below.

<i>Variable</i>	<i>Description</i>
cal_energy	The energy, in keV, for the calibration peak; eg 662 for Cs-137.
cal_pulse_height	The pulse height for the calibration peak; eg 187 for Cs-137 and 3.2MeV maximum measurable energy.
auto_update	0→ Off; 1→ Apply the new operating voltage and restart the mca after user clicks on "Calibrate" button.
keV_bin	Desired MCA calibration, as keV per MCA bin.
gain_exp	PMT's have a power law for their gain vs voltage function: $\frac{gain}{gain_0} = \left( \frac{HV}{HV_0} \right)^{gain\_exp}$ Typical values are 5.6 for R6231, 6.0 for CR105 and 7.5 for many other 10-stage PMT.
mass	The mass of the NaI crystal, in kg; used for computing the deposited dose rate from the count rate and deposited energy.

Fig. 7: The autocal.json data explained.

You can manually calibrate the detector, in which case you have more freedom on how to do it. If you want to use the **Calibrate** button on the **Sample** panel, then the software will always pick the tallest peak in the histogram, beyond 50 MCA bins, to use in the

calibration computations. This works best for isotopes with a well-separated medium-energy peak; eg Cs-137 and Na-22. Co-60 and K-40 are not recommended for the software-assisted calibration.

## 6.3 Step 1 of the calibration

**First, adjust the operating voltage to set the maximum measurable energy.** Open the **DSP** panel and set the **ha\_run** control to 1. Start a new histogram acquisition on the **Sample** panel by clicking on **New**. Wait until there are around 1000 counts in the peak maximum. Then click **Calibrate** on the **Sample** panel.

After 1 second, you will receive a message with the suggested new operating voltage. The software has applied the new operating voltage already if `auto_cal.auto_update` is set to 1.

There is some variation between the PMT and you may have to repeat a second time to achieve the desired accuracy. Note that even if you do not know the exact gain exponent of your PMT, the process will still converge. It may just take another iteration or two.

Here is how to compute the *cal\_pulse\_height* (*cph*):

$$cph = 900 * \frac{cal\_energy}{E_{max}}$$

## 6.4 Step 2 of the calibration

**Now change the digital gain to achieve the desired MCA calibration in keV per MCA bin.** From here on, the operating voltage and the electronic gain *gain\_select* should be kept constant. Instead we use the *digital\_gain* from the **Gain** panel.

Set the **ha\_run** control to 0 to return to normal energy measurements, and start a new histogram acquisition. Wait for sufficient accuracy and click **Calibrate**.

This time, you will receive a notice of the new *digital\_gain*.

The new value was computed from a linear formula

$$dg_{new} = dg \cdot \frac{cal\_energy}{E_{peak}}$$

and it will be accurate on the first try.

However, keep in mind that there is a statistical uncertainty in determining the actual energy of the histogram peak. You can compute the  $1\text{-}\sigma$  error from the fwhm (in %) and the number of net counts ( $N$ ) above background in the peak as

$$\varepsilon = \frac{\text{fwhm}}{2.3655 \cdot N}$$

## 7. Gain Stabilization

**Performance stabilization is more than gain stabilization.** For the eMorpho BPI offers a stabilizer software client to the MCA Data Server that provides gain stabilization which keeps the gain, the trigger threshold and the maximum measurable energy constant. NaI(Tl) is the most commonly used scintillator, and it also one of the most challenging. Its brightness and pulse shape change with temperature, while the application engineer wants to keep constant not only the MCA gain (keV per bin) but also the trigger threshold and maximum measurable energy that is independent of temperature.

The eMorpho stabilizer offers a user-programmable set of lookup tables to change the operating voltage and the digital gain to achieve this goal. In addition, for NaI(Tl) the stabilizer also varies the hold-off time with temperature to avoid retriggering on the tail end of the same pulse.

**The eMorpho stabilizer also offers LED-based gain stabilization to counter PMT aging.** Vacuum photomultiplier tubes lose gain over time, even under light to moderate loads. For example the popular R6231 PMT loses half its gain after an accumulated anode charge of about 50C. Distributed over one year, this equivalent to an average anode current of just  $1.6\mu\text{A}$ , with a 6% gain loss per month. With a built-in LED that injects light into the back of the photomultiplier, the eMorpho can gain-stabilize on the LED response and counteract the PMT aging.

### 7.1 General theory of operation

**All types of gain stabilization use lookup tables, and users can supply their own.** The standard code includes a 64-long float array with three look up tables (**LUT**). The MCA frequently determines the temperature and adjust the operating parameters according to the values found in the lookup tables.

**The LUTs depend on the scintillator and one DSP setting, namely the integration time.**

For NaI(Tl) the LUT supplied by the factory use an integration time of  $1.2\mu\text{s}$ .

**The NaI(Tl) integration time is fixed.** For NaI(Tl) the LUT supplied by the factory was derived for an integration time of  $1.2\mu\text{s}$ . That value remains fixed over the entire temperature range, as it provides the best energy resolution for small and large energies at all temperatures

**But the hold-off time has to be varied.**

NaI(Tl) pulses become very long when the crystal temperature falls below  $0^\circ\text{C}$ .

Consequently, the hold-off time has to be increased at low temperatures to avoid retriggering on the tail end of the same pulse. Since this phenomenon is determined by the scintillator physics, it has been implemented as an immutable function in the MCA, instead of a lookup table. The function is used if `arm_ctrl["fields"]["cal_scint"]==1`.

The implemented hold-off vs time function for NaI(Tl) is

$$\max(1.25, (0.634 + 1.27 * \exp(-T/32.26)))$$

in  $\mu\text{s}$  and with  $T$  being the temperature in  $^\circ\text{C}$ .

**How the LUT are constructed.** Developers may find that they need to construct the correct LUT for their detectors, as they may use a different scintillator or a different PMT. To this end the detectors needs to be run through a temperature cycle. The detector is recalibrated to desired specification at each temperature and the operating voltage, digital gain and possibly the LED value are recorded. Using interpolation on then constructs the LUT at the fixed temperature steps.

Keep in mind that many crystals don't allow for rapid temperature changes or they will crack.  $10^\circ\text{C}$  per hour is a safe bet for 50mm NaI.

Secondly, it takes a while for a detector to attain steady state. The vacuum PMT needs an hour and the crystal characteristic time depends on the material and the size. For a 50mm NaI it takes about half an hour, for a 76mm NaI it takes 1.5hrs. Hence, it is necessary for a precise measurement, to maintain the detector at a fixed temperature for 2 to 4 hours before moving to the next.

**Thermal equilibrium is necessary.** Keep in mind the the PMT and the scintillator both have different temperature coefficients. So, if the two are not in equilibrium, there is no easy way to provide gain stabilization. Hence an outdoor detector should be packaged with good thermal insulation to achieve thermal time constants of 2 hours or more – to ensure that all components are in a steady state situation. However, care must be taken not to insulate the electronics. Even if it only dissipates 300mW, that small amount of power stills needs to drain away to avoid self heating.

<i>LUT components</i>	
<i>Index: name</i>	<i>Description</i>
lut_len	Number of entries in the LUT; default is 19, 2..19 are allowed
lut_tmin	Minimum temperature in the lookup table; Typically -30°C
ut_dt	Temperature step size in the lookup table; Typically 5°C
lut_ov	Change of operating voltage vs temperature; Array with lut_len entries
lut_dg	Change of digital gain vs temperature; Array with lut_len entries
lut_led	Change of LED target vs temperature; Array with lut_len entries

The components of the gain stabilization look up tables

The stabilizer software and algorithms are described in detail in the [stabilizer](#) document.

## 8. Mechanical and Pin Outs

### 8.1 8-Pin Connector

The usbBase uses a Bulgin PX0447 mini-B USB connector for power and USB communication. Mating cables are the Bulgin PX0441 (USB-A) and PX0442 (USB mini-A) series cables. They come in lengths from 2m to 4.5m.

The usbBase uses a Switchcraft EN3P8MPX 8-pin connector for GPIO and optional UART communication. The mating connector is part of the EN3C8 series.

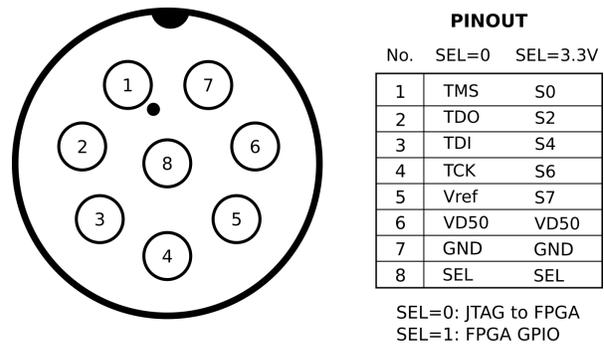


Fig. 8: Pinout of the EN3P8 connector.

## 9. Product and Part Numbers

### 9.1 Product numbers

The eMorpho MCA comes in different packages and speed grades (40MHz, 80MHz, and 120MHz) and with different voltage dividers for different PMT. Below we show the most popular part numbers; more are shown on our website.

<i>P/N</i>	<i>Description</i>
usbBase-4012-N81T	40MHz ADC, negative high voltage, R6231 PMT pinout; Used with standard NaI detectors sold by BPI.
usbBase-8012-P10T	80MHz ADC, positive high voltage, 10-dynode pinout; General-purpose device for all scintillators.
usbBase-12012-N81T	120MHz ADC, negative high voltage, R6231 PMT pinout; Used with LaBr <sub>3</sub> detectors for highest throughput.

Table : Part numbers.

### 9.2 USB-ID

On the USB bus devices are recognized by their Vendor ID (VID), Product ID (PID) and Serial Number (SN). The vendor ID for Bridgeport Instruments is 0x1FA4. The Product ID's are shown in the table below. Within a product the serial number is fixed, unless BPI makes a custom device that requires a non-standard driver. Note that simple extensions, such as adding a variable to the controls, does not require a new driver.

The BPI software recognizes individual devices by the unique serial number written into the FT245RL USB controller.

<i><b>P/N</b></i>	<i><b>PID</b></i>	<i><b>SN</b></i>
eMorpho	0x6001	eRC12345

Table : Product ID and USB bus serial numbers. The vendor ID is always VID=0x1FA4. The eRC serial number uses 4 digits until serial number eRC9999, then 5 digits.