mPMT Demo Manual

FD2 Working Group

December 2021
v1.0

1 Introduction

This document explains how to use the FD mPMT demo system to test PMTs and the associated electronics. In the document there will be a first part to explain how the system is composed and how to connect each component and then a brief explanation of the procedure to acquire and control the system will follow.

2 Overview and first operations

The provided system is composed by four main boards (fig. 1):

- 1 main board, that is a Nexys A7 from Digilent
- 1 custom PMOD adapter board, to interface with the channels
- 1 PMOD USB to UART converter
- FEB + HV board for each channel

The system components must be assembled the first time you use them. To do so follow the instruction below.

- Connect the PMOD USB to UART to the upper row of JA connector on the Nexys A7 and connect ist own USB cable
- Connect the Custom PMOD adapter to JC and JD and connect the USB cable to its own USB to UART adapter
- Connect the USB cable to the NEXYS A7 as shown in figure
- Connect one or two channels to the adapter according to your necessity using the Flexible cable in dotation.
- Turn on the system

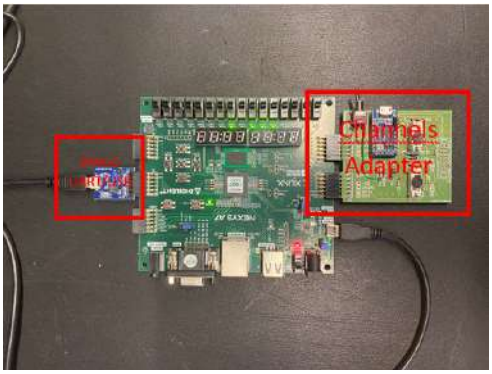

Figure 1: Demo system for PMT testing with FD mPMT Electronics

On the PMT side the necessary assembly is the connection of the HV board to the FEB and to the PMT itself. To acquire with the FEB you have to put two jumpers on the angled connectors between the two boards, see fig. 2. The PMT assembly is straighforward because there is an asymmetric pattern on the dynodes, so only one direction is correct.

The FPGA firmware is preloaded in the QSPI of the Digilent board, so the board starts working immediatly after booting. The bit file is however attached to this manual in case of boot failure.

In addiction to the hardware a set of Utility software are provided. There are 3 main softwares sections (Python scripts, V3.8). For each utility the manual will desscribe the presequisites, the functions and the output.

At the end of the manual you will find also a reference procedure to start the acquisition each time, from the booting of the board to the plotting of the data in real time.

In v1.0 of this manual no analysis tool will be provided, in future updates also them will be provided.

3 Software utilities

In this section a brief introduction on each utility software is provided.

3.1 HV-FEB control

This first tool let the user to control the HV parameters and the FE parameters. The connection is a serial bus but the protocol is ModBus so an utility is always needed to control the system. The USB connnection for this utility is the one on the channels adapter on the right of the picture 1.

The FE embeds a STM32 microcontroller already flashed. In case of problems with the system contact us for the recovery procedure of the microprocessor.

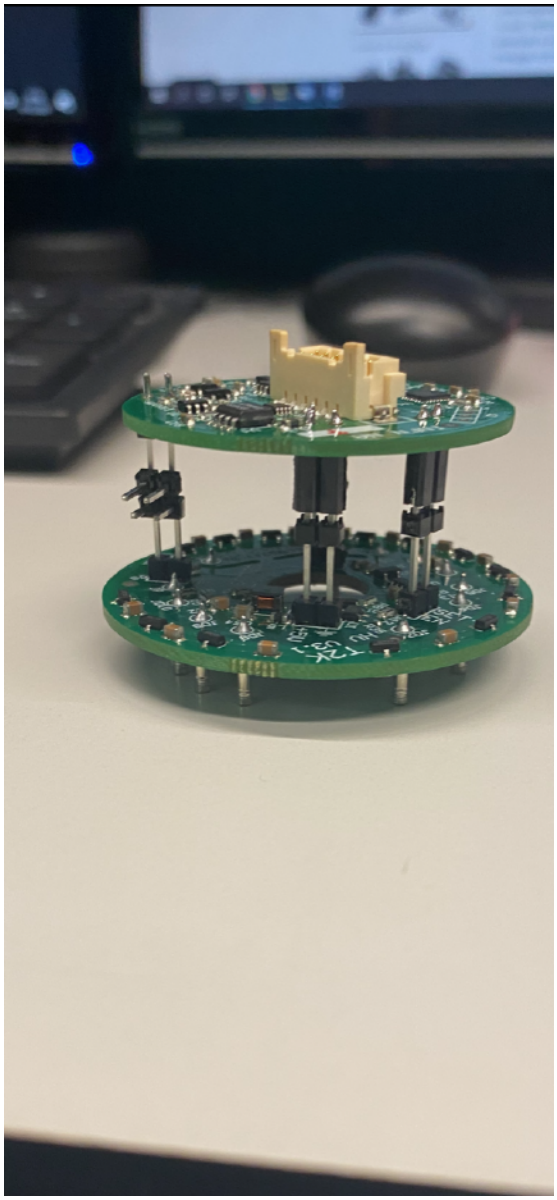

Figure 2: HV and FEB sandwich. The signal can be sent to the ADC closing the angular connections with two jumpers or acquired with an oscilloscope connecting a coaxial cable to the angular connector on the HV board. The signal is on the external pin of the couple and the ground is the internal one.

3.1.1 Prerequisites

The tool is a python script that requires some libraries to work.

Type the following commands to install all the libraries:

$ sudo apt-get install argparse os time sys cmd2 functools getpass numpy struct minimalmodbus==1.0.2

Many of them should be already installed on your laptop or PC. Just pay attenction that the tool works only with the specified version of minimalbodbus, so if already present a newer version remove it and install the necessary one for the tool.

3.1.2 Overview

To start the script the port parameter is required, so launch it according to yout port with the command

$python hv.py –port path − to − port

example *$python hv.py –port COM2* on Windows or *$–port /dev/ttyUSB*2 on Linux

For each section and for each command of the tool an help function is provided. Now, let's see how to use the tool. Just be sure to connect a FE board to the custom PMOD adapter, otherwise the tool will not find any valid connection.

To connect with a FE board the first command is the probe one. The probe command will give you an overview of the boards connected to the system with their address. The connected boards will be highlighted in greeen, while the not connected in red.

Once you are safe that at least a board is found, you can select that board to start the communication, with the command select, ex. select 2 to select the board at the address 2 if 2 was a greeen board.

After connected a useful command is the info one, that gives you some info on the board connected.

Another important command is the monitoring one, where you can specify the number of monitoring calls that you want to perform.

Ex. *mon 200* will print the monitoring parameters 200 times, once each second.

With *help–v* you will have access to the verbose help menu with all the available commands of the software.

- Address changes the Modbus address of the board, do not use this commands if not necessary
- Calibration starts an automatic routine to calibrate the HV board value with respect to the set value. Start this calibration each time a different

```
HV [] > probe
0
1
2
3
4
5
6
7
8
9
10
HV [] > select 6
HV module with address 6 selected
HV [6] >
```

Figure 3: Probe command output

```
HV [6] > info
FW ver                : 2C
PMT s/n               :
HV s/n                :
FEB s/n               :
Vref                  : 2428 mV
Calibration slope     : 0.0
Calibration offset    : 0.0
HV [6] >
```

Figure 4: Info command output

```
HV [6] > mon
    status   Vset        V       I       T    rate UP/DN     limit V/I/T/TRIP   trigger thr      alarm
               [V]      [V]    [uA]    [°C]   [V/s]/[V/s]   [V]/[uA]/[°C]/[s]          [mV]
◉    DOWN    1000   171.453   2.907      24         25/25           20/5/50/5             1      none
HV [6] > mon 5
    status   Vset        V       I       T    rate UP/DN     limit V/I/T/TRIP   trigger thr      alarm
               [V]      [V]    [uA]    [°C]   [V/s]/[V/s]   [V]/[uA]/[°C]/[s]          [mV]
◉    DOWN    1000   170.847   2.902      24         25/25           20/5/50/5             1      none
◉    DOWN    1000   170.847   2.900      24         25/25           20/5/50/5             1      none
◉    DOWN    1000   170.755   2.900      24         25/25           20/5/50/5             1      none
◉    DOWN    1000   170.697   2.902      24         25/25           20/5/50/5             1      none
◉    DOWN    1000   170.480   2.899      24         25/25           20/5/50/5             1      none
HV [6] >
```

Figure 5: mon command output

HV board is connected to a FEB. This is a long task, so just wait.

- Limit is a command to set the different limit of the board, the default parameters are fine:
  - Limit temperature is the maximum temperature before an alarm is generated
  - Limit current is the maximum current desidered
  - Limit voltage is the margin on the HV value to not generate Under/Over voltage alarm with respect to the set voltage value
  - Limit triptime is the maximum allowed time before turning down the board in case of an alarm from the previous limits
- On/off commands just enable or disable the HV output
- Rate controls the rumpup and rumpdown values of the HV. Using the default value is fine.
- Reset command resets the board in case of a trip occurency after an alarm.
- Threshold controls the threshold of the acquisition and it is in mV. The threshold is related to the amplified signal, so no direct relation is expected with the signal amplitude. The gain is 30, so a good starting point is 0.25PE that corresponds almost to 37 mV of threshold.
- Voltage commands is the value of the desidered HV output value.

A different utility must be used to store the parameters in a CSV file, called hvmon. There is the usual help to facilitate the usage.

To start the monitoring send the following command:

$\$\ hvmon.py\ –port\ PORT\ –m\ x,y\ –f\ file-name.csv$

where PORT is your USB port and x,y are the ModBus address of the connected boards.

3.2 Run control

The Run Control tool let the user to control the FPGA registers. The protocol is serial and you could use a terminal like Putty or Minicom to control the registers, but a cli utility is provided to facilitate the process. The cli utility let also to store the parameters in a CSV file. The USB cable for this utility is the one on the left of the picture 1, the one in the PMOD UART/USB.

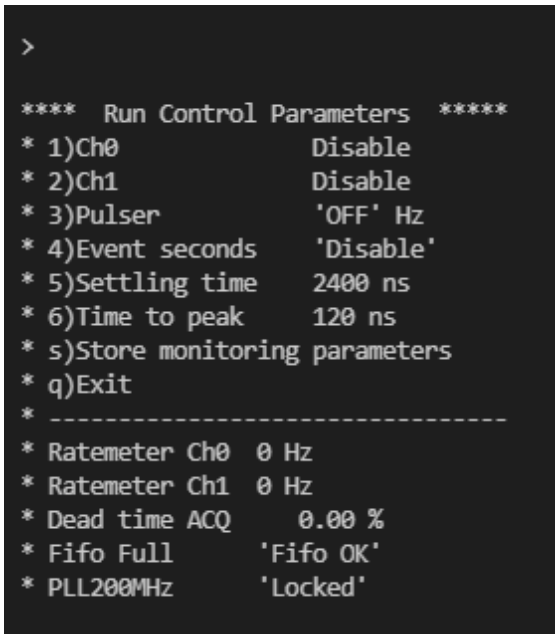

Figure 6: Run Control utility menu

3.2.1 Prerequisites

The tool is a python script that requires some libraries to work.
Type the following commands to install all the libraries:

$ sudo apt-get install time os time sys pytimedinput serial numpy math

3.2.2 Overview

To launch the utility just run *$python RunCrtlV1.py*

A list of the available serial port will be printed, insert the desidered one. Pressing enter the menu will be printed each time. To run one of the command just press the number of the command and enter, then insert the desidered value based on the possibility of the choosen command.

- 1) Ch0 command enables or disables the channel acquisition on JC.
- 2) Ch1 command enables or disables the channel acquisition on JD.
- 3) Pulser generates an ADC pedestal event in the data with a programmable frequency

- 4) Event seconds enables the transmission in the data of the counter of the elapsed seconds. En event will be transmitted every second, the channel number is 0x31
- 5) Settling time is the time after each event in which the ACQ does not accept any more data.
- 6) Time to Peak control the time interval between the trigger generation and the starting of the ADC conversion

On the bottom of the menu some monitoring information are provided.

- Ratemeter from Channel 0 and 1
- Acquisition dead time in percentage
- The statut of the FPGA FIFO, if ok or full
- The status of the internal FPGA PLL

With the q command is possible to wuit the utility, while the s command gives you the possibility to store parameters into a CSV file. After pressing s the user will be asked to insert the frequency at whick monitor the parameters and a file name, that will be completed by the date and time. The file will be saved in a log folder locally (remember to create the folder before).

3.3 Data Acquisition

The last utility is the acquisition tool. This python script manages the acquisition through the Serial port of the Nexys A7 (the one near the power switch of the board) at a baudrate of 921600.
The script automatically scans the serial looking for new events, checks if the event is valid and generates a CSV file with the valid events of the acquisition. When reading the serial port, some values may not be recognized by the software, in this case a "non-ASCII character" message is printed and the relative event is discarded.
Another python script is provided to plot the CSV file in real time to check if the acquisition is going as planned and configured. Only the ADC values are plotted.

3.3.1 TDC Calibration

It is necessary, when acquiring for the first time, to calibrate the TDC. Infact the time fine, as well as the t.o.t. fine, assume a range of values that goes from 0 to 19. These steps correspond to a total time interval of 5 ns and the width of each step depends on different weights. In order to calibrate the values a software has been developed that acquires data for 5 minutes and calcuates the correct steps.

```
> s
Storage rate (sec) >1
File name >file



****  Run Control Parameters  *****
* 1)Ch0              Disable
* 2)Ch1              Disable
* 3)Pulser           'OFF' Hz
* 4)Event seconds    'Disable'
* 5)Settling time    2400 ns
* 6)Time to peak     120 ns
* s)Store monitoring parameters
* q)Exit
* --------------------------------
* Ratemeter Ch0  0 Hz
* Ratemeter Ch1  0 Hz
* Dead time ACQ     0.00 %
* Fifo Full       'Fifo OK'
* PLL200MHz       'Locked'

To stop press x
x
```

Figure 7: Run Control utility in storage mode, to stop press x

```
prova_2021_11_30-16_54_22.csv
Time,Freq_CH0,Freq_CH1,DeadTime,FifoFull
2021_11_30-16_54_27 ,0,0,0.00,Fifo OK
2021_11_30-16_54_28 ,0,0,0.00,Fifo OK
2021_11_30-16_54_29 ,0,0,0.00,Fifo OK
2021_11_30-16_54_30 ,0,0,0.00,Fifo OK
2021_11_30-16_54_32 ,0,0,0.00,Fifo OK
2021_11_30-16_54_33 ,0,0,0.00,Fifo OK
2021_11_30-16_54_34 ,0,0,0.00,Fifo OK
2021_11_30-16_54_35 ,0,0,0.00,Fifo OK
2021_11_30-16_54_36 ,0,0,0.00,Fifo OK
```

Figure 8: Run Control utility CSV output example

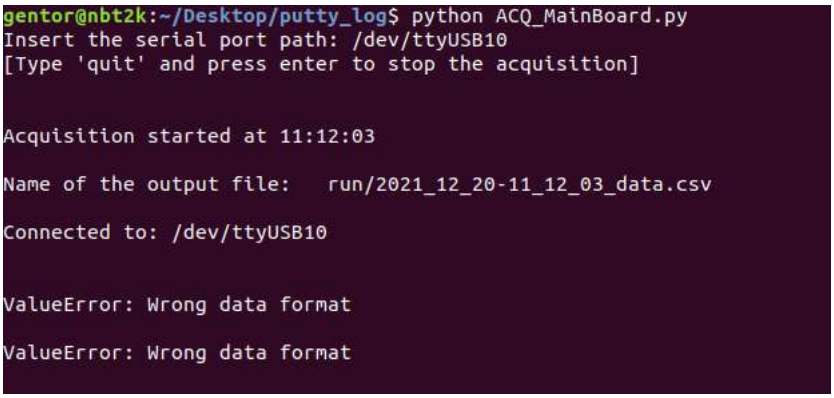

Figure 9: Example of how the acquisition tool works

The first step is to create a subdirectory named "TDC_Calibration" in the same directory that contains the code, then run:

$ python TDC_Calibration.py

For each channel, the weighted values are stored in a csv file in the \TDC_Calibration folder. These files are then automatically read by the acquisition software.

3.3.2 Acqusition procedure

To run the acqusition it is mandatory that in the same directory that contains the code there are two subdirectories: \ *TDC_calibration*\ and \ *run* \.
The acqusition starts with the following command:

$ python ACQ_MainBoard.py

Then the user has to insert the serial port path and press enter.

The software will print the starting time of the acquisition as well as the name of the output file and a check that the connection to the serial port went well. Then the acquisition starts and each time the software will find non-readable raw data it will print an on-line string "ValueError: Wrong data format".
To stop the acqusition it is possible to type *quit* and press enter or to type ctrl+c. An example of how the software works can be seen in Fig.9

It is possibile to check the acquisition by plotting in real time the histogram of the energy. In order to do this, one has to run the following command:

$ python EnHist_realtime.py

The output will be two histograsms, one for each channel, that will show the energy values while acquiring them. The histograms are updated every five

seconds.

3.3.3 Output file

Each time an acquisition is made, a CSV file is created. All the output files can be found in the */run* subdirectory. Files' names contain information about the date and starting time of the acqusition. Event's information are stored in 9 columns:

- **Channel:** it indicates the PMT that has seen an event. With this demo it can only be '0' or '1', since only two channels can be connected.
- **Time coarse:** it is the time measured by the 5ns clock .The unity is ns.
- **T.o.t coarse:** it is the time over threshold (i.e. the width) of the signal, measured by the 5 ns clock. The unity is ns.
- **Time fine:** it is the time between two clocks measured by the TDC. The unity is ns.
- **T.o.t. fine:** it is the width of the signal measured by the TDC. The unity is ns.
- **Event time:** the actual time of the event. The unity is ns. It is calculated as:

$$\text{Event time} = \text{Time coarse} - \text{time fine}$$

- **Event t.o.t.:** the actual width of the signal. The unity is ns. It is calculated as:

$$\text{Event t.o.t.} = \text{T.o.t. coarse} - \text{T.o.t. fine} + \text{Time fine}$$

- **Energy:** the charge collected by the PMT. The unity is ADC channels.
- **Acquisition time:** the time at which the event was acquired.

An example of an output file can be found in Fig.10.

3.4 Acquisition using oscilloscopes

In order to digitalize the PMT signal with an oscilloscope the jumper between the HV board and the FE board must be disconnected and a cable must be connected from the HV board to the oscilloscope. The ground of the signal is on the internal pin, so the signal is on the external one. The HV utility is however necessary to control the HV parameters.

Channel	Time Coarse (u=ns)	T.o.t. coarse (u=ns)	Time fine (u=ns)	T.o.t. fine (u=ns)	Event time (ns)	Event t.o.t. (ns)	Energy (u= ADC channels)	Acquisition time
0	331132435	25	4.71061921194408	2.278748077984	331132430.289381	27.43187113396	361	11:15:09
0	331568695	30	0.832276958108854	3.8866685446671	331568694.167723	26.94560841344	348	11:15:09
0	332153360	30	0.832276958108854	4.9985254762492	332153359.167723	25.83375148186	344	11:15:09
0	333078570	25	4.35864085589868	2.9475436341655	333078565.641359	26.41109722173	339	11:15:09
0	333086895	30	1.13947674213881	3.2571642780849	333086893.860523	27.88231246405	367	11:15:09
0	333309225	25	2.62624417526439	1.5145412984025	333309222.373756	26.11170287686	340	11:15:09
0	333927665	25	4.16379831449464	4.7225767926101	333927660.836202	24.44122152188	333	11:15:09
0	334165770	30	2.94068453114546	3.6107492047842	334165767.059315	29.32993532636	402	11:15:09
0	334652350	20	2.62624417526439	2.0658077749217	334652347.373756	20.56043640034	313	11:15:09
0	335143820	25	1.13947674213881	3.6107492047842	335143818.860523	22.52872753735	324	11:15:09
0	335511200	20	1.81479394814371	4.1762033874432	335511198.185206	17.6385905607	301	11:15:09
0	335807595	25	2.05687260114793	2.0658077749217	335807592.943127	24.99106482623	338	11:15:09
0	336794090	20	3.6076461025623	0.8502206650469	336794086.392354	22.75742543752	322	11:15:09
0	337292600	30	0.832276958108854	1.5145412984025	337292599.167723	29.31773565971	393	11:15:09
0	337882980	25	5	3.8866685446671	337882975	26.11333145533	344	11:15:09
0	338147700	15	4.16379831449464	4.1762033874432	338147695.836202	14.98759492705	297	11:15:09
0	338883945	25	2.05687260114793	0.2087588177987	338883942.943127	26.84811378335	356	11:15:09
0	339716130	25	2.94068453114546	4.7225767926101	339716127.059315	23.21810773854	326	11:15:09
0	340428035	25	3.6076461025623	0.8502206650469	340428031.392354	27.75742543752	371	11:15:09
0	340442060	25	4.71061921194408	3.6107492047842	340442055.289381	26.09987000716	345	11:15:09
0	341310235	20	2.05687260114793	2.9475436341655	341310232.943127	19.10932896698	306	11:15:09
0	341588625	15	4.16379831449464	2.6327804969659	341588620.836202	16.53101781753	299	11:15:09
0	342091000	30	0.186795016256441	4.9985254762492	342090999.813205	25.18826954001	339	11:15:09
0	342108135	25	1.50644242168151	0.8502206650469	342108133.493558	25.65622175663	343	11:15:09
0	342185930	25	0.548060671150393	4.3676934046974	342185929.451939	21.18036726645	341	11:15:09
0	342861435	35	0.186795016256441	4.1762033874432	342861434.813205	31.01059162881	396	11:15:09
0	343036580	25	3.6076461025623	3.8866685446671	343036576.392354	24.7209775579	336	11:15:09
0	343624340	25	0.832276958108854	2.6327804969659	343624339.167723	23.19949646114	322	11:15:09
0	343793300	25	3.6076461025623	0.5596294470463	343793296.392354	28.04801665552	367	11:15:09

Figure 10: A view to the output file created when acquiring data

3.5 Example of acquisition

After everything is connected and powered, it is possible to staart the cli utilities. Starting with the HV and the Run control ones. Open the run control utility and enable the necessary channel with 1 or 2 command, than open the HV utility and enable the probe to understand which board is connected. After selected the desidered board the correct procedure to turn on a board is the following. First reset, then choose the voltage value and finally enable the hv with the on commnand. Once everything is on and enabled you can start the acquisition as described in the acquisition tool section.