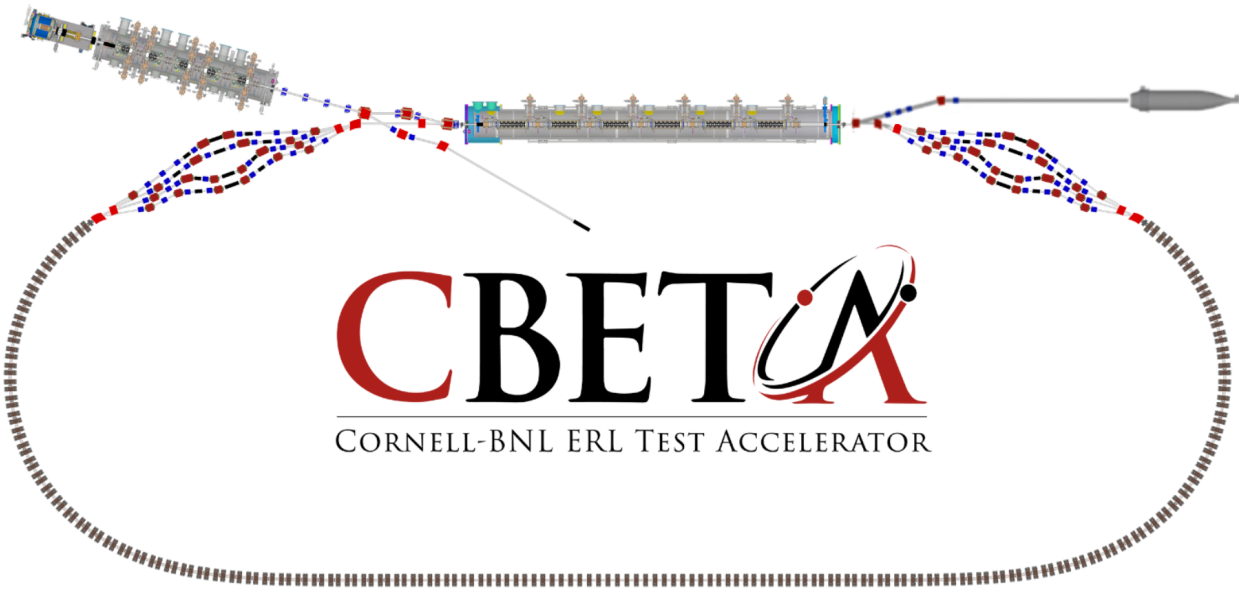
BROOKHAVEN
NATIONAL LABORATORY

Fast Beam Loss Monitor System for CBETA Beam Loss Detection

J.Renta, R. Michnoff, R. Hulsart

Brookhaven National Laboratory, Upton, NY 11973-5000, USA

March 25th 2020



Overview

There are seven Fast Beam Loss Monitor (FBLM) Chassis in which contain two Red Pitayas each. A Red Pitaya is a hardware module used for laboratory measurement and control instrumentation. In CBETA, it is being used to read signals from Photomultiplier Tubes (PMT). These PMTs detect photons from a special fiber that scintillates when exposed to radiation. The fiber is concealed within a black tube that is dressed alongside the beam pipe. Each FFA girder is equipped with one scintillating fiber on the inside and one on the outside of the beam pipe. These two fibers are connected to one PMT. The entire FBLM subsystem presently provides the detection of beam loss from twenty seven PMTs. Additional PMTs will be added in the future for the splitter beam pipe sections. The FBLM system latches a low output signal as soon as it detects a beam loss that has exceeded a user specified limit/threshold and duration window. This output signal can then be used to halt the laser beam as a method for protection overall.

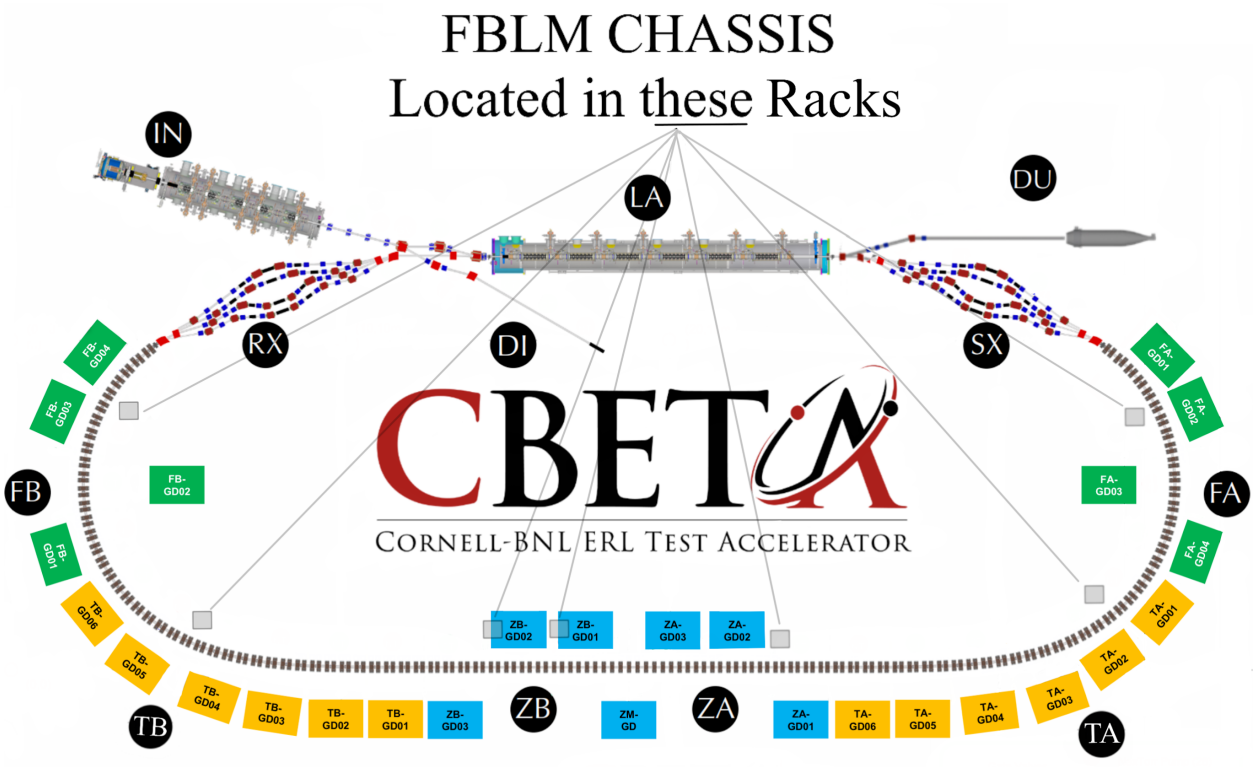


Figure 1 - CBETA Layout of Racks Containing VME and FBLM Chassis

Purpose

The FBLM Chassis is designed to provide a fast shutdown system for the protection of accelerator system components.

Interconnectivity

As shown in Figure 2, A girder is seen with eight magnet assemblies that have a scintillating fiber running across its side. One end of this fiber has a plug while the other is connected to an interconnect box shown in Figure 3. This small box provides the interface to the PMT, from Figure 4, via a fiber patch cable. The PMT in Figure 4 shows three electrical connections which are +15VDC, High Voltage Set Point and Signal Out. These electrical cables are connected to the FBLM Chassis.

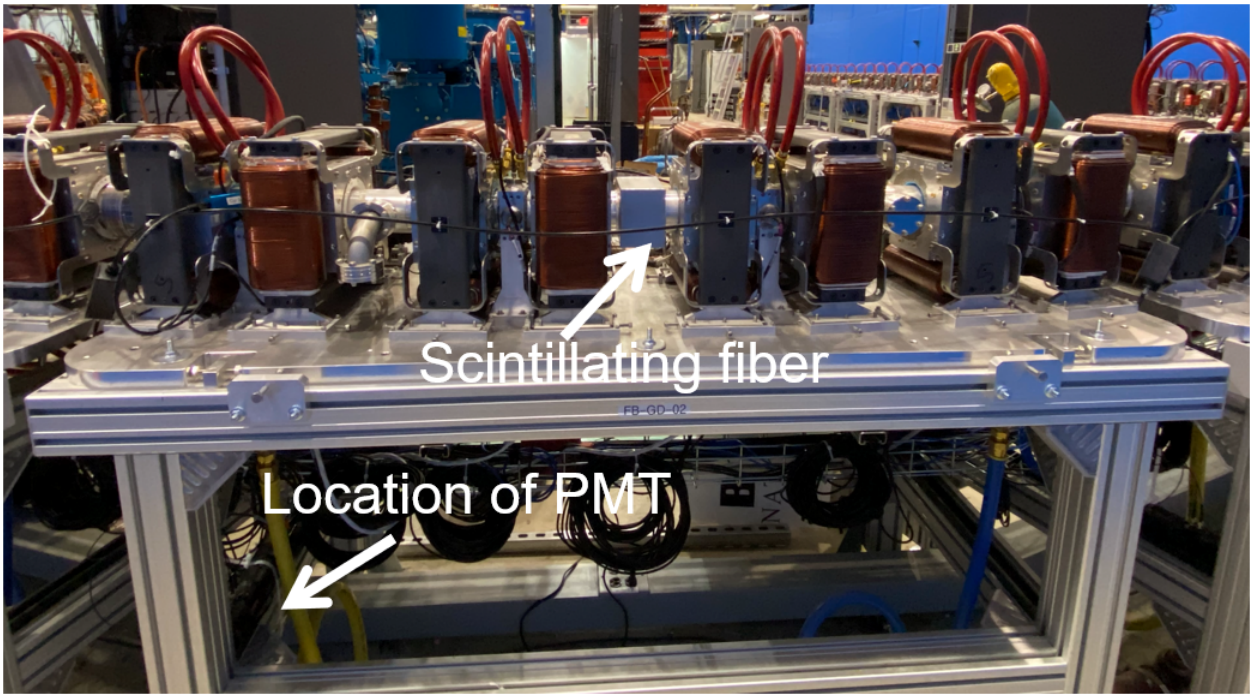


Figure 2 - Girder Image showing Fiber and PMT

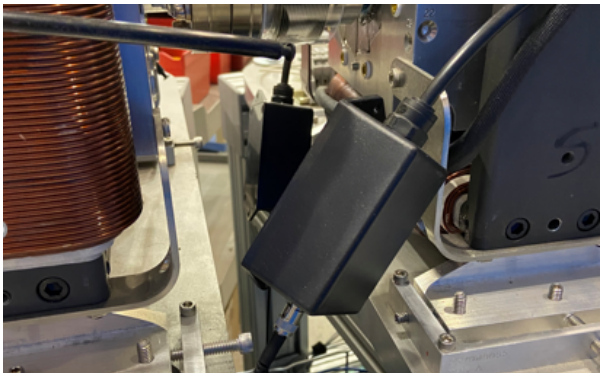


Figure 3 - PMT Fiber Interconnect Box

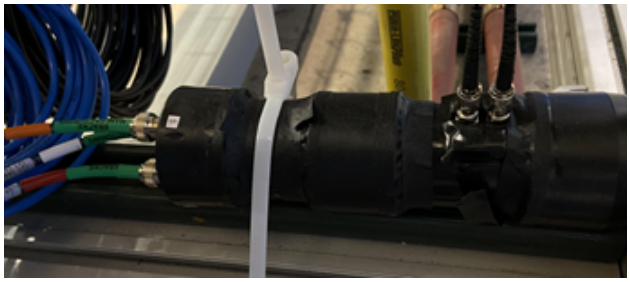


Figure 4 - PMT tie-wrapped to Girder

FBLM Chassis

The FBLM Chassis from Figure 5 has a vertical line in the center of the front panel to create two sections. Each section takes 5V and accommodates connectivity to two PMTs each for a total of four PMTs per chassis. The OUT1 connection is the output signal that gets latched to a low level when the beam loss limit has been tripped. The Enable, Stop and OUT2 connections are used for administrative/debugging purposes but can be available for extended future functionalities. The LAN port gets connected to the Local Area Network and the Clk/Trig port connects to the BPM Clock & Trigger.

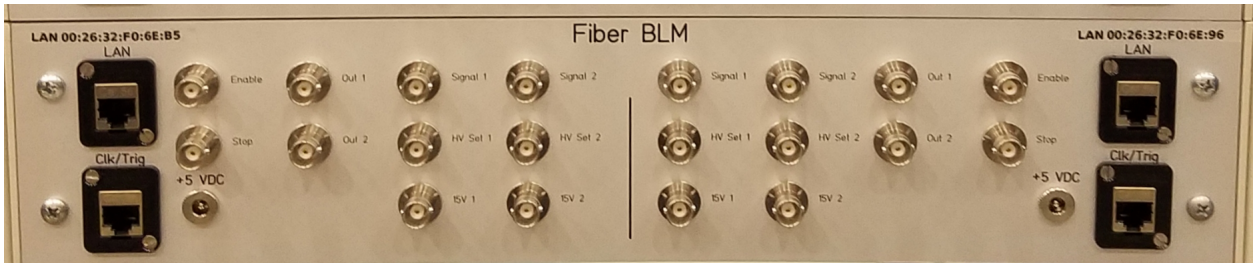


Figure 5 - FBLM Chassis

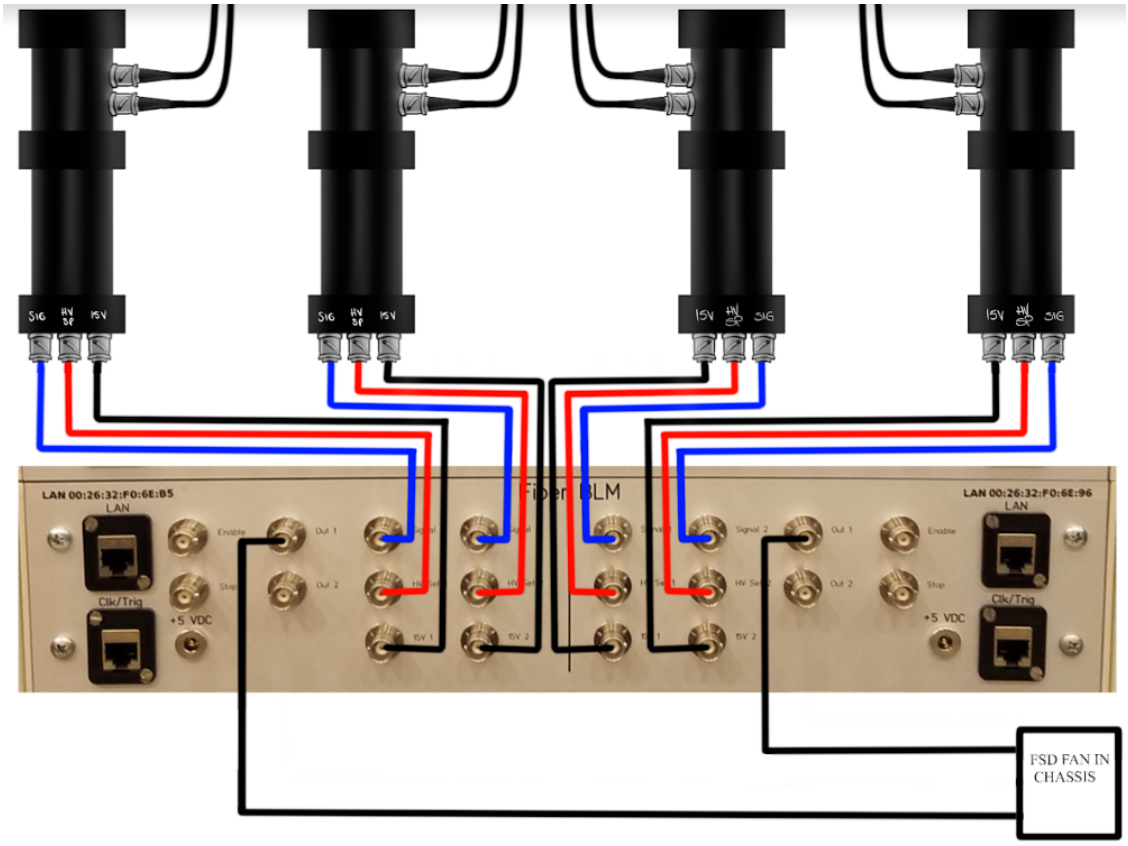


Figure 6 - FBLM Chassis wired to four PMTs and Fast Shut-Down(FSD) FAN IN Chassis

The FBLM Chassis is loaded with two sets of Expansion Boards connected to Red Pitayas. The Red Pitaya is equipped with twenty six pin expansion slots which can be seen in Figure 7. These expansion slots provide further communication with the outside world. The Expansion Board from Figure 8 is designed with a 5V DAC as opposed to the 1V DAC that the Red Pitaya offers. This 5V DAC is essential for the PMT's High Voltage Set Point as the Tubes used in this system come with their own High Voltage Power Supply Socket assembly which will be discussed later. Refer to the Hardware Section of this document for more information about the Red Pitaya and Expansion Board. All of the connections from the front panel of the FBLM Chassis are connected to these hardware devices, as shown in Figure 9.

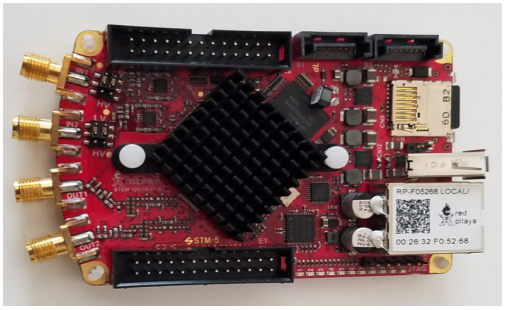


Figure 7 - Red Pitaya

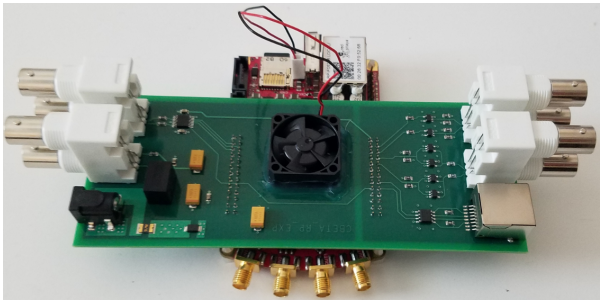


Figure 8 - Expansion Bd on Red Pitaya

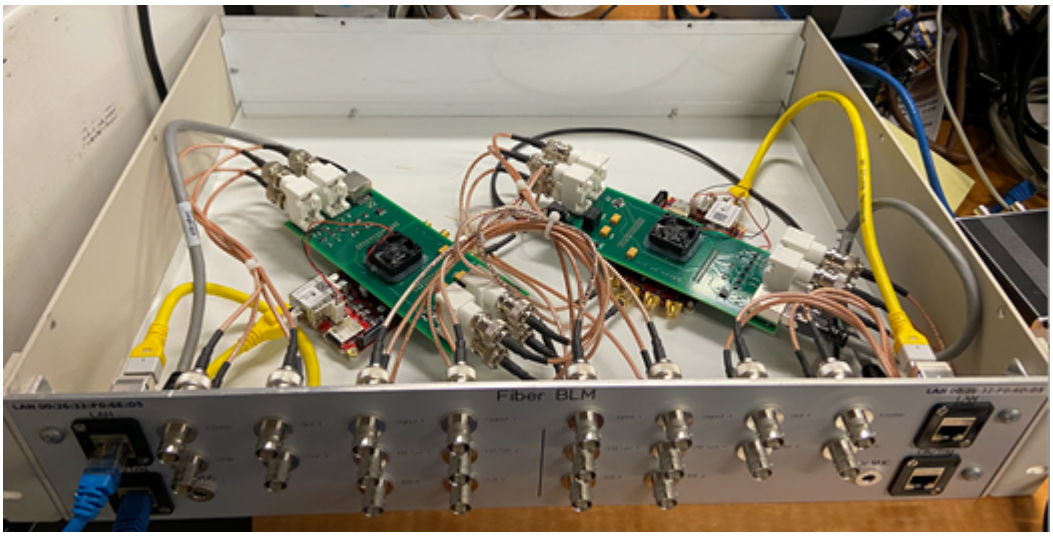


Figure 9 - Inside the FBLM Chassis

Features

The following features are provided with the FBLM System

- FBLM Chassis Supports 4 Photomultiplier Tubes
- PMT’s equipped with their own High Voltage Power Supply
- Customizable Beam Loss Limit
- Customizable Beam Loss Limit Window
- 4096 Samples at 8ns/S
- Post-Mortem Data Hold supported
- Simple Graphical User Interface
- Trigger Readback on Front Panel

Subsystem Design

Hardware

RedPitaya STEMlab 125-14

- FPGA Xilinx Zynq 7010 SoC
- RAM 512MB
- 14 bit input / output channel
- 2 RF inputs (Sampling rate = 125MS/s)
- Input Impedance (1MOhm/10pF)
- Input Voltage range +/- 1V or +/- 20V
- Output Load Impedance (50 Ohm)
- Output Voltage range +/- 1V (Expansion bd provides 5V out)

Expansion Board (Designed by John Dobbins from Cornell University)

- 16 Bit Dual DAC
- RJ45 Port for Clk/Trg
- 4x Dual BNC throughhole connectors
- FAN to keep Red Pitaya's FPGA cool

Photomultiplier Tube

- Hamamatsu P/N: R11558
- Spectral Response: 300 to 650nm
- 28 mm diameter, Side-on Type

HV Power Supply Socket for PMT

- Hamamatsu P/N: C12597-01

Scintillating Fiber

- Saint Gobain P/N: BCF60 MC 1.00 DIA, R/C Black

Fiber Protective Tubing

- McMaster-Carr P/N: 9349T1
- Material: Crack-resistant Polypropylene Plastic Tubing, Hard Rating
- Size: ⅛" ID, ¼" OD

Software

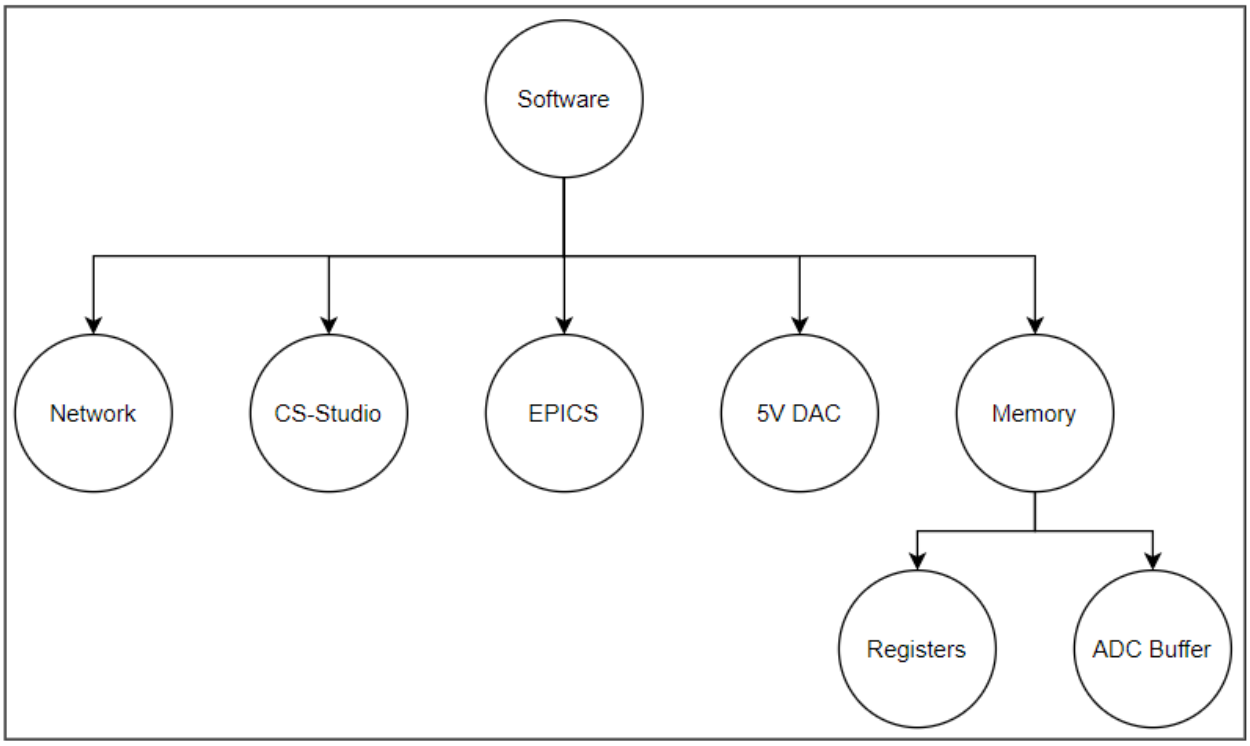


Figure 10 - Software Support Overview

Software was developed to support the following five sections: Network, EPICS, 5V DAC, Memory and CS-Studio as illustrated in Figure 10. The FBLM code for the EPICS IOC, 5V DAC and Memory are stored in the Red Pitaya's 16GB Class X micro SD Card within this directory /EPICS/base-3.14.12.5/fblm/fblmApp/src/ but can be accessible from the gitlab.pbn.bnl account specified in the address link below: [1]CBETA_FBLM_SOFTWARE. The following sections provide a breakdown for the software support that was needed for the FBLM system.

Network

Connecting the Red Pitaya to the Network was initiated by submitting this IT form: [2]IP Form Being able to SSH was essential inorder to mount the operations file system. The following command was used similarly to what was done with the V301 BPM boards:

```
mount  -t nfs -o nolock
fec.pbn.bnl.gov:/home/here/opfecs
/operations
```

However, this reported the following error:

```
mount: wrong fs type, bad option, bad superblock on fec.pbn.bnl.gov:/home/here/opfecs,
       missing codepage or helper program, or other error
       (for several filesystems (e.g. nfs, cifs) you might
       need a /sbin/mount.<type> helper program)

       In some cases useful info is found in syslog - try
       dmesg | tail or so.
```

There were a few reasons, one was that the new network never made it into the control's system /etc/hosts.allow file so the Control's System Group. Another issue was that the base installation for "NFS Utilities" needed to be installed. As Root on an internet connected network the following command is used to install the NFS.

```
apt install -y nfs-common
```

Seek your local network administrator for additional Network help.

Table 1 below represents the IP address assignments that were needed for the CBETA FBLM system. An email was sent to service-classe@cornell.edu for this support.

IP Address	Host Name	Alias Host Name	MAC Address
172.18.43.39	erpred19.classe.cornell.edu	erpred-f0705e.classe.cornell.edu	00:26:32:F0:70:5E
172.18.43.40	erpred20.classe.cornell.edu	erpred-f06dd1.classe.cornell.edu	00:26:32:F0:6D:D1
172.18.43.41	erpred21.classe.cornell.edu	erpred-f07159.classe.cornell.edu	00:26:32:F0:71:59
172.18.43.42	erpred22.classe.cornell.edu	erpred-f070fe.classe.cornell.edu	00:26:32:F0:70:FE
172.18.43.43	erpred23.classe.cornell.edu	erpred-f070cc.classe.cornell.edu	00:26:32:F0:70:CC
172.18.43.44	erpred24.classe.cornell.edu	erpred-f070c1.classe.cornell.edu	00:26:32:F0:70:C1
172.18.43.45	erpred25.classe.cornell.edu	erpred-f076df.classe.cornell.edu	00:26:32:F0:76:DF
172.18.43.46	erpred26.classe.cornell.edu	erpred-f0751a.classe.cornell.edu	00:26:32:F0:75:1A
172.18.43.47	erpred27.classe.cornell.edu	erpred-f07158.classe.cornell.edu	00:26:32:F0:71:58
172.18.43.48	erpred28.classe.cornell.edu	erpred-f07138.classe.cornell.edu	00:26:32:F0:71:38
172.18.43.49	erpred29.classe.cornell.edu	erpred-f070b6.classe.cornell.edu	00:26:32:F0:70:B6

Table 1 - CBETA FBLM IP Address Assignment

5V DAC

Firmware support was provided to establish the I/O pins needed for this to work. For more information on Firmware Support for the DAC, refer to the Firmware section of this document. This DAC is the AD5689 chip which is a dual 16-Bit DAC with SPI interface. It provides the High Voltage Set Point from the Red Pitaya's expansion board to both connected PMTs. A functional block diagram is shown in Figure 11 followed with the schematic on Figure 12.

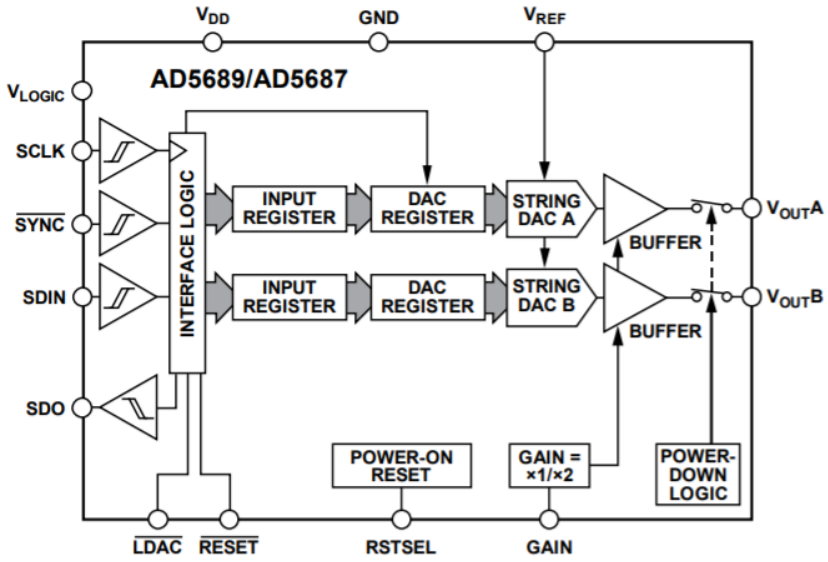


Figure 11 - AD5689 Functional Block Diagram

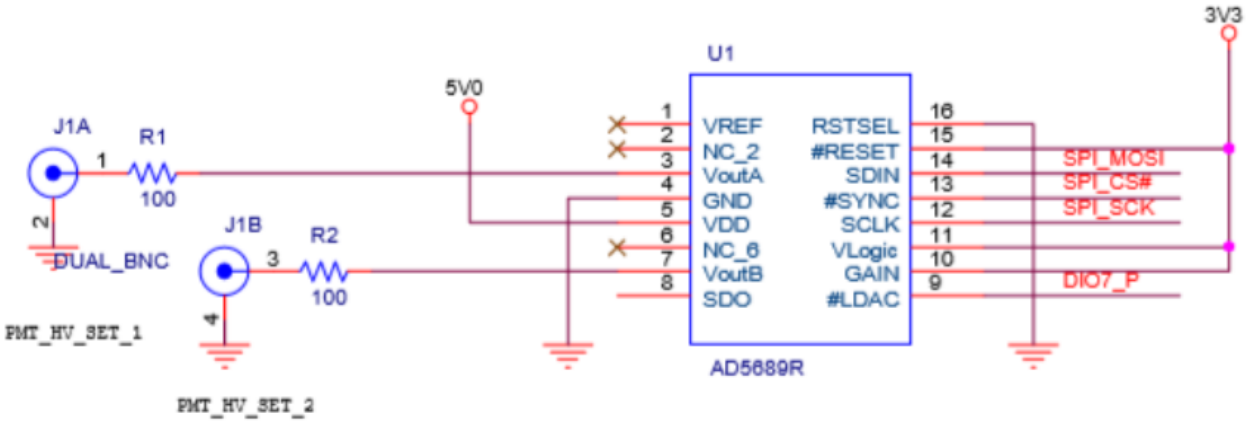


Figure 12 - DAC Schematic

DAC Communication

The Red Pitaya communicates with the DAC via spi and uses the [3]rp_spi.c code to update a 24 bit input shift register from MSB. These 24 bits are divided into three categories: Command Bits, Address Bits and Data Bits as shown in Figure 13. The command and address bits are held at a constant binary value of 0b00011001 and can be referenced from Figures 14 & 15. Lastly, the remaining 16 bits suggests a max value of 65,535 which is also the max output voltage the DAC can output: 5 volts. The following equation is the conversion formula for Vout.

$$V_{OUT} = V_{REF} * Gain\left[\frac{D}{2^N}\right]$$

$$(V_{REF} = 2.5, Gain = 2, D = Decimal\ Eq.of\ Binary\ Code, N = DAC\ Resolution)$$

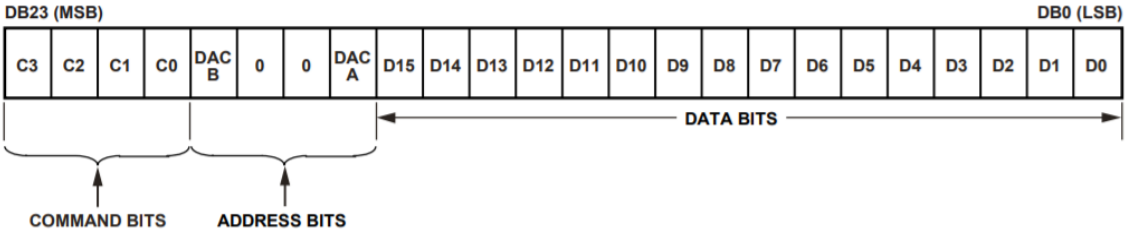


FIGURE 13 - DAC, Input Shift Register Content

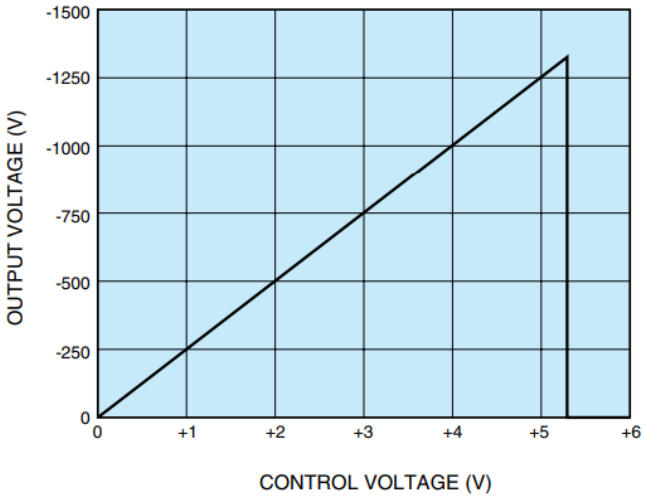
Command				
C3	**C2**	**C1**	**C0**	**Description**
0	0	0	0	No operation
0	0	0	1	Write to Input Register n (dependent on $\overline{LDAC}$)
0	0	1	0	Update DAC Register n with contents of Input Register n
0	0	1	1	Write to and update DAC Channel n
0	1	0	0	Power down/power up DAC
0	1	0	1	Hardware $\overline{LDAC}$ mask register
0	1	1	0	Software reset (power-on reset)
0	1	1	1	Reserved
1	0	0	0	Set up DCEN register (daisy-chain enable)
1	0	0	1	Set up readback register (readback enable)
1	0	1	0	Reserved
…	…	…	…	Reserved
1	1	1	1	No operation, daisy-chain mode

FIGURE 14 - DAC, Command Description

Address (n)				
DAC B	**0**	**0**	**DAC A**	**Selected DAC Channel**
0	0	0	1	DAC A
1	0	0	0	DAC B
1	0	0	1	DAC A and DAC B

FIGURE 15 - DAC, Address Commands

DAC Output Results



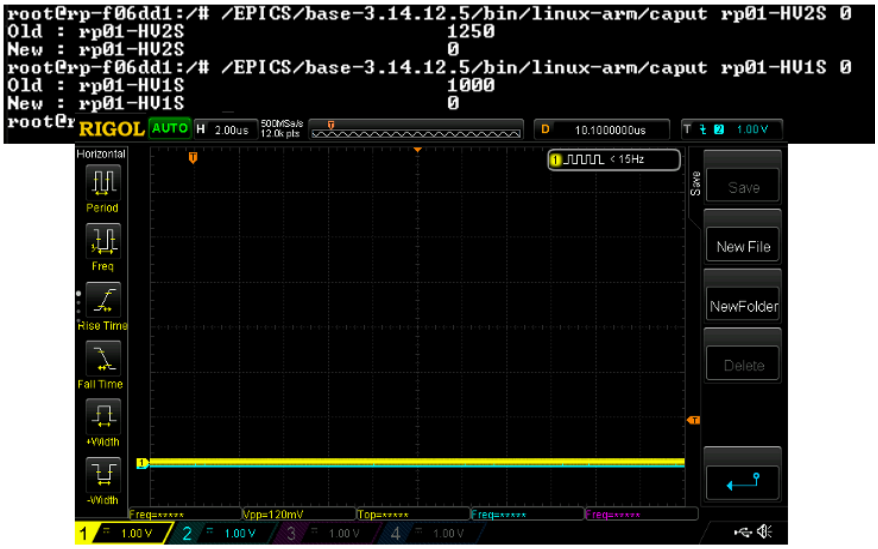


Illustration 1 - DACs set to

Illustration 1 – DACs @ 0V

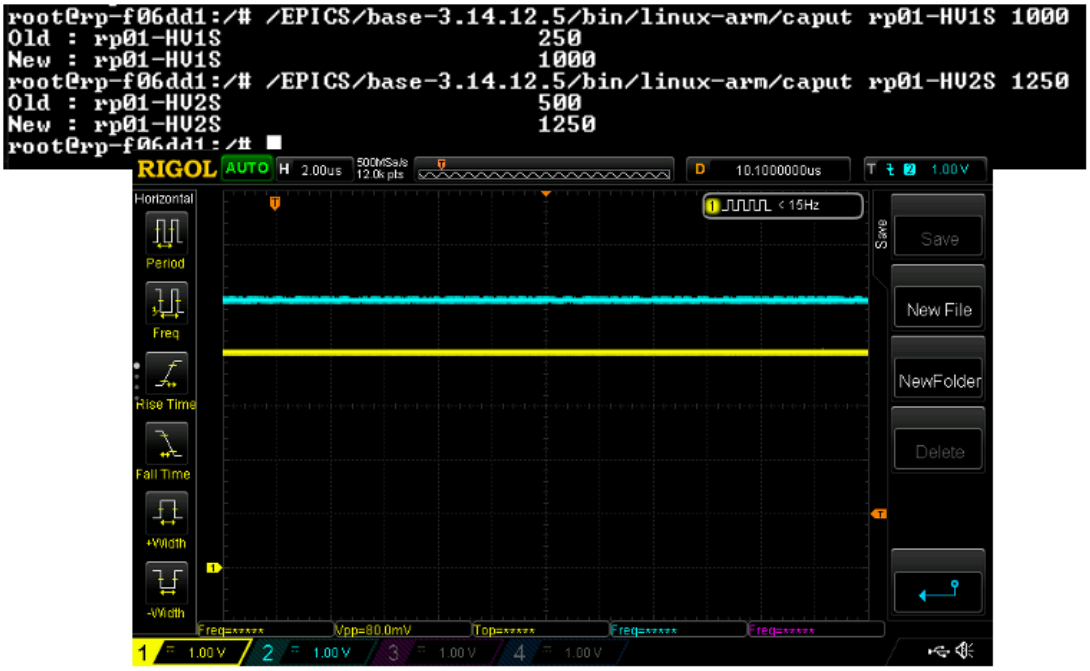


Illustration 2 – DAC 1 set to 1000 / DAC 2 set to 1250

(Scope Setting: 1V/DIV; DAC 1 = 4V, DAC 2 = 5V)

Memory

The code that accesses memory for both ADC Buffers and Registers can be found in [4]ReadData.c. This file is used with EPICS to get and set updated values by including an epics function that contains the name of the method within.
ie, epicsRegisterFunction(*nameOfMethod*);

EPICS

Setting up the IOC

The Experimental Physics and Industrial Control System (EPICS) contains an Input / Output Controller (IOC) to handle data structures, records and devices. The following are steps needed to build and run the IOC via a terminal.

To build the IOC from the directory: /EPICS/base-3.14.12.5/fblm
Enter “make” at the prompt.

Type the following to start the IOC: cd /EPICS/base-3.14.12.5/fblm/iocBoot/iocfblm
And enter the following command: ../../bin/linux-arm/fblm st.cmd

To see a list of IOC records in the database type “dbl” then “exit” to quit, see Figure 16.

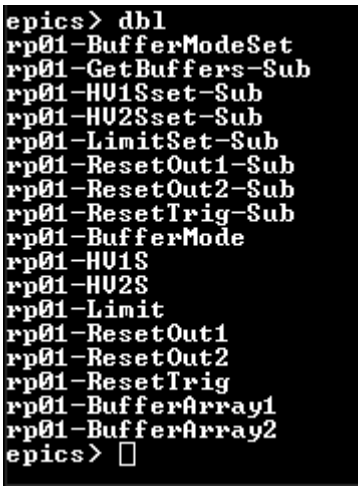


FIGURE 16 - EPICS IOC Database Records

Adding a new Record

The following files were updated when adding a new record to the Database:

Database records: /EPICS/base-3.14.12.5/fblm/fblmApp/Db/Data.db

Record code: /EPICS/base-3.14.12.5/fblm/fblmApp/src/ReadData.c

Subroutines: /EPICS/base-3.14.12.5/fblm/dbd/fblm_subroutines.dbd

Save/Restore: /EPICS/base-3.14.12.5/fblm/iocBoot/iocfblm/autosave

For additional details on EPICS Record, refer to the [5]EPICS Record Manual.

CAGET Function

While EPICS is running on a different terminal, type the following to get a record value:

```
/EPICS/base-3.14.12.5/bin/linux-arm/caget rp01-HV1S
```

The result will be the following:

```
rp01-HV1S              0
```

CAPUT Function

The caput function is used to update record values. Type the following to do so:

```
/EPICS/base-3.14.12.5/bin/linux-arm/caput rp01-HV1S 10
```

The result will be the following:

```
Old: rp01-HV1S              0
New: rp01-HV1S             10
```

CS-Studio

An app with Control System Studio (CS-Studio) was created to view the FBLM sampled data from all 4 PMTs connected to the FBLM Chassis as shown in Figure 17. Open No Machine then on a terminal type the following: cd /home/cfsd/jrenta/epics Then type: ./css

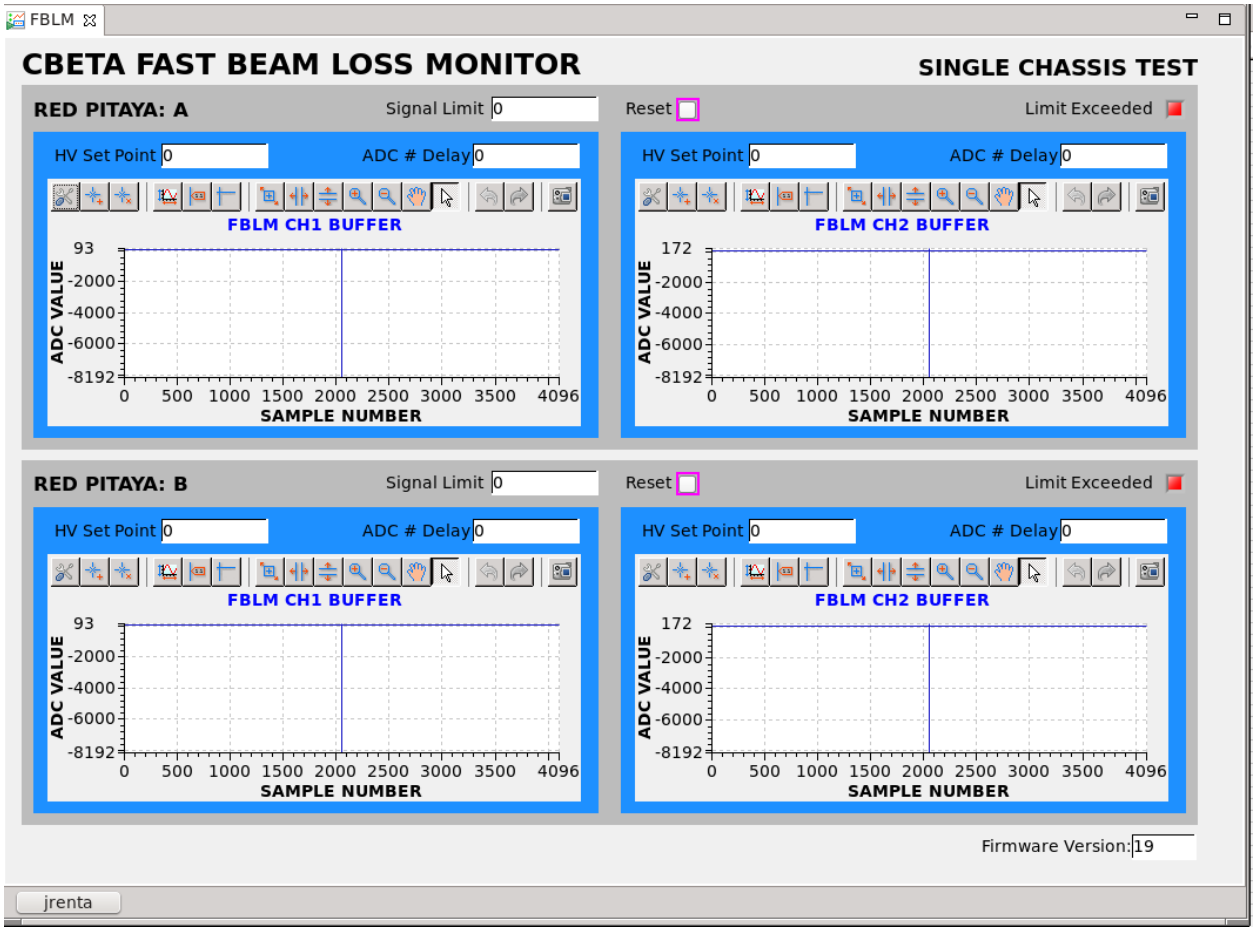


FIGURE 17 - CS-Studio, FBLM.opi

The “HV Set Point” sets the voltage to the PMT’s with a max value of 1250 being equal to 5V. “Signal Limit” is used to provide an input threshold to the two PMTs connected to the Red Pitaya. “ADC # Delay” sets the number of allowed ADC samples that is equal or greater than the limit specified, providing a tunable window. When the “Limit Exceeded” LED turns RED, it means that the system has sampled a limit for the duration specified in ADC # Delay and has halted the buffer so it can be used for post mortem analysis.

Current Development

The CS-Studio app is currently under development and comes with a python file named [6]HVSP2V.py for converting HV counts to voltage as shown in Figure 18.

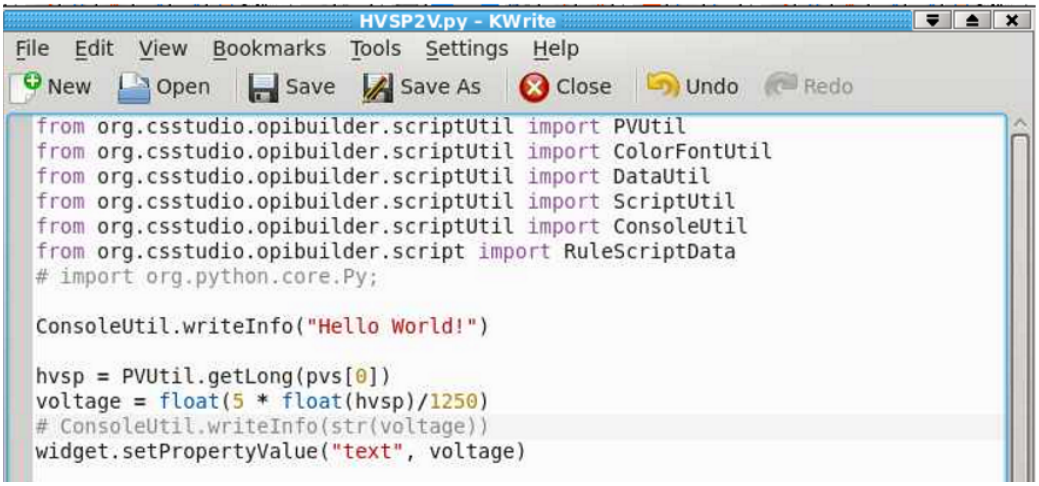


FIGURE 18 - HVSP2V.py

In Figure 19 below, there are two graphs. The graph on the left is an example of data that has been sampled from a function generator with a triangle wave at 62.5KHz. The graph on the right is data sampled from a PMT that is ON but has no input signal. The following equation proves 8ns per sample for this design.

$$2000\ samples\ *\ \frac{1}{125MHz\ (Sampling\ Freq)} = 2000\ *\ 8ns\ (sampling\ time)\ = 16us$$

$$\frac{1}{16us} = 62.5KHz\ (Input\ Frequency)$$

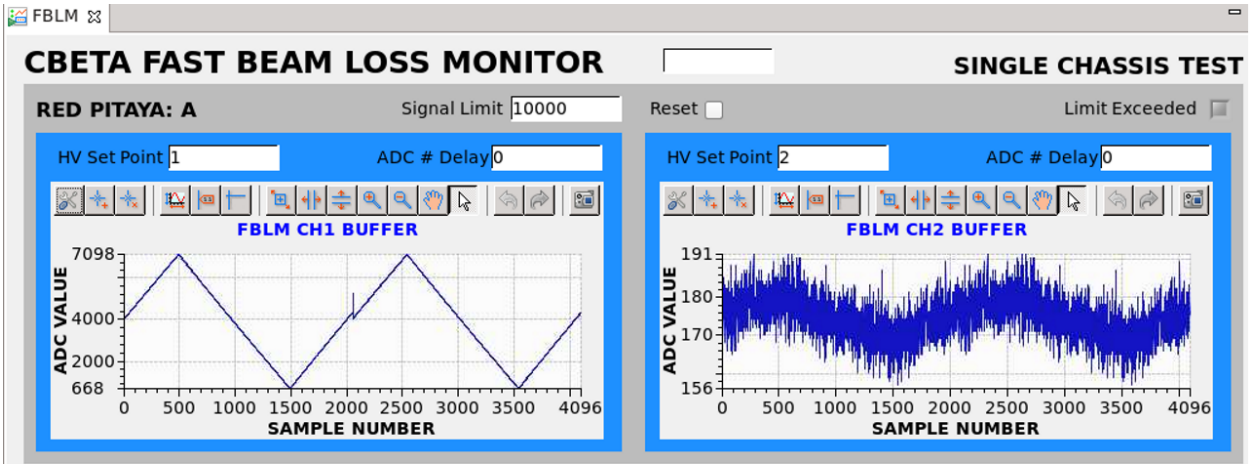


FIGURE 19 - FBLM.opi, example with data

Firmware

Vivado 2019 is used to synthesize and analyze the Hardware Design Language for the FBLM System. The software builds are provided in Figure 20. The FBLM firmware design includes the following file seen in Figure 21 and can be downloaded from the gitlab directory here: [7] CBETA_FBLM_FPGA.

```
Vivado v2019.1 (64-bit)
SW Build: 2552052 on Fri May 24 14:49:42 MDT 2019
IP Build: 2548770 on Fri May 24 18:01:18 MDT 2019
```

FIGURE 20 - Vivado Build Versions

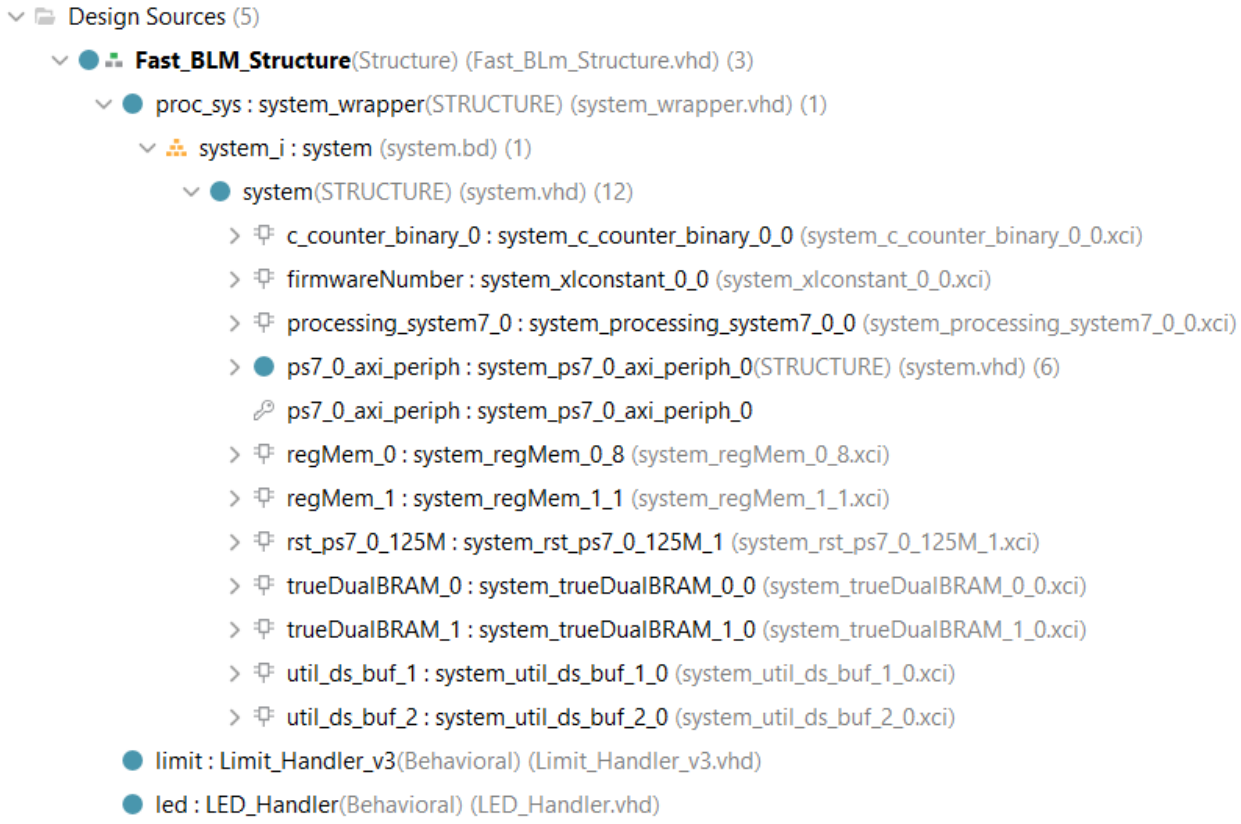


FIGURE 21 - FBLM Firmware, Design Files

The Block Design in Figure 21 shows that the ZYNQ processing system communicates with a set of Registers and TrueDualBlockRams via the Advanced eXtensible Interface (AXI) because the FPGA is ARM-based. These Registers contain inputs and outputs needed to update values in EPICS and CS-Studio. Both Registers and Block RAMs are accessed with the ReadData.c file discussed earlier. The Address Editor shown in Figure 22 provides the Base and High Addresses in memory for access.

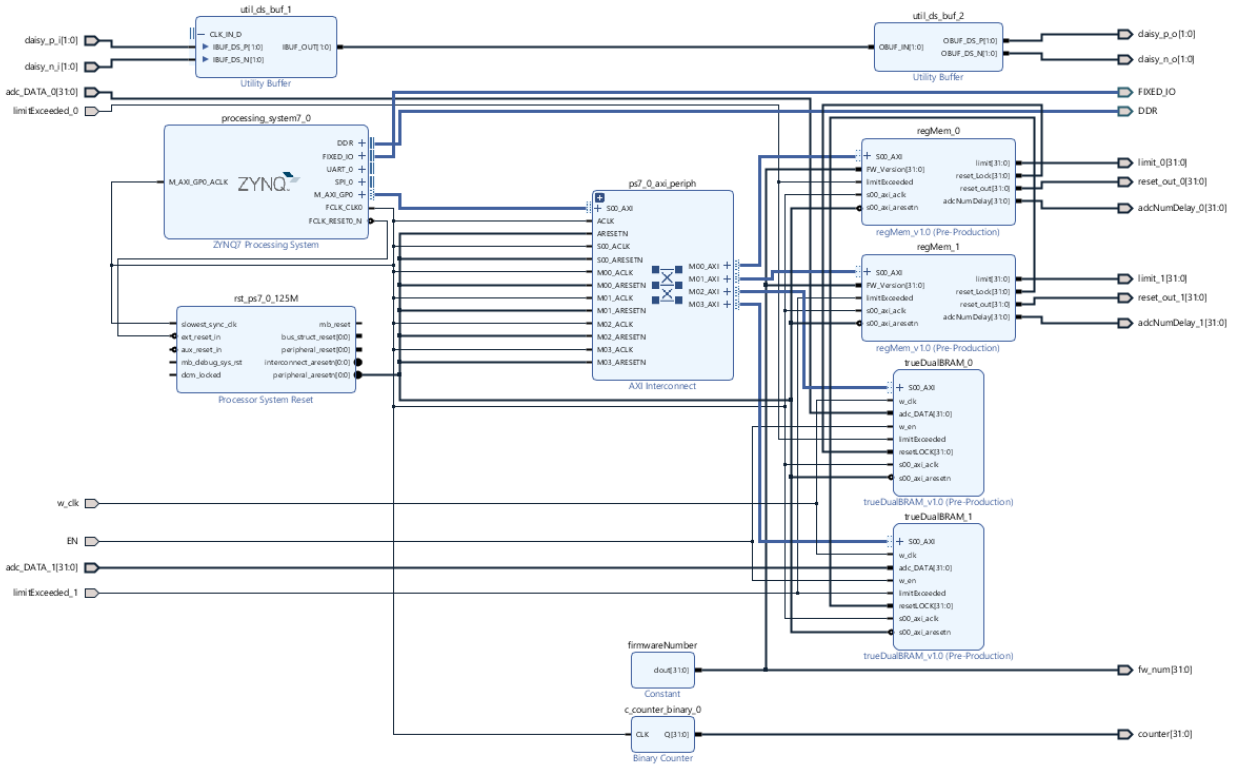


FIGURE 21 - Design Block diagram

Diagram | Address Editor

Cell	Slave Interface	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 0x40000000 [1G])					
regMem_0	S00_AXI	S00_AXI_reg	0x43C0_0000	64K	0x43C0_FFFF
regMem_1	S00_AXI	S00_AXI_reg	0x43C1_0000	64K	0x43C1_FFFF
trueDualBRAM_0	S00_AXI	S00_AXI_reg	0x43C2_0000	64K	0x43C2_FFFF
trueDualBRAM_1	S00_AXI	S00_AXI_reg	0x43C3_0000	64K	0x43C3_FFFF

FIGURE 22 - Address Editor

The Limit Handler from Figure 23 is responsible for taking in the 14 bit ADC and sign-extending it to 32 bits which goes into the buffer. It is also responsible for comparing the ADC samples with the Limit and Limit Window provided by the user and latching the low level signal out through “OUT1” if the conditions are satisfied. As you can see, there are duplicates which suggests that this handler handles both input channels on the Red Pitaya.

Other features of this design include reset, limit_exceeded and trig feedback. This design can reset the latched output signal at the request of the user as well as provide a signal when the limit has been exceeded. “OUT2” is special and designed to readback the trigger signal for debugging purposes.

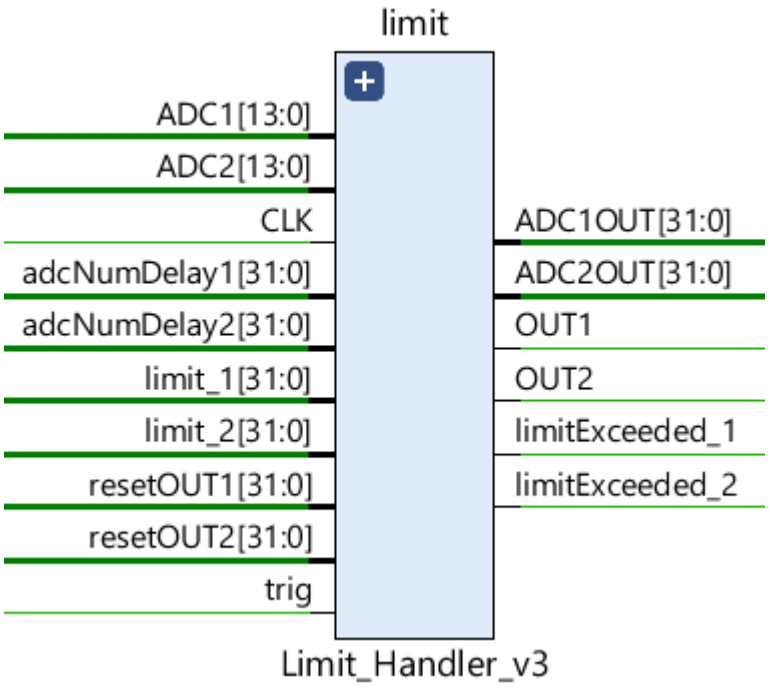


FIGURE 23 - LIMIT HANDLER

Project Status

Live Data from CBETA

The following graph in Figure 24 shows the first data collected from the actual beam and represented with Cornell's EDM application. A newer design providing better resolution has been implemented and will be shown at a later time, once beam operations begin again in the future.

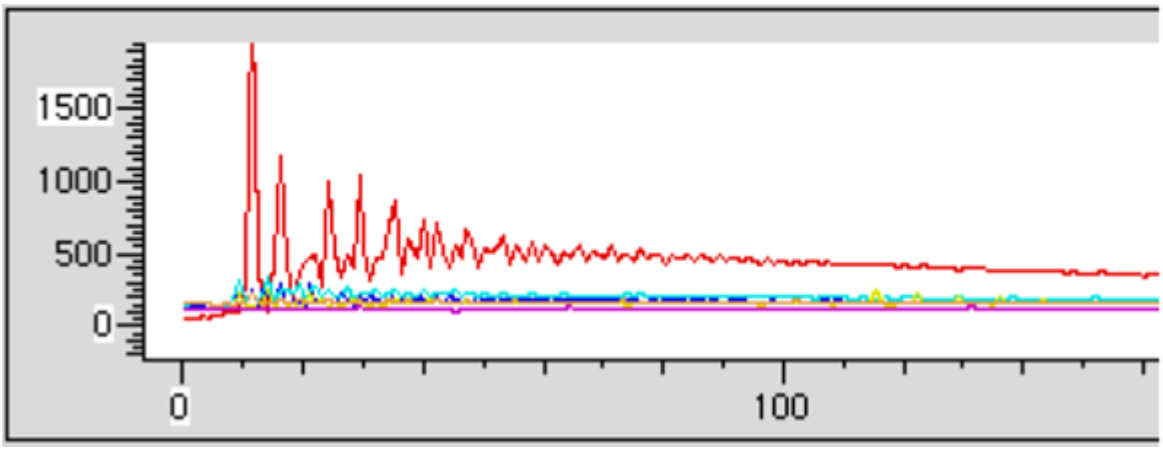


FIGURE 24 - DATA from CBETA (FBLM ver 1.0, JANUARY 29, 2020)

Post Mortem Data

Figure 25 has a red indication on “Limit Exceeded” suggesting that the buffer has been locked and the postmortem data can now be analyzed. Once the postmortem data has been analyzed, press the Reset button to unlock the buffer and continue reading new samples.

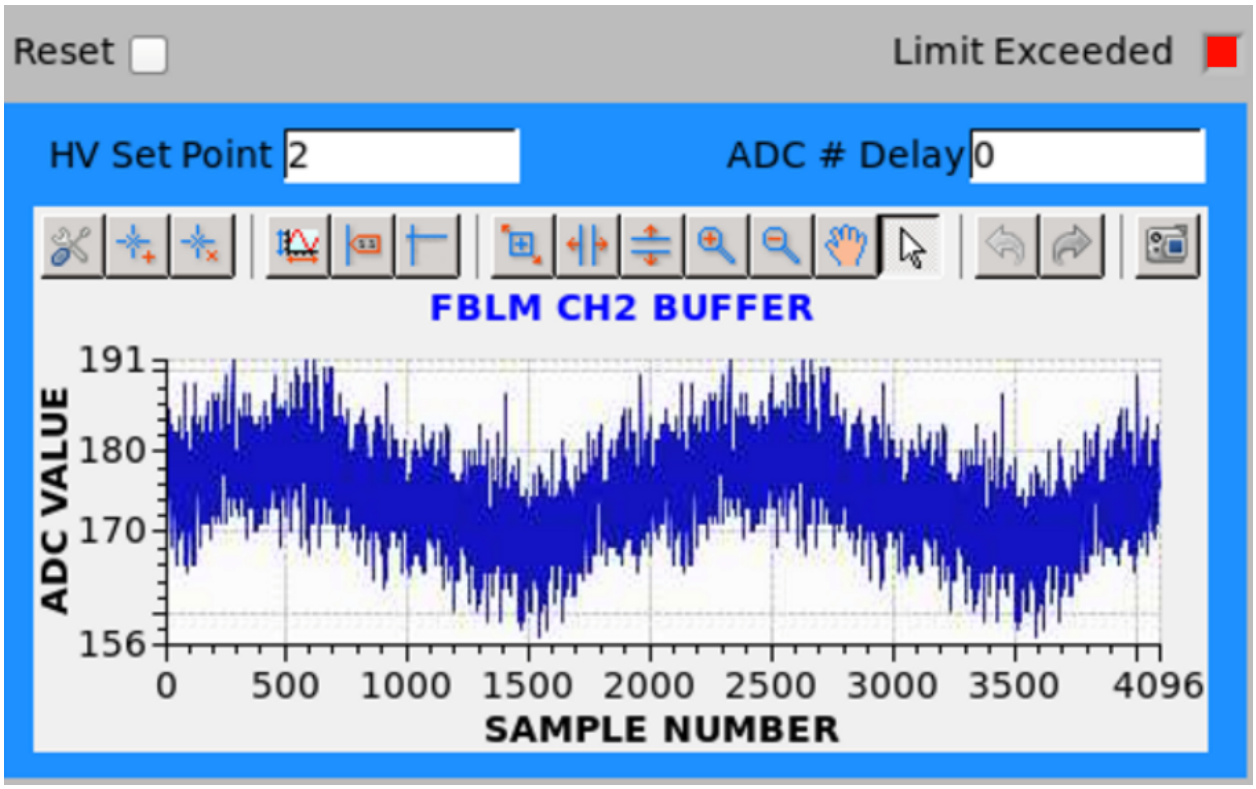
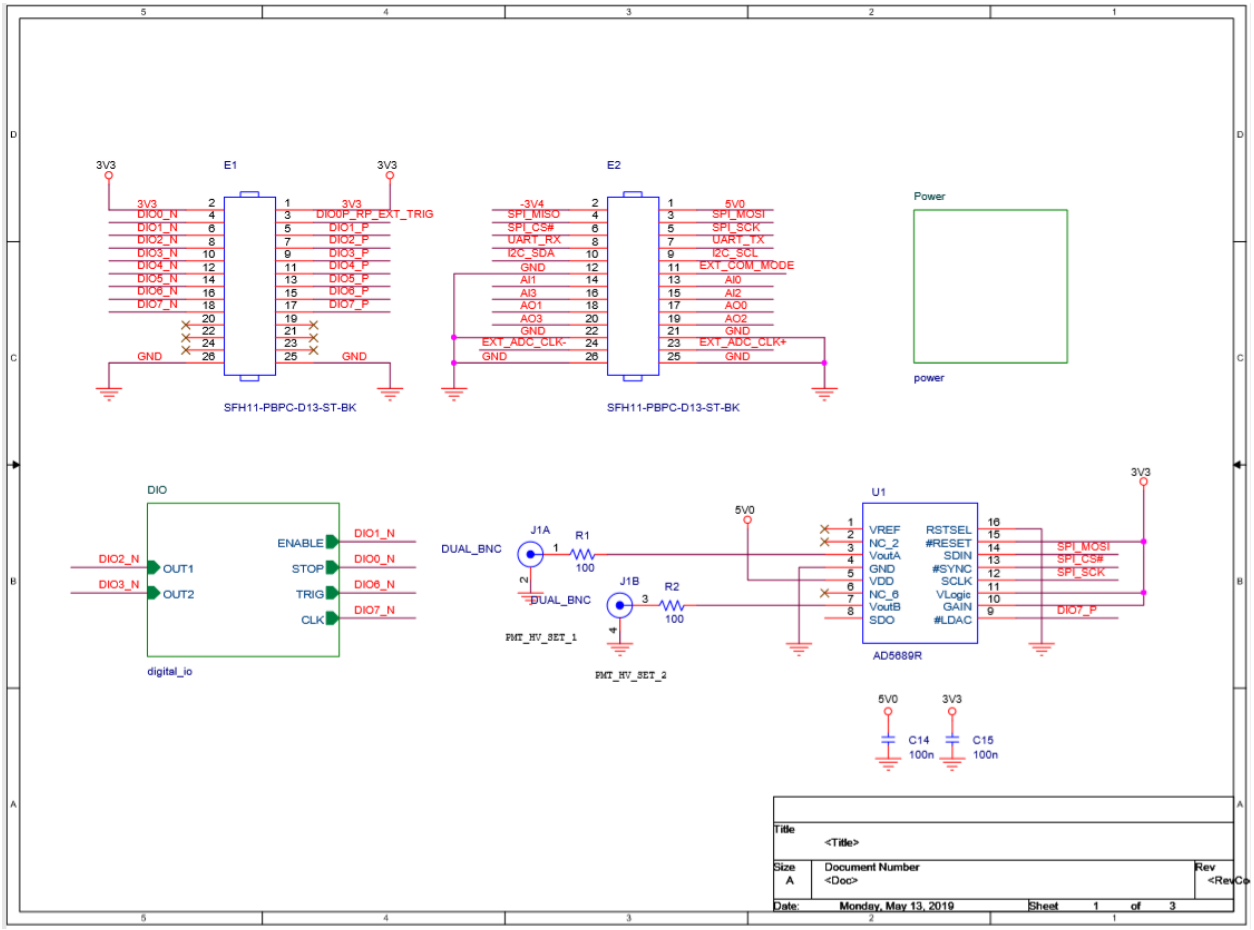


FIGURE 25 - Post Mortem Data

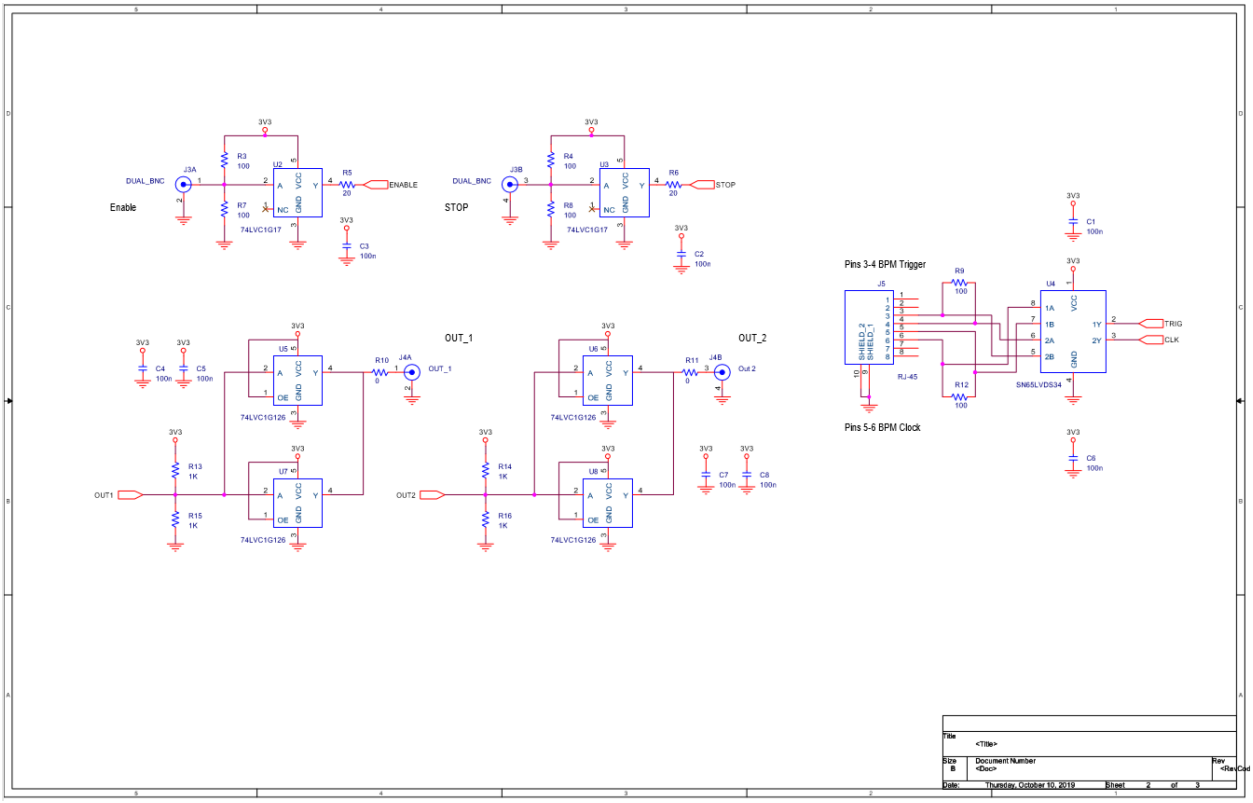
Schematics

DAC/Expansion Slots Schematic



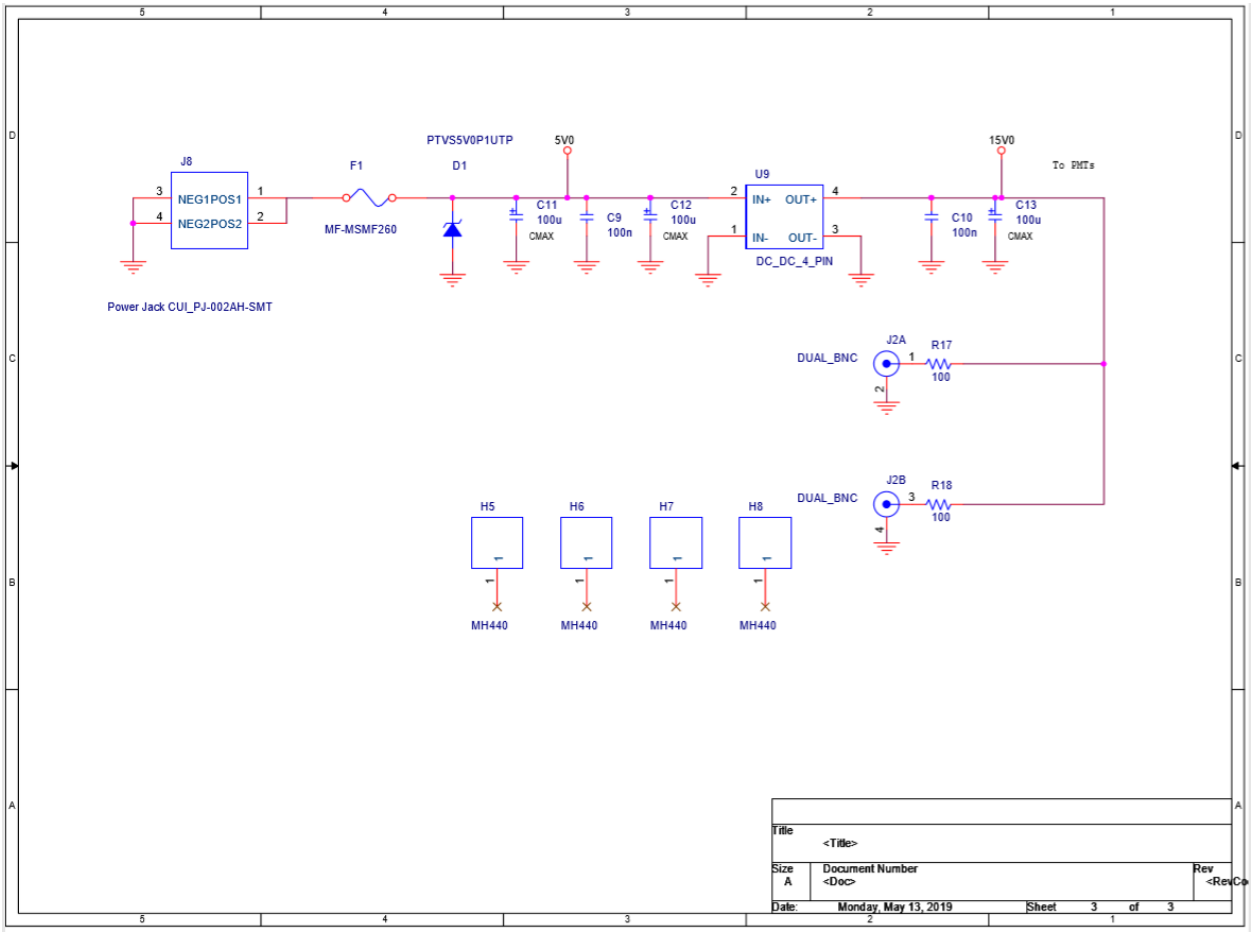
SCHEMATIC 1 - DAC & Expansion Slot Schematic

Tigger/Clock/Buffer Schematic



SCHEMATIC 2 - Trigger/Clock/Buffer, Expansion Board

5V/15V Power Schematic

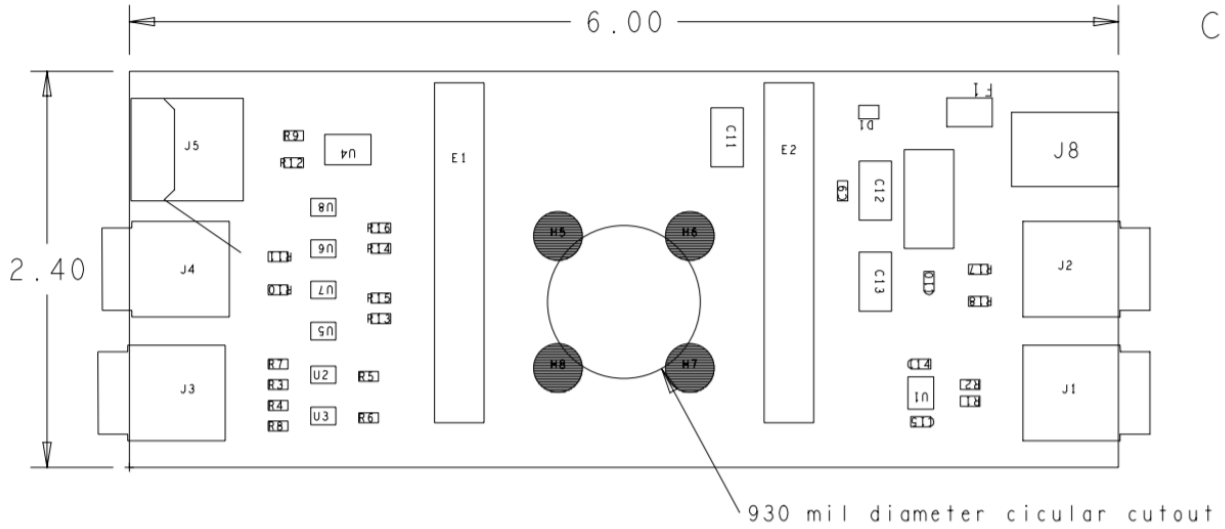


SCHEMATIC 3 - Power Rails, Expansion Board

Drawings

RP_Expansion_Assembly Drawing

The Expansion Board drawing, found below, was provided by John Dobbins from Cornell. This board uses its E1 and E2 headers to connect to the Red Pitaya Board.



DRAWING 1 - Expansion Board

References

[1] (CBETA FBLM Software, 2019)
https://gitlab.pbn.bnl.gov/Instrumentation_System_Integration_Group/CBETA_FBLM_SOFTWARE/

[2] (IP Form)
http://corvette.ags.bnl.gov/cgi-bin/IP/ipForm.pl

[3] (rp_spi.c, ver 1.1, 2019)
https://gitlab.pbn.bnl.gov/Instrumentation_System_Integration_Group/CBETA_FBLM_SOFTWARE/-/blob/master/rp_spi.c

[4] (ReadData.c, ver 1.0, 2019)
https://gitlab.pbn.bnl.gov/Instrumentation_System_Integration_Group/CBETA_FBLM_SOFTWARE/-/blob/master/ReadData.c

[5] (EPICS Record Manual, ver 3.13 1995)
https://epics.anl.gov/EpicsDocumentation/AppDevManuals/RecordRef/Recordref.pdf

[6] (HVSP2V.py, ver 1.0, 2019)
LINK TBD

[7] (CBETA_FBLM_FPGA, ver 1.0, 2019)
https://gitlab.pbn.bnl.gov/Instrumentation_System_Integration_Group/CBETA-Fast_BLM

[8] (BCF60-MC Scintillating Fiber)
https://www.crystals.saint-gobain.com/sites/imdf.crystals.com/files/documents/fiber-product-sheet.pdf

[9] (9349T1 Crack Resistant Tubing)
https://www.mcmaster.com/9349t1

[10] (R11558 28mm dia. Bialkali PMT)
https://www.hamamatsu.com/jp/en/product/type/R11558/index.html

[11] (XC7Z010-1CLG400C FPGA)
https://www.avnet.com/shop/us/products/xilinx/xc7z010-1clg400c-30744573456259883 68/