

BCAL pulse sequence  
Orlando Soto  
May 1, 2014

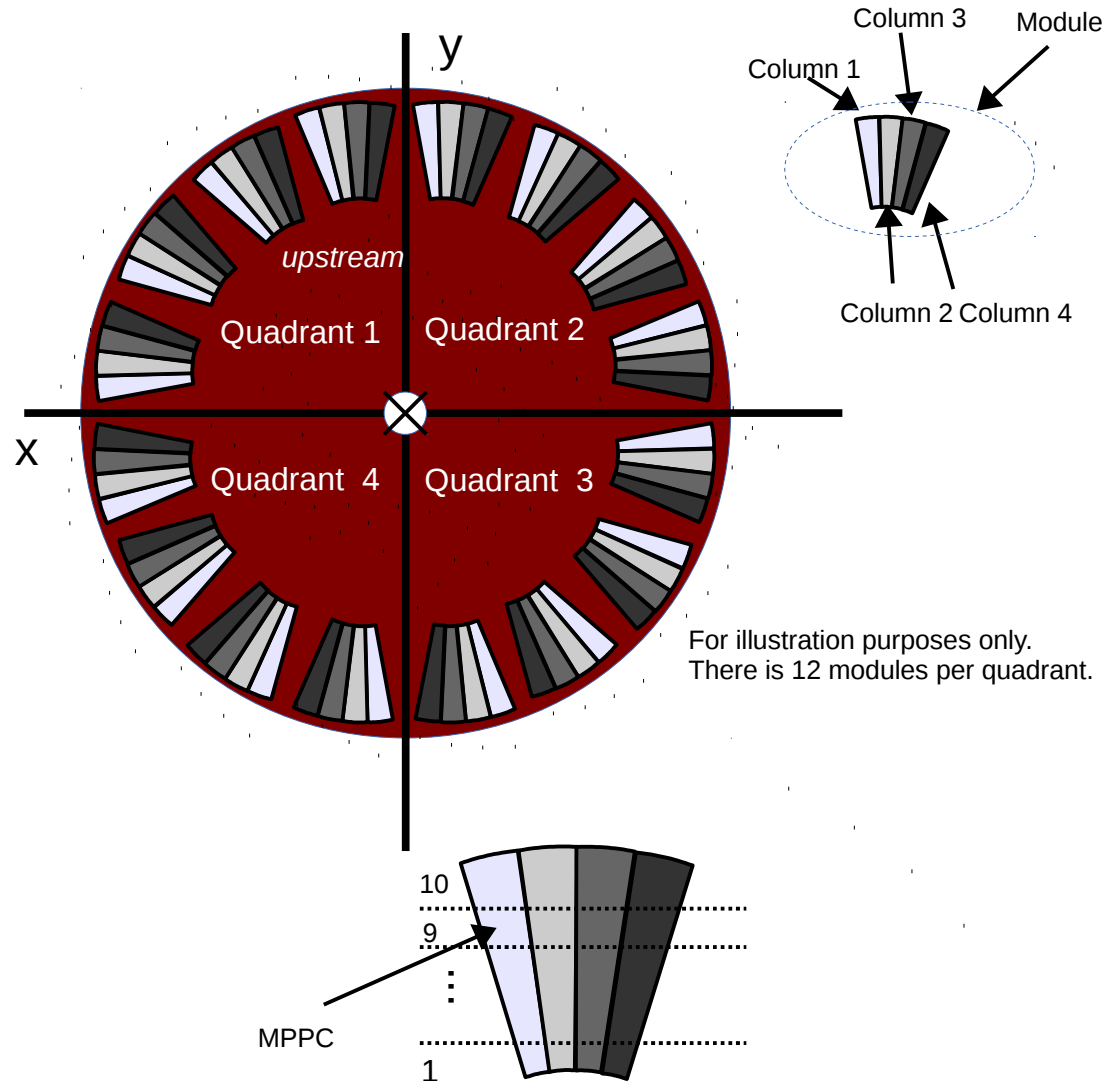
# Introduction

The outputs and power supply of the MPPCs in the BCAL are grouped in a special way.

Each module column is divided in 10 segments, each segment has an MPPC associated. There are 4 different voltage channels in each module each of this supply groups of 10 MPPCs with an special structure along the module.

The MPPC output groups are 4, and join 1, 2, 3 and 4 segments ( $\{1\}$ ,  $\{2,3\}$ ,  $\{4,5,6\}$ ,  $\{7,8,9,10\}$ ).

In order to know if every MPPC in the detector is working properly, every voltage channel must be turned on alone and then every led in each column must be pulse alone, giving 16 combination per module.



To perform this task, a python script was designed.

# Routine

- Select Upstream or Downstream.
  - + Setting parameters values (MPPC bias, LED parameters, etc... )
  - + Turn on all low voltage for MPPCs and LEDs.
  - + Turn on all bias for LEDs.
  - + Turn off LED pulsers and disable all.
  - + Turn off all MPPC bias.
  - + Enable Trigger.
  - + Start loop for Quadrant.
    - Wait until all low voltages and LED bias are on.
    - Start loop for MPPC bias channels (k: 1,...,4)
      - + Turn on bias channel k for every module in the quadrant both sides.
      - + Wait until channel i is on and the three other are off.
      - + Start loop for Column (j: 1,...,4)
        - Enable LED column j.
        - Enable .
        - Start pulsing.
        - Wait : (npulses/frequency\*1.2+0.1) seconds.
        - Wait until LED column j is off.
        - Disable LED column j.
    - + Turn off channel k
  - + Turn off all low voltage for MPPCs and LEDs.
  - + Turn off all bias for LEDs.
  - + Return.

It must include two epics variables:

- State information (longout).
- Synchronization (bo), semaphore handled by the DAQ, default value "green".

# Notes

- The required external python modules are: epics and sqlalchemy.
- The script handle Ctrl-c by shutting everything off and then exit.
- The script is located in:  
[https://halldsvn.jlab.org/repos/trunk/online/daq/tools/bcal\\_pulse\\_sequence/](https://halldsvn.jlab.org/repos/trunk/online/daq/tools/bcal_pulse_sequence/)
- In this location there are two files:  
    BCALModule.py : Python module containing python classes used in the routine.  
    BCAL\_test.py : The scripts that performs the routine (the name could be change)
- The file BCALModule.py could be used to program other routines easily.
- The script includes the following help:

```
./BCAL_test.py [option value]
```

options:

- s < U | D >: Select side U for upstream or D for downstream. If this option is not given, it will run through both sides.
- q < number >: Select quadrant with number in {1,...,4}. If this option is not given, it will run through all quadrants.
- f < freq >: Set pulser frequency in Hz. Default value 100.000000 Hz
- w < width >: Set pulser width in ns. Default value 100 ns
- n < npulses >: Set pulser number of pulses before stop. Default value 1
- t < temperature >: Set mppc temperature in C, used in small temperature corrections. Default value 18.000000 C
- o < offset >: Set mppc voltage above breakdown. Default value 1.200000 V
- single: Turn on mppc bias in one side only.
- h,--help: Show this help.

Note: Ctrl-c interruption handled, it will turn all off and then exit.

Example:

```
./BCAL_test.py -s U -q 1 -f 100 -w 100 -n 1
```

Questions, Comments or Doubts?