Black Box data taking

Carmen Romo Luque

April 15, 2021

1 First steps

- Place the LED in the setup and connect it to the pulse generator (PG). To be sure that it works, in the PG you can do:

  ```
  Utility -> output setup -> inverted
  Amplitude of the pulse: 3V, output ON
  ```

  The LED will emit visible light if it works and if it's correctly connected (Figure 1). Remeber to switch to normal output setup after this test. Note that the light will be visible only in the inverted mode.

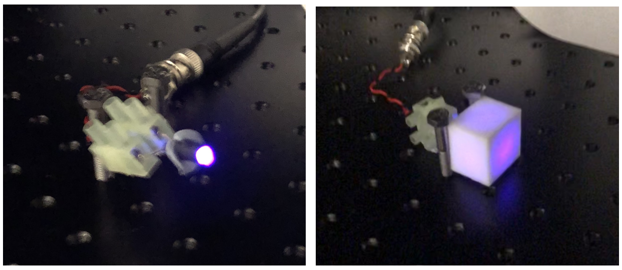

Figure 1: Naked LED (left) and encapsulated within a teflon block (right).

- Place the SiPM dice board and connect the cable to the electronics cable.
- In case Demo++ is taking data: stop the run (save the configuration), switch OFF sensors and HHV.
- Make sure the blackbox frontend is in the rack, otherwise place it in there. Connect the green plugs (Figure 2) and the back cable.

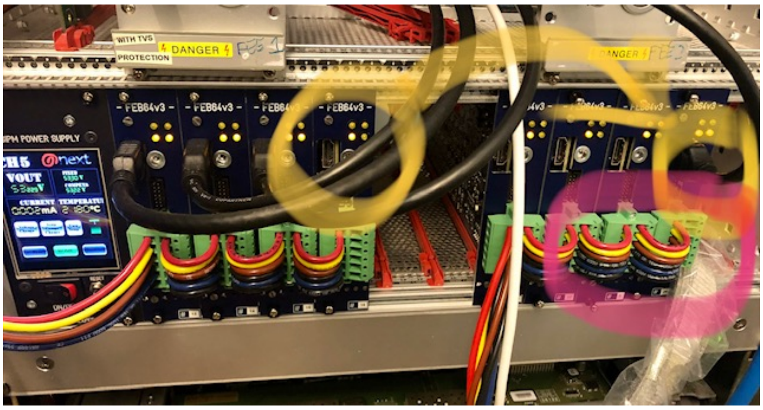

Figure 2: Frontends connections: hdmi cable (yellow line) and green plugs (pink line).

- Change the hdmi cable from the fourth demo frontend to our frontend (yellow line).

- Switch ON the frontends power supply (it is located under the table, close to the rack) by clicking the output buttom .

- In case Demo++ is OFF, electronics and power supplies must be set to ON in Anydesk (SiPMs FE and SiPMs PS).

- Set the bias voltage to the sensors: SensL: ∼ 27.65V and Hamamatsu: ∼ 53.75 V (Figure 3 right).

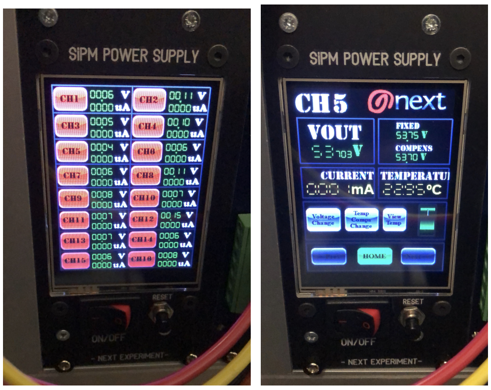

Figure 3: SiPMs power supply screen. List of all the channels (left) and parameters for channel 5 (right).

2 Pulse generator configuration:

In the pulse section

- **Frequency:** 10 kHz
- **Period:** 100 μs
- **Amplitude:** variable (depends on the sensors, but common numbers: between 2V and 5V).
- **High level:** + Ampl/2
- **Low level:** - Ampl/2
- **Offset:** 0
- **Pulse width:** 175 ns
- **Edge time:** 5ns

In case more than one pulse is needed, you can set a train of pulses ⇒ burst.

In the burst section:

- # **cycles:** 5 (for example).
- **Burst period:** 5 ms

In the amplitude section change these parameters:

- **Frequency:** 100 kHz
- **Period:** 10 μs

3 JAVA configuration:

- **Circular buffer:** 200 ms
- **Pretrigger:** 100 ms
- **Trigger frequency:** 200 Hz
- **Trigger mask:** ON
- **External trigger:** ON

- **Auto trigger:** OFF
- **Zero Suppression:** OFF
- **Presamples:** 0
- **Number of events:** variable. For tests, 10 events is enough, for bursts with 5 pulses, 10.000 events is ok, if only one pulse, 30.000 events is ok.

4 Take a run:

Once the JAVA is configured:

- Click all the ”Config register” in the JAVA.
- In date (first tab), click Start Processes.
- **LED output ON**
- In date click Start.
- In JAVA click Start Run.

In case the trigger is not working and events are not being registered, perform a Java RESET.

5 Processing data

Once the run has finished, it should be proccessed. In the case of the BB runs, the procedure is the following:

- Access to frontend1demo:

  ```
  ssh  -Y hwuser@frontend1demo.ific.uv.es
  ```

- Activate the environment and enter the directory:

  ```
  rawdata
  cd CheckFEB
  ```

- Decode the run (transform from binary to hdf5 waveforms):

  ```
  python process_demo.py -r <run_number>
  ```

-m limits the number of files to decode.

The files will be written in:

```
/home/hwuser/h5files/<run_number>/data
```

- Check the pulse in the waveforms:

  ```
  python plot_canfranc.py --file
  h5files/<run_number>/data/run_<run_number>.gdc1demo.next-demo.000.rd.h5
  -sipm --sipm-range 25000 25063
  ```

  This will represent event to event and sensor to sensor, all waveforms.

  In the case of the black box the elec ids of the sensors go from 25000 to 25063.

  Other useful options are:

  ```
  --overlay
  --sum
  --file <file_number>
  ```

  That will represent for each event all the sensors together, the sum of their waveforms or the desired file, respectively.

- To create the spectra the city called Trude must be executed (an script can be provided).

  The spectra are stored in:

  ```
  /home/hwuser/CheckFEB/calibration/spectra/
  ```