

Calibration Instruction - 31/05 - Flo

This document offers a quick overview of the calibration process for a single detector unit with a signal generator TTi TG5011 which has the calibration waveform stored.

A dedicated script controls the signal generator and triggers the calibration signal waveform, in parallel, the scripts reads out the detector signal and stores it as a csv file. Once the signal generator is connected to the PC, it switches automatically to REMOTE mode, disabling manual input. Restart the signal generator without connecting it to the PC to retrieve the LOCAL mode.

Instructions

1. connect the detector unit **without scintillator** (and Oled.ino uploaded) to with the micro USB to the PC

2. retrieve the name of the USB port on the PC: open the terminal and run
`ls /dev/ttyUSB*`

3. turn on the signal generator, and connect the MAIN OUT (right BNC) to the detector

4. connect the signal generator with a macro USB (at the back) to the PC, again, retrieve the port name (the signal generator will appear as a ACM port):

```
ls /dev/ttyACM*
```

5. grant the user permission to the ports

```
sudo chmod 666 /dev/ttyUSB<see above>
```

```
sudo chmod 666 /dev/ttyACM<see above>
```

These commands are needed each time you reconnect either the detector or the signal generator. If any script throws you any permission error regarding these ports, just rerun these commands

6. in the terminal go to the directory `muon/Nikhef-Project-2021`

7. **ensure having the latest scripts from main and that you are working in your own branch, etc.** (see git manual e.g. `git checkout <>`)

8. change to the directory `Calibration/SignalCalibration/`

9. source the correct python path

```
. setup.sh
```

10. switch to `share/`

11. execute the calibration script with your desired options parsed to it (e.g. the USB port, V_{min} , etc.). To see all options call the helper with

```
python3 simple_calibration.py -h
```

An example command could look like this

```
python3 simple_calibration.py --p /dev/ttyUSB3 --fname banana --nsteps 5
```

12. you can interrupt the execution by `Ctrl-C`

13. while the script is running, verify the signal generator display, see Figure 1

14. the file is saved in `fpath`, by default this is `SignalCalibration/data/`

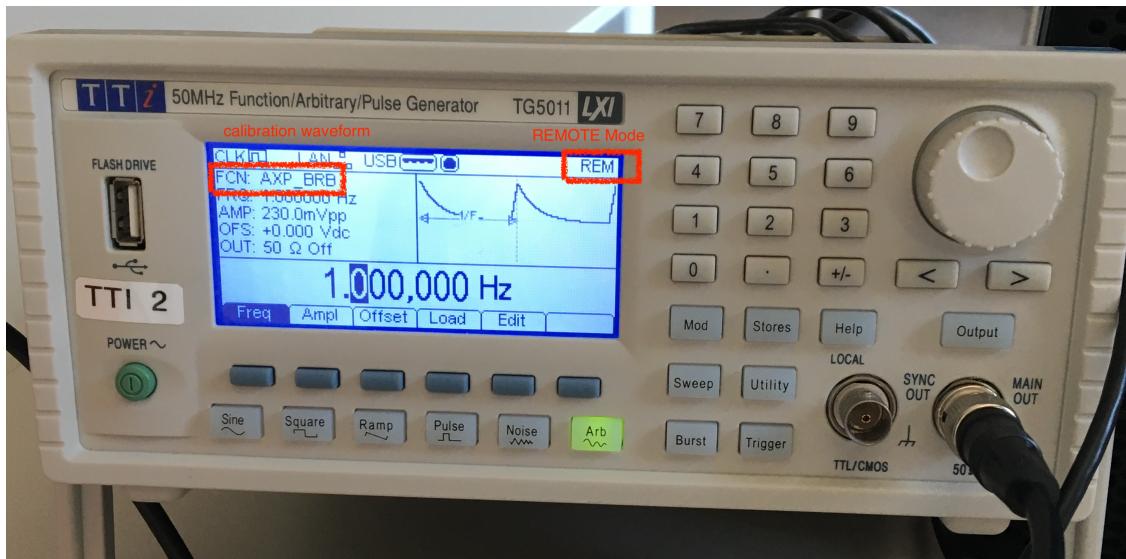


Figure 1: The display as it should look like during the calibration. The BURST button will blink orange each time a new signal is injected