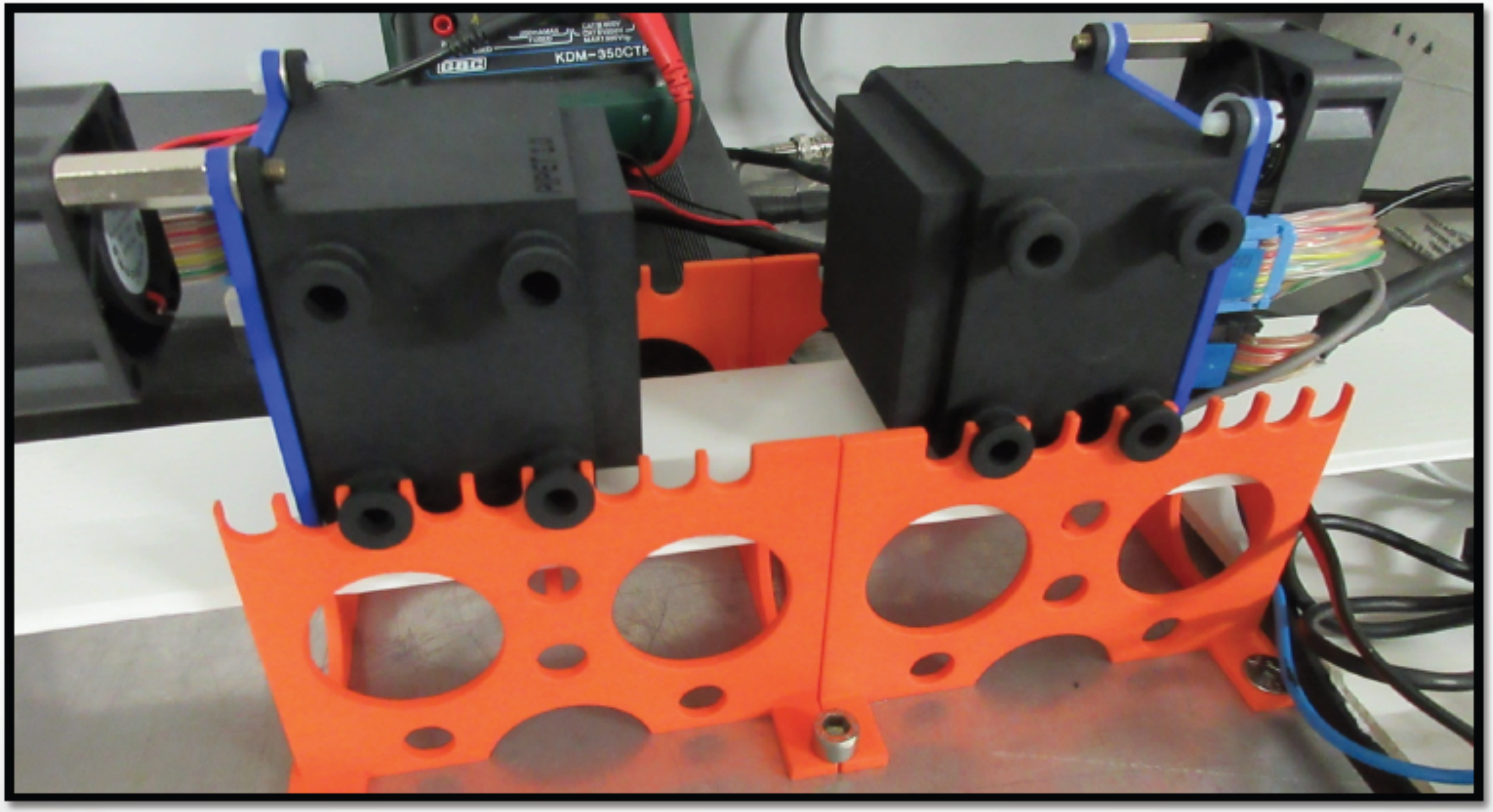
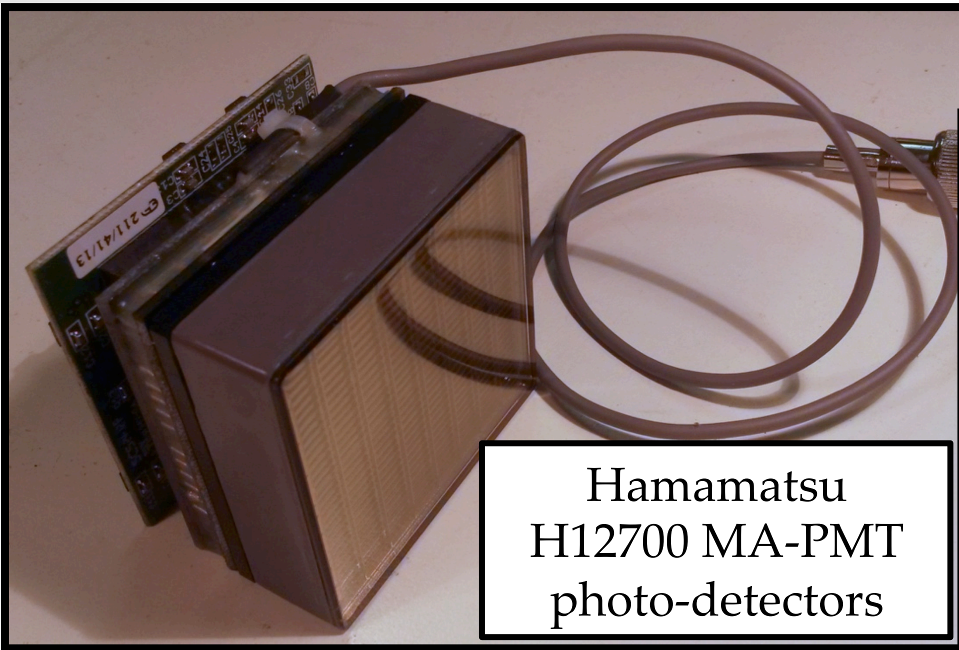
A PET training demonstrator

Dr. Giancarlo SPORTELLI
Department of Physics, University of Pisa and INFN, Italy

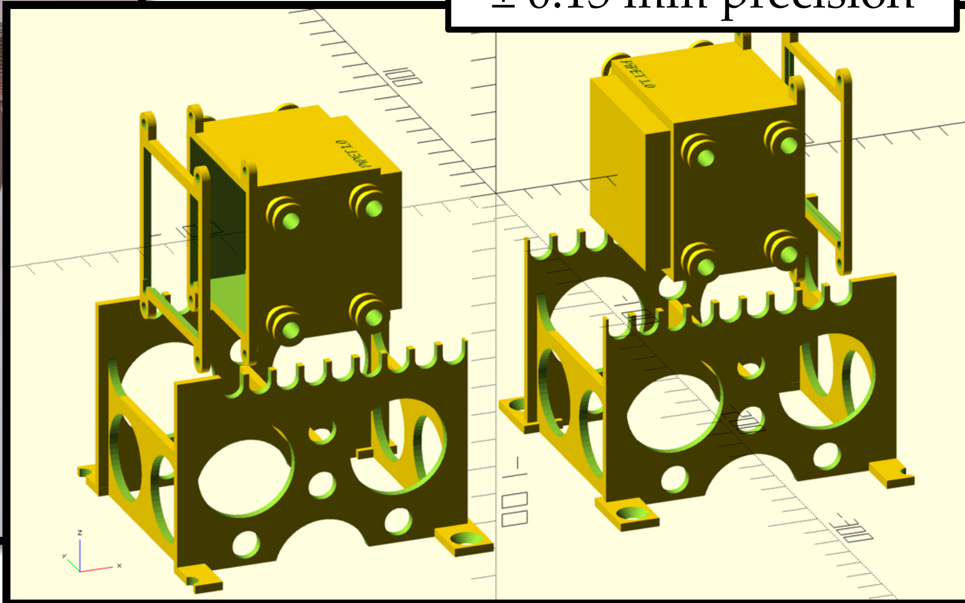
- pipet



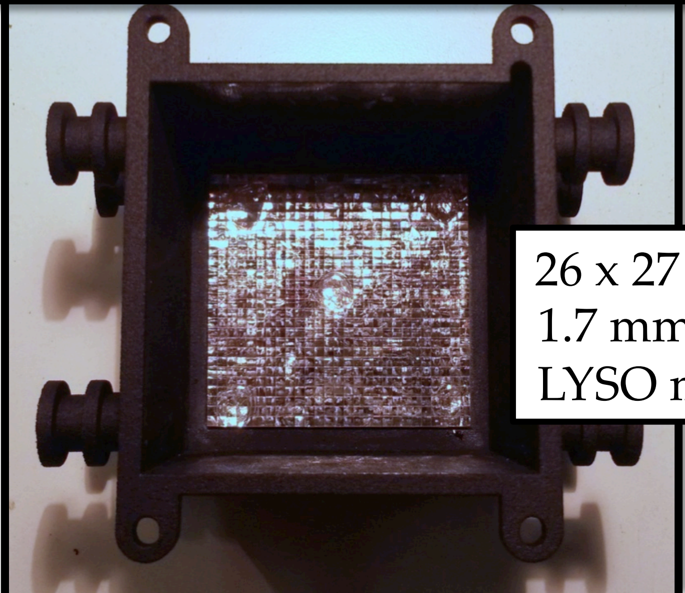
PiPET main specs



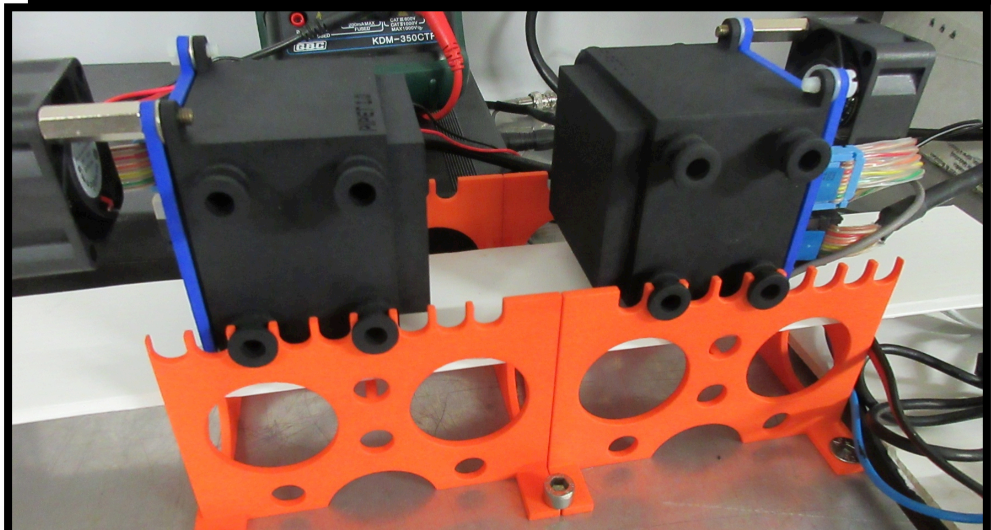
Hamamatsu H12700 MA-PMT photo-detectors



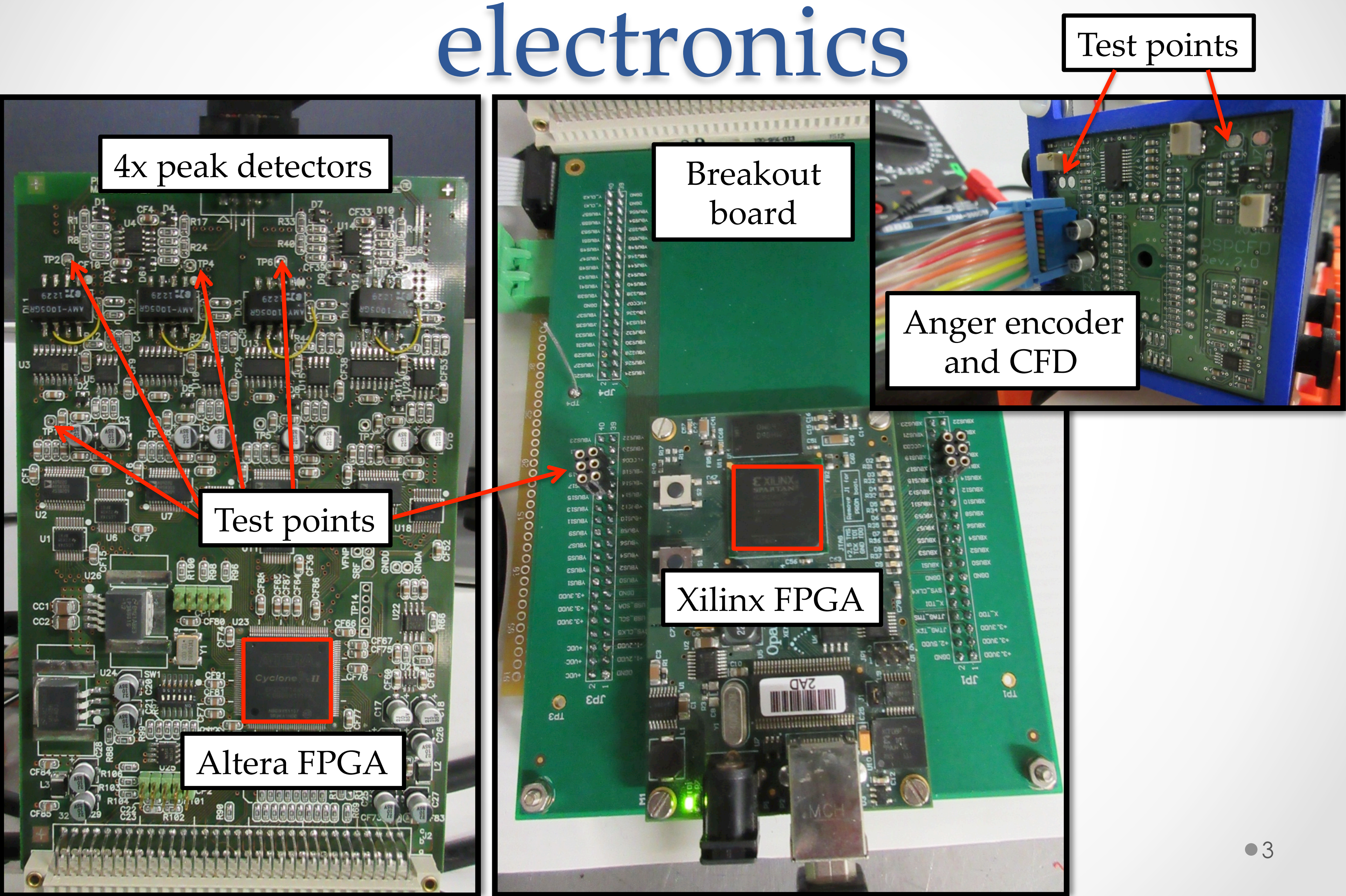
3D printed mechanics ± 0.15 mm precision



26 x 27 pixels 1.7 mm pitch LYSO matrix



Accessible FPGA-based electronics



4x peak detectors

Test points

Altera FPGA

Breakout board

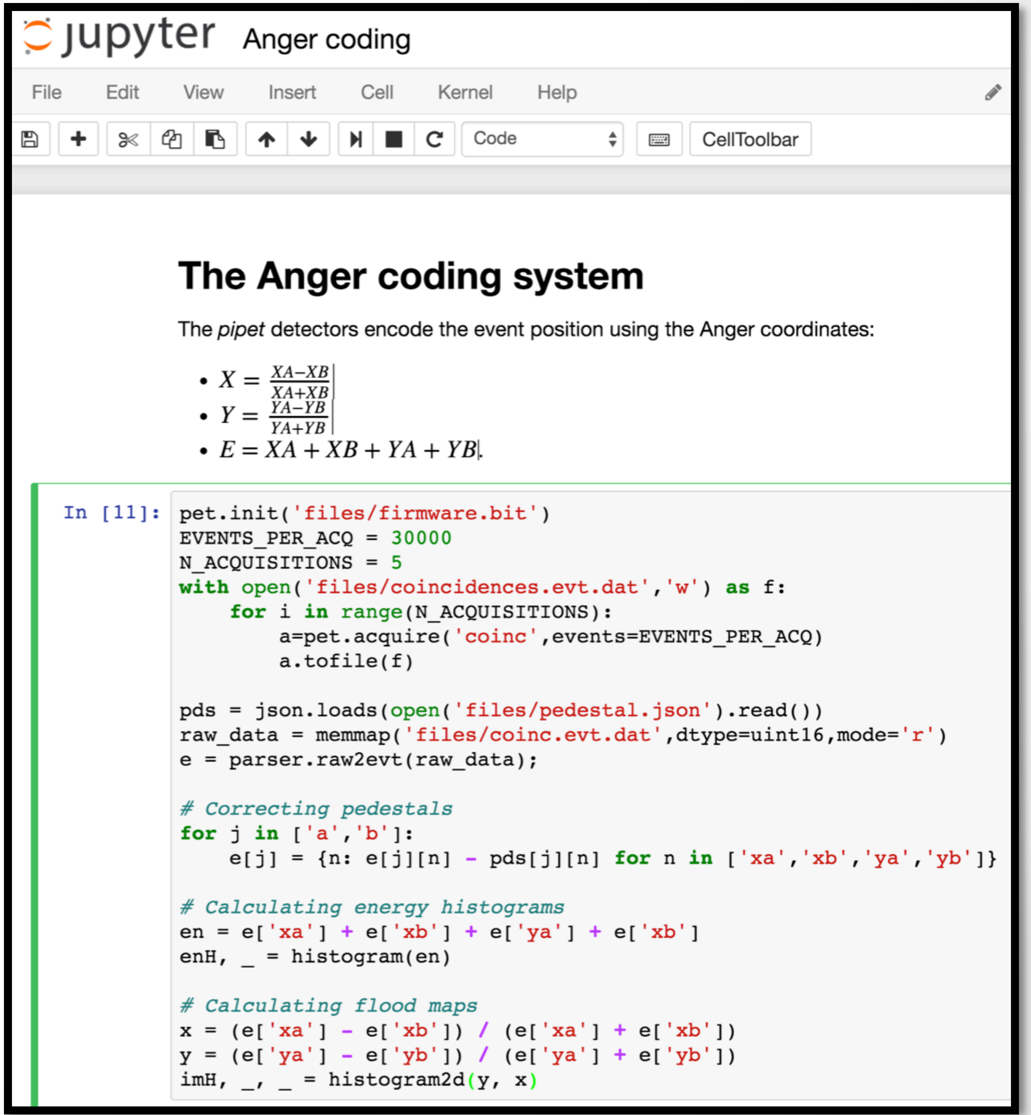
Xilinx FPGA

Test points

Anger encoder and CFD

Python-based high level development framework

- Connect remotely with an own laptop
- Follow example calibration techniques
- Measure key characteristics of the PET system
- Customize firmware and software to improve system performances



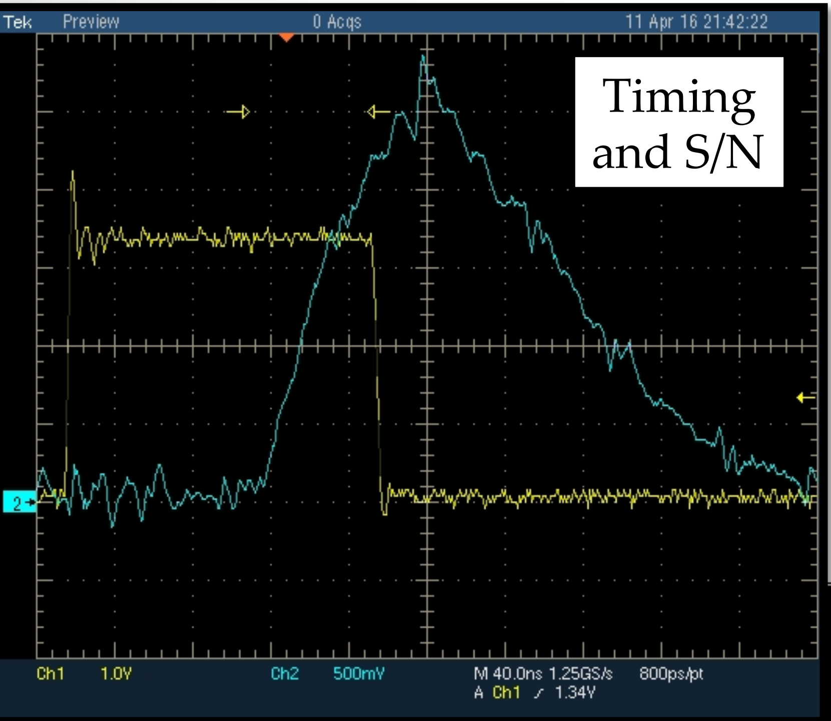
The Anger coding system

The *pipet* detectors encode the event position using the Anger coordinates:

- $X = \frac{XA-XB}{XA+XB}$
- $Y = \frac{YA-YB}{YA+YB}$
- $E = XA + XB + YA + YB$.

```
pet.init('files/firmware.bit')
EVENTS_PER_ACQ = 30000
N_ACQUISITIONS = 5
with open('files/coincidences.evt.dat','w') as f:
    for i in range(N_ACQUISITIONS):
        a=pet.acquire('coinc',events=EVENTS_PER_ACQ)
        a.tofile(f)

pds = json.loads(open('files/pedestal.json').read())
raw_data = memmap('files/coinc.evt.dat',dtype=uint16,mode='r')
e = parser.raw2evt(raw_data);

# Correcting pedestals
for j in ['a','b']:
    e[j] = {n: e[j][n] - pds[j][n] for n in ['xa','xb','ya','yb']}

# Calculating energy histograms
en = e['xa'] + e['xb'] + e['ya'] + e['xb']
enH, _ = histogram(en)

# Calculating flood maps
x = (e['xa'] - e['xb']) / (e['xa'] + e['xb'])
y = (e['ya'] - e['yb']) / (e['ya'] + e['yb'])
imH, _, _ = histogram2d(y, x)
```

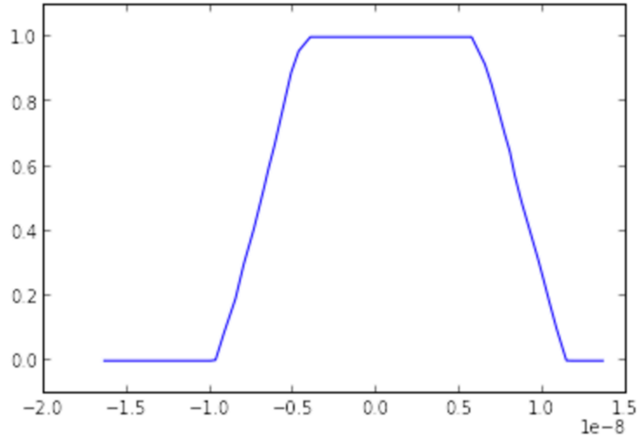
Calibration exercises

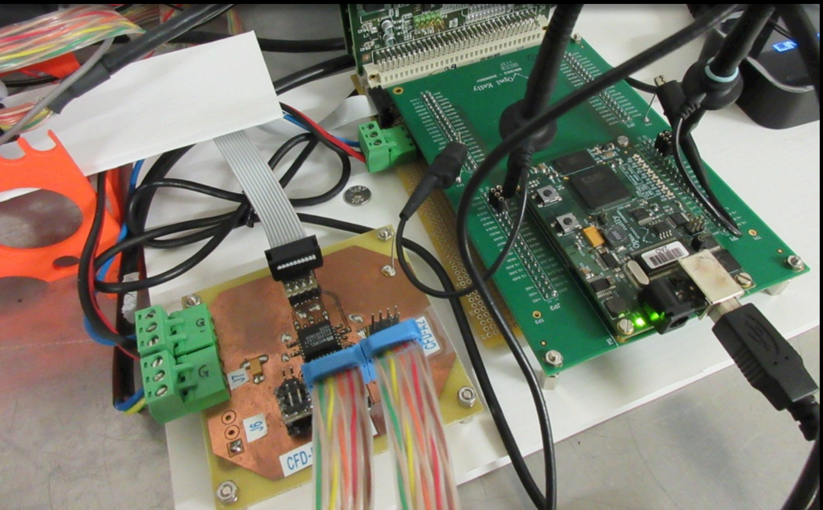


Timing and S/N

Coincidence window

```
Expected FWHM = 15.62 ns
Measured FWHM  = 16.98 ns
Offset = 1.12 ns
```





Pixel identification