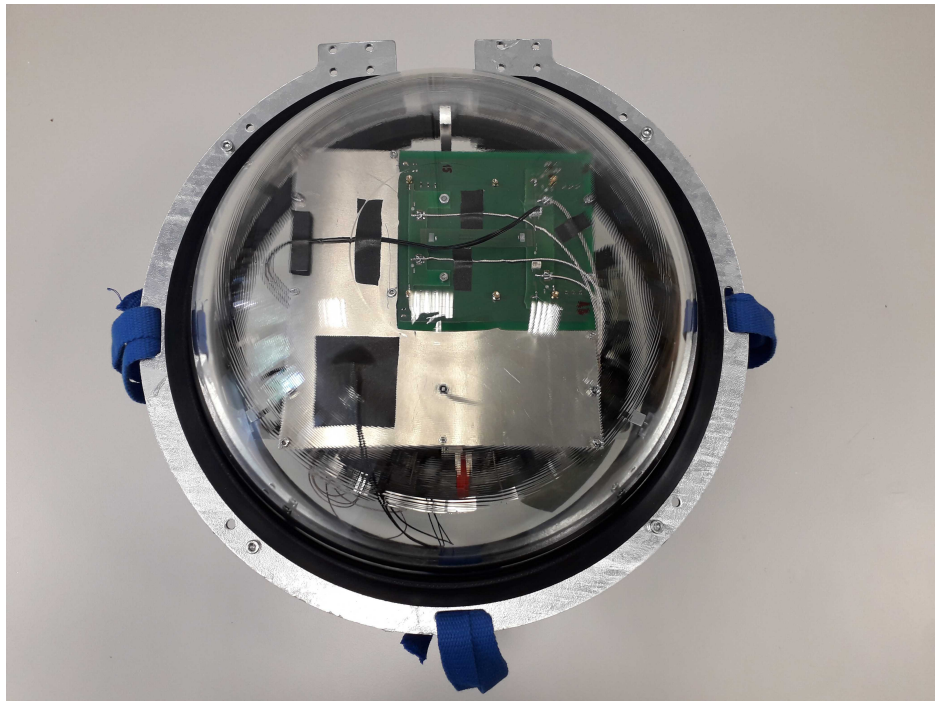
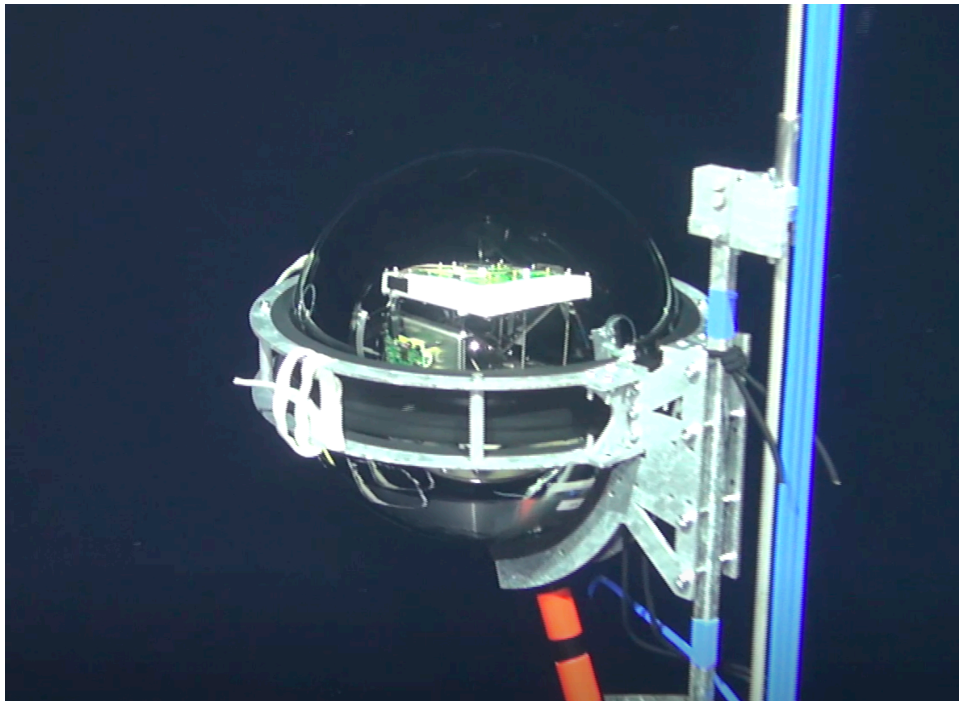
The Muon Tracker

Specialised Module of STRAW-b

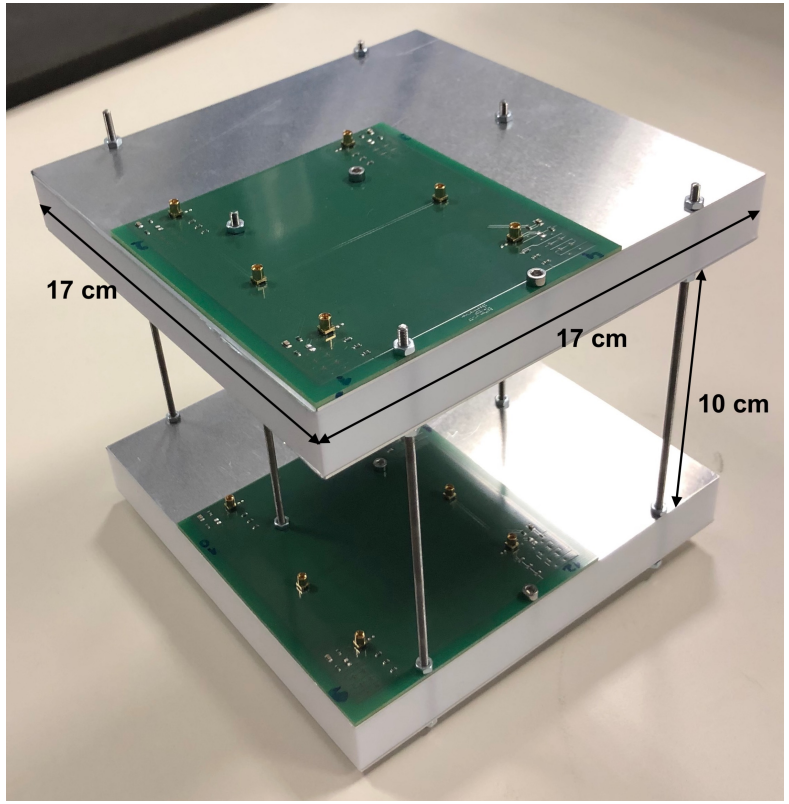


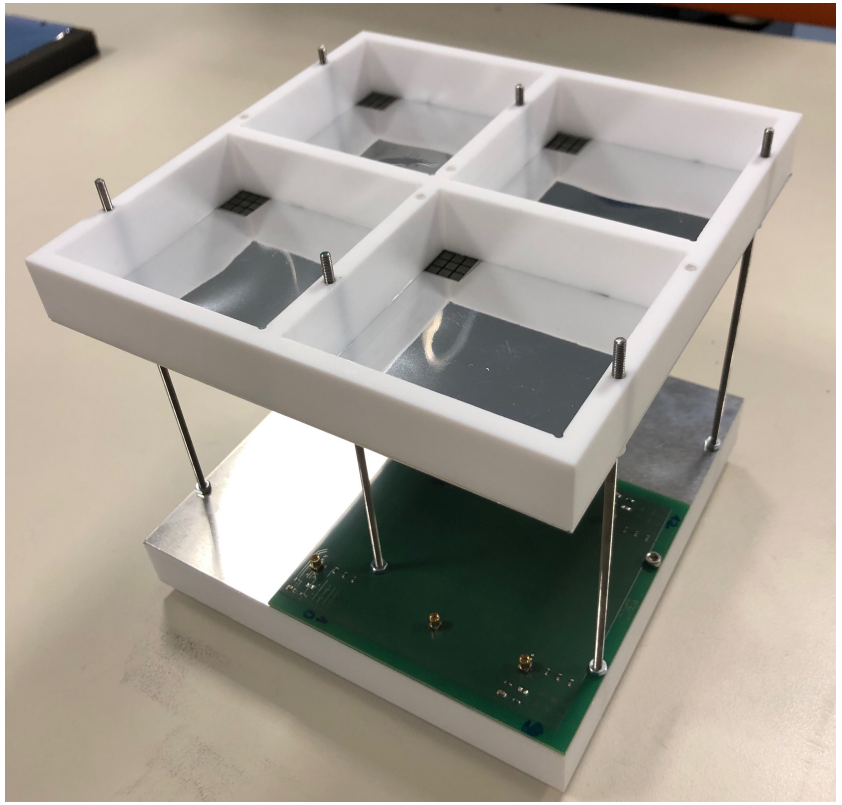


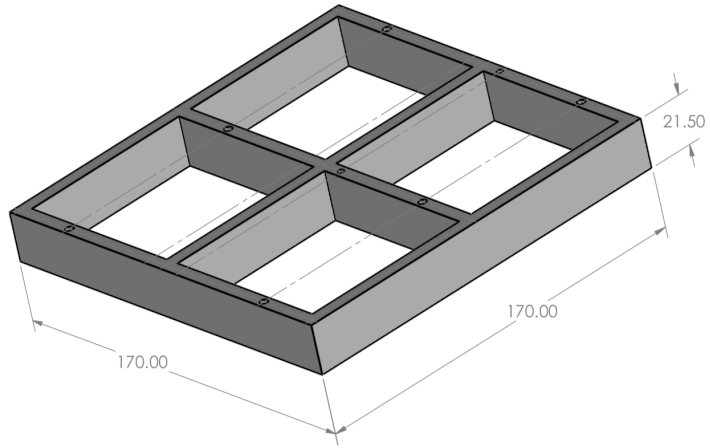
L. Winter, Na. Khera, F. Henningsen, C. Spannfellner

Structure

- Two housing boxes
  - Teflon frame
  - Reflective foil
  - Aluminium plates
- Two 2x2 plastic scintillator arrays (BC-404)
- Four PCBs with SiPMs

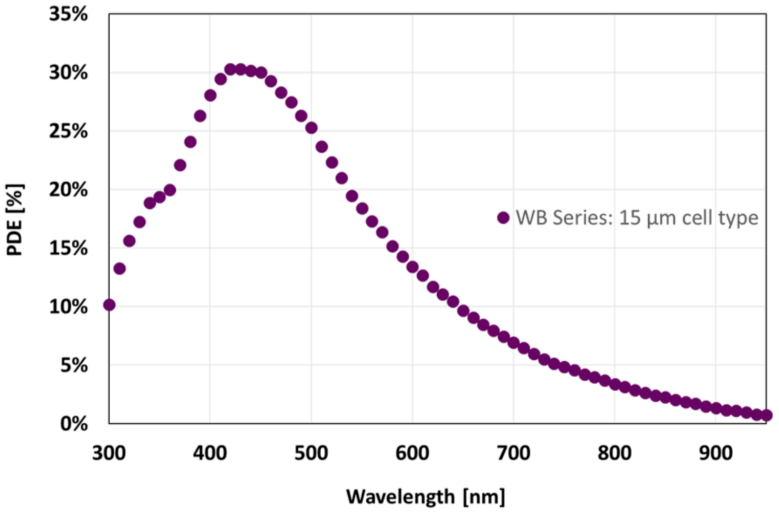


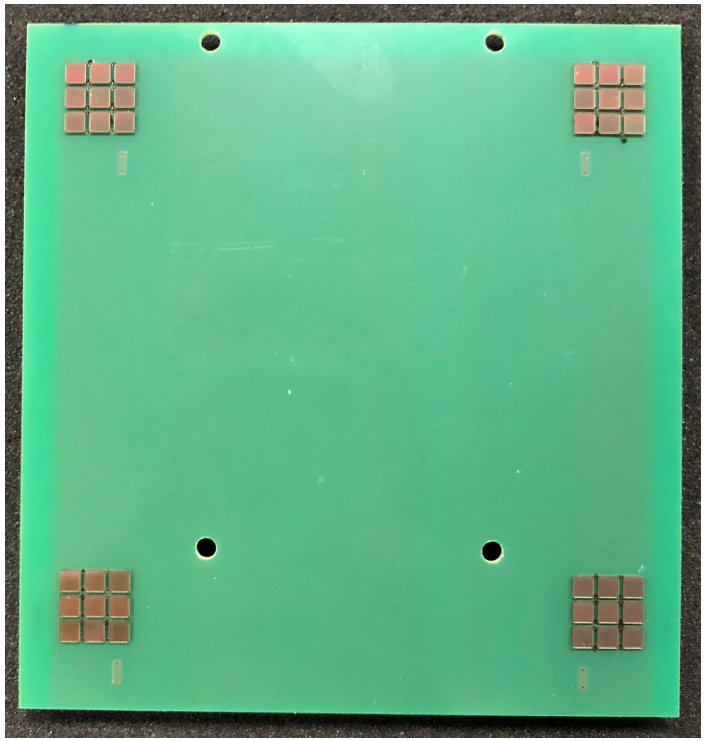


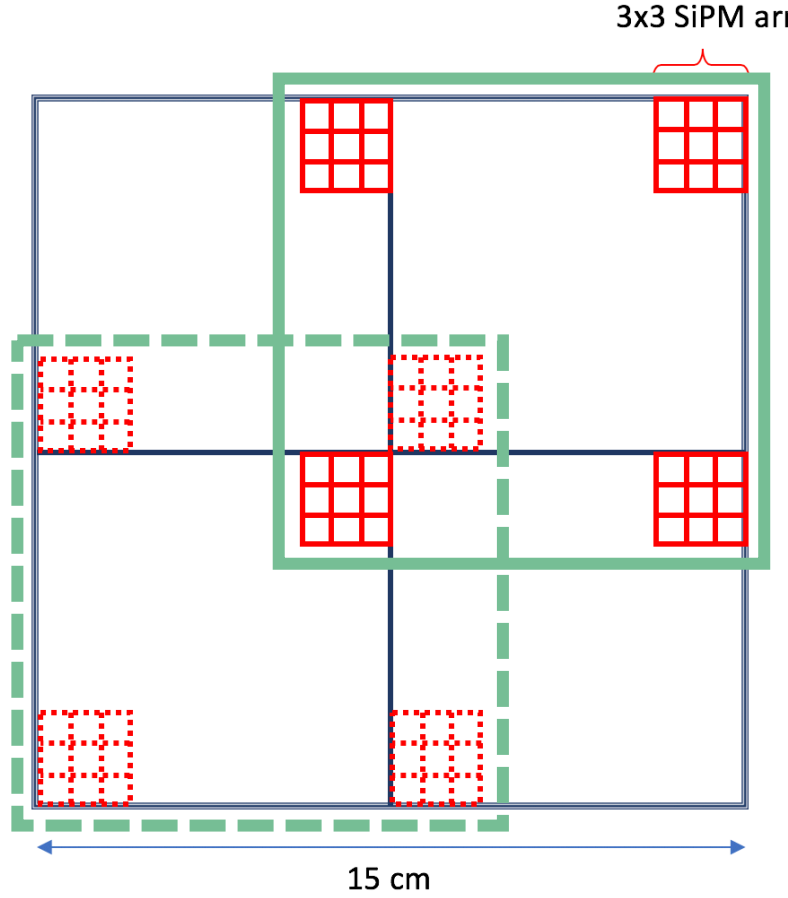


Light Detection with SiPMs

- PCB with four 3x3 SiPM arrays (KETEK PM3315-WB)
- Always two SiPM arrays look at the same scintillator (top and bottom)

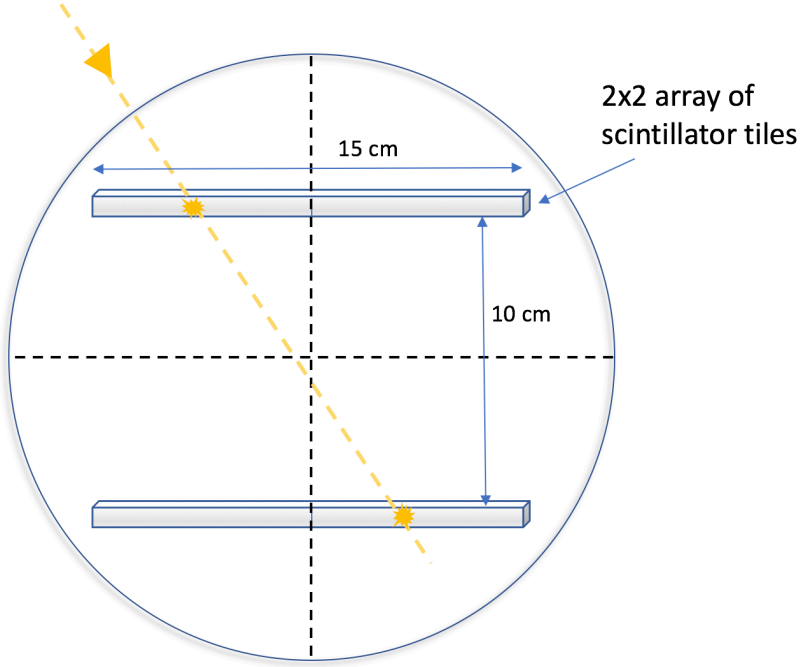


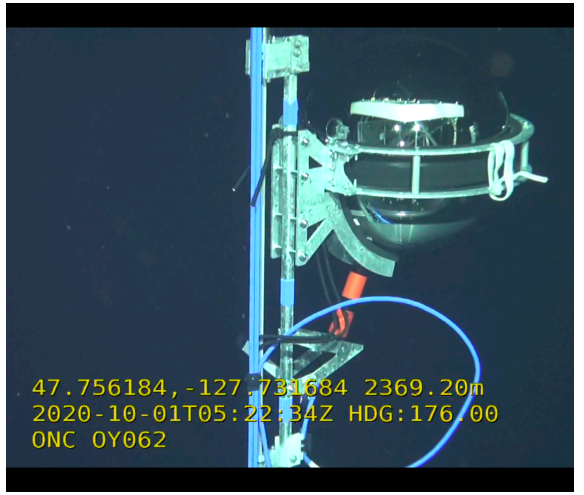




Working Principle

Muon measurements as part of the background characterisation	
Expected atm. muon rate at 2370 m depth	0.4 muons a day
Expected detection rate	0.1 muons a day
Coincidence	2 layers of scintillators
Direction reconstruction	Division of scintillators in tiles, dual ended read-out with SiPMs





Status

- Continuous data taking
- Work on analysing software not yet finished
- Time calibration with Lab Data

```
1 2262.3794487269 35.401
1 2262.3831380329 26.558
5 2262.3736325791 32.726
7 2262.3421926605 26.880
7 2262.3577049384 27.350
7 2262.3762404417 27.279
7 2262.3762585029 27.391
7 2262.3762759509 27.339
10 2262.3761187592 25.868
10 2262.3762404687 25.423
12 2262.3762759694 26.022
16 2262.3277469239 25.650
16 2262.3281664771 25.077
16 2262.3341658797 25.590
16 2262.3411692064 25.071
16 2262.3468078721 25.317
16 2262.3528152603 24.982
16 2262.3573402749 25.570
16 2262.3762759681 27.695
16 2262.4178283004 24.985
1 2262.4382460616 26.313
1 2262.4914961348 25.947
1 2262.6044894671 26.570
3 2262.5284287672 30.441
7 2262.6174919405 27.132
9 2262.5284287621 31.789
16 2262.5494200913 25.278
1 2262.6262444371 26.557
1 2262.6724230303 26.499
3 2262.6386621312 26.309
7 2262.6260872540 27.147
7 2262.6260942429 26.936
7 2262.6261012238 26.916
7 2262.7158969820 27.549
10 2262.6260872808 25.892
10 2262.6260942694 25.796
10 2262.6261012504 25.959
16 2262.6591109386 25.498
16 2262.6617812082 25.587
7 2262.7261507528 26.896
16 2262.8043556146 25.148
1 2262.8760526633 26.215
```

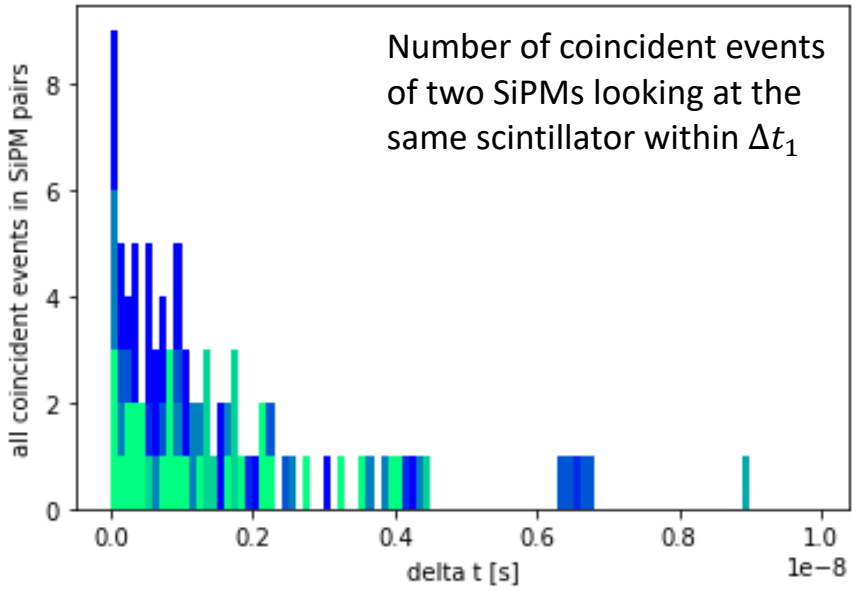
Analysis

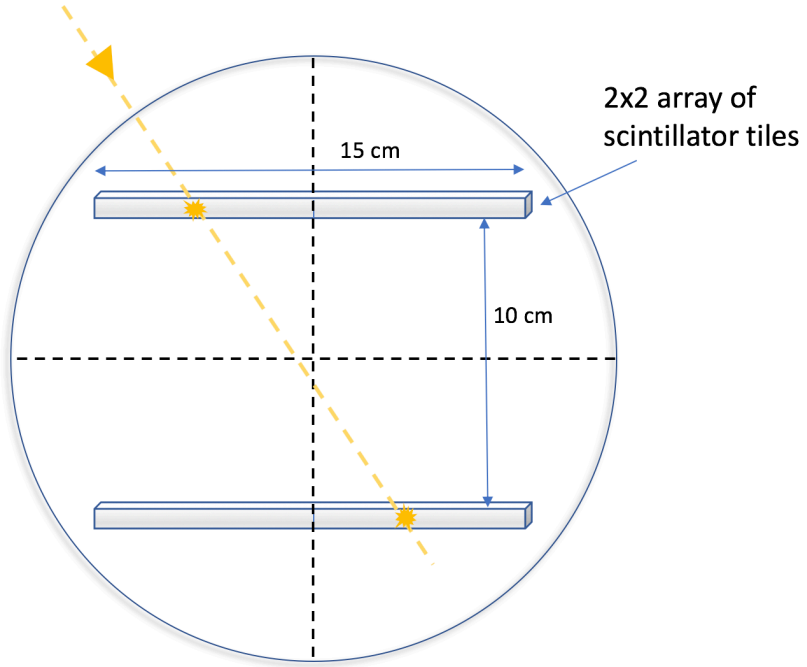
- Match coincident events in SiPM pairs within a certain time interval Δt_1
- Then match coincident events in top and bottom within Δt_2
- Problem: Which time intervals?

  → Calibration of lab data

```
top=[2,4,6,8,10,12,14,16]
bottom=[1,3,5,7,9,11,13,15]

arrayScintillatorDependence_TDC = {1:11,11:1,
                                   2:12,12:2,
                                   3:9,9:3,
                                   4:10,10:4,
                                   5:15,15:5,
                                   6:16,16:6,
                                   7:13,13:7,
                                   8:14,14:8}
```





First Pacific Ocean Muon found on 12/10/20

```
In [18]: print(down[0][0].ch, down[0][0].tS, down[0][0].tOT, down[0][0].coincidences)
12.0 40100.3424998723 35.572 [array([2.00000000e+00, 4.01003425e+04, 5.73050000e+01])]

In [19]: print(down[0][1].ch, down[0][1].tS, down[0][1].tOT, down[0][1].coincidences)
13.0 40100.3424998733 42.834 [array([7.00000000e+00, 4.01003425e+04, 4.45000000e+01])]
```

Outlook

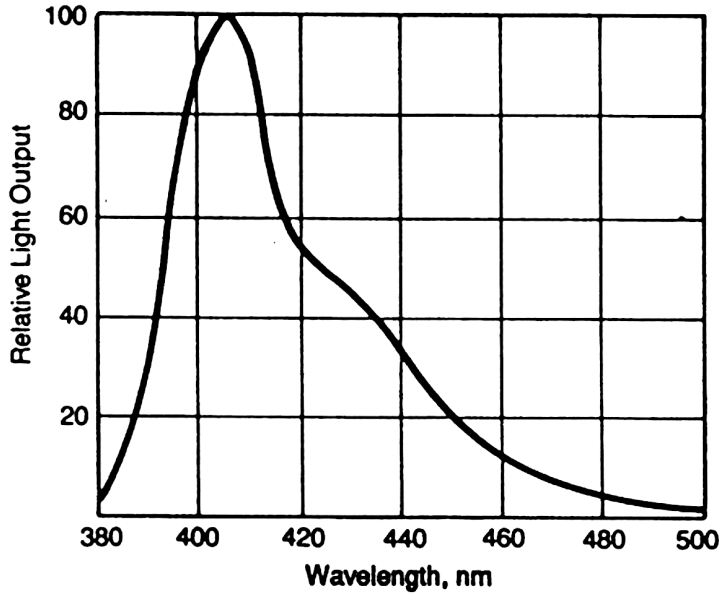
- Finish calibration
- Setting final threshold value (padiwa offset)
- Implementation in Sea Script
- Long-run continuous data taking
- Automatic analysing software

Thank you for your attention! Questions?

Scintillators

7.5x7.5x2.0 cm^3

Plastic scintillators (BC-404)



light output	68% of anthracene
rise time	0.7 ns
decay time	1.8 ns
pulse width, FWHM	2.2 ns
wavelength of max. emission	408 nm
light attenuation length	140 cm
bulk light attenuation length	160 cm

```
class pulse:
    def __init__(self, iD, channel, timeStamp, timeOverThreshold):
        self.iD = iD
        self.ch = channel
        self.tS = timeStamp
        self.tOT = timeOverThreshold
        self.coincidences = []

    def searchForCoincidence(self, eventListe, timeInterval):
        maximumTimeStamp = self.tS+timeInterval
        if eventListe:
            arrayEventList = np.array(eventListe)
            possibleCoincidences = arrayEventList[arrayEventList[:,1]<= maximumTimeStamp]
            for entry in possibleCoincidences:
                if entry[0] == arrayScintillatorDependence_TDC[self.ch]:
                    self.coincidences.append(entry)


def findCoincidences(TopBottomEventList):
    listOfCoincidences = []
    for i in range(len(TopBottomEventList)):
        pu = pulse(i,*TopBottomEventList.pop(0))
        pu.searchForCoincidence(TopBottomEventList,timeIntervalScintilator)
        if pu.coincidences:
            listOfCoincidences.append(pu)
    return listOfCoincidences

def crossCoin(liste1, liste2crossReference):
    crossCoins = []
    for i,pulse1 in enumerate(liste1):
        maxTimestamp = pulse1.tS+timeIntervalTopBottomCoincidence
        for pulse2 in liste2crossReference:
            if pulse1.tS <= pulse2.tS <= maxTimestamp:
                crossCoins.append((pulse1,pulse2))
    return crossCoins

def getMuonen(topList,bottomList):
    downgoing = crossCoin(topList, bottomList)
    upgoing = crossCoin(bottomList, topList)
    return downgoing, upgoing
```

```
topListsorted = []
bottomListsorted=[]

for entry in positiveSorted:
    if entry[0] in top:
        topListsorted.append(entry)
    elif entry[0] in bottom:
        bottomListsorted.append(entry)

coincidencesTop = findCoincidences(topListsorted)
coincidencesBottom = findCoincidences(bottomListsorted)

down, up = getMuonen(coincidencesTop, coincidencesBottom)
```