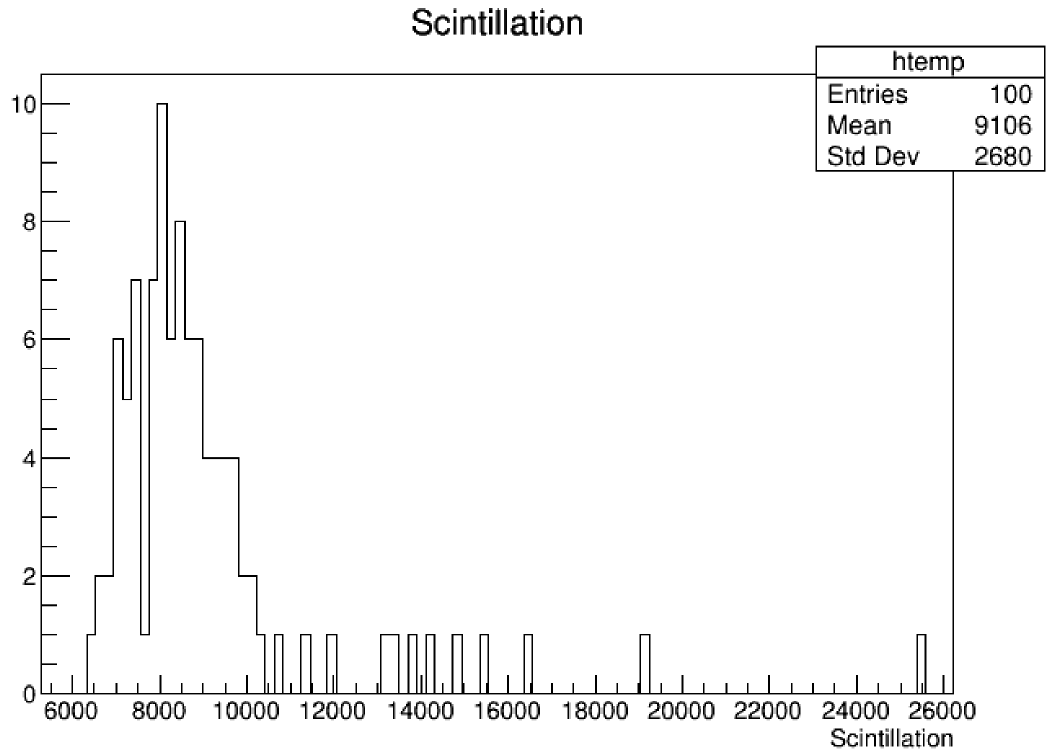
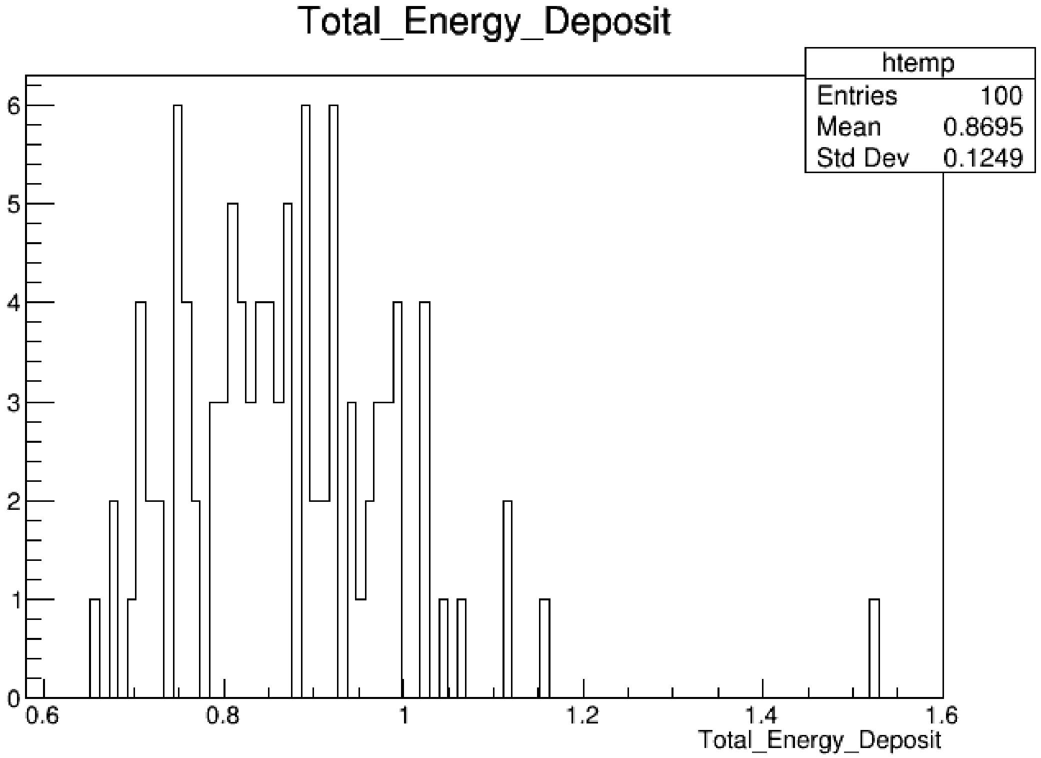
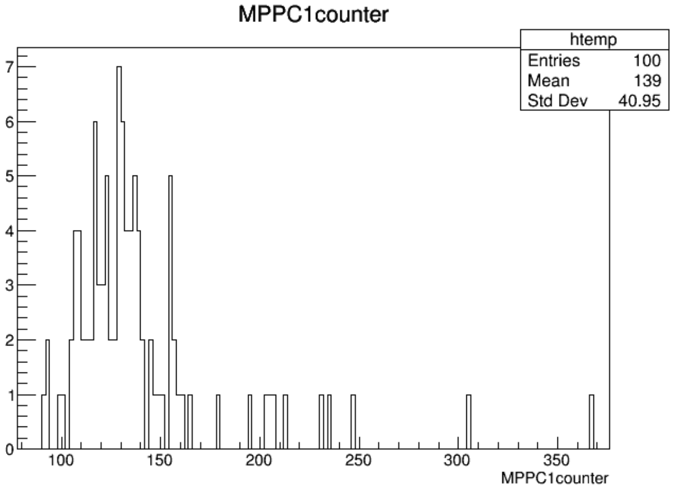
Position dependency

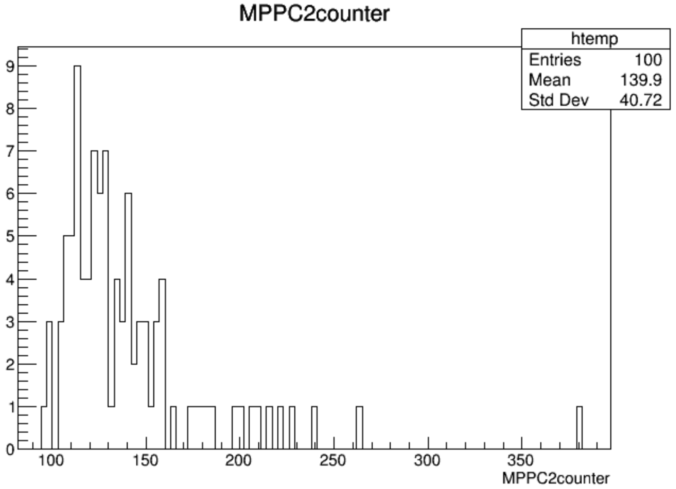
Energy deposit

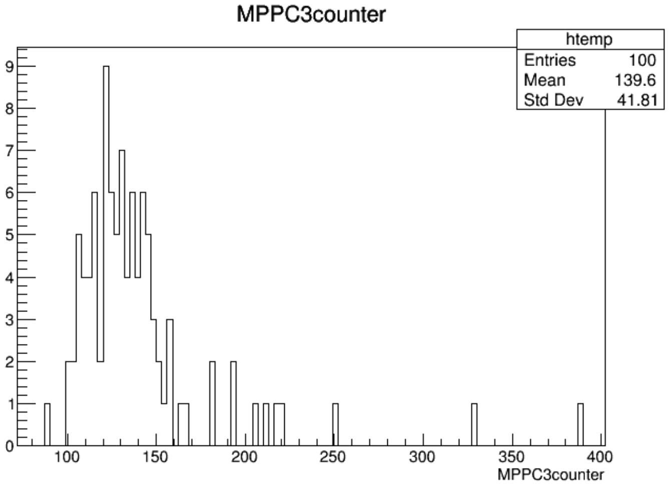


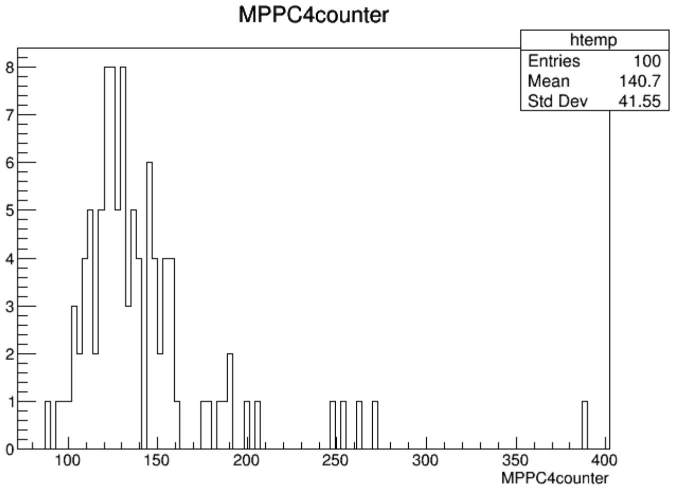


The number of photon arrived MPPC

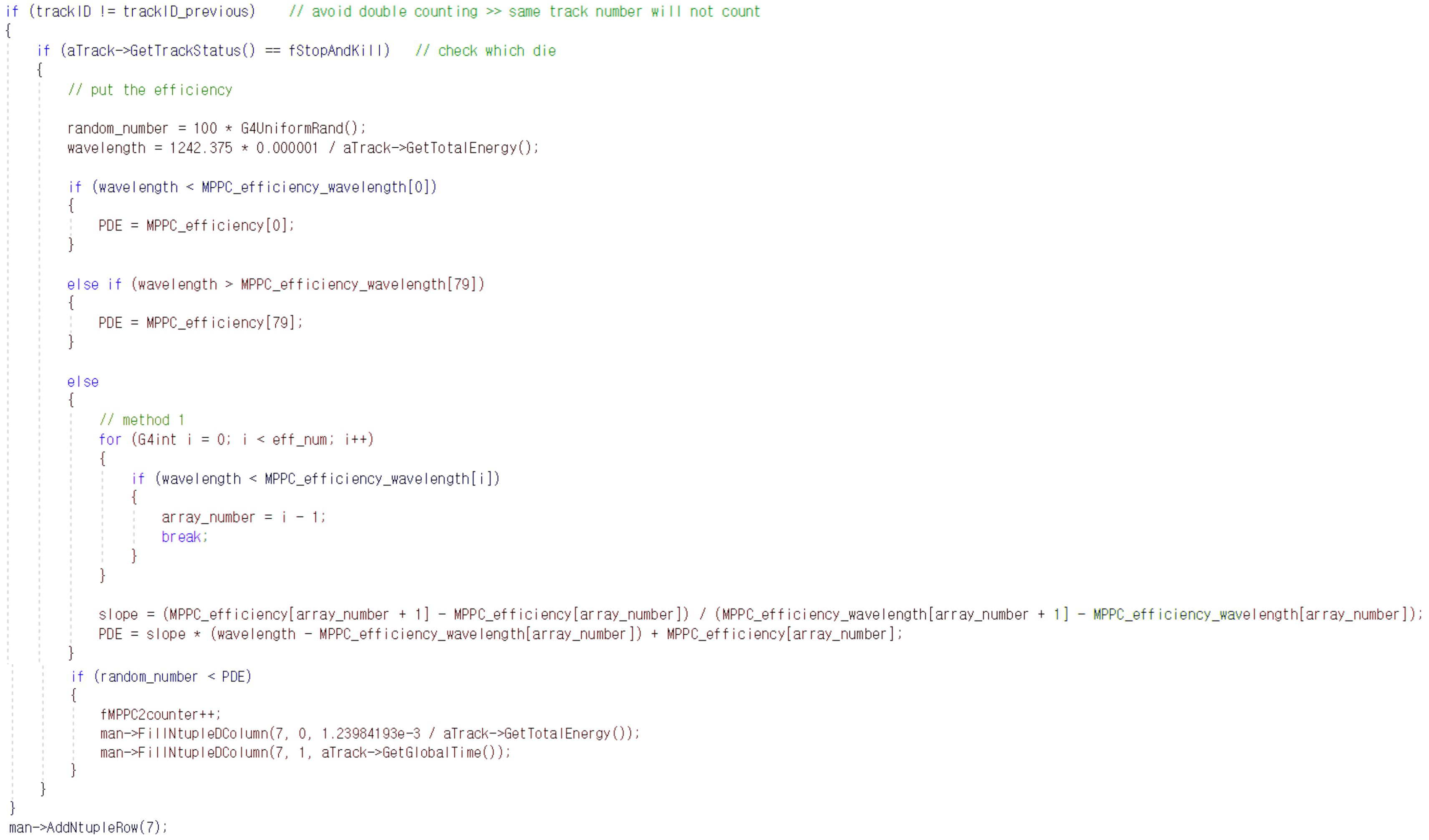






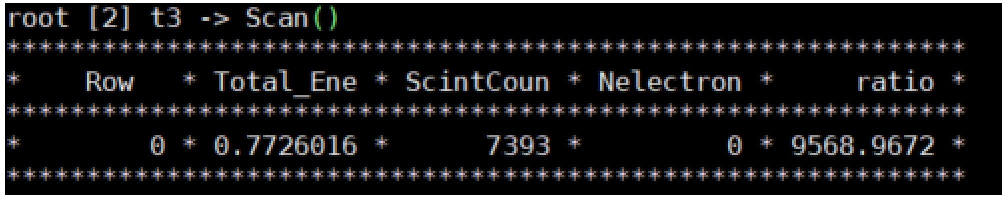


MPPC PDE

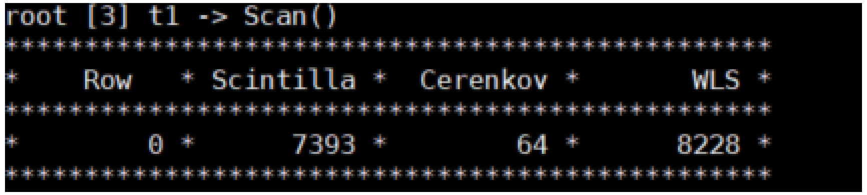


```
if (trackID != trackID_previous)    // avoid double counting >> same track number will not count
{
    if (aTrack->GetTrackStatus() == fStopAndKill)   // check which die
    {
        // put the efficiency

        random_number = 100 * G4UniformRand();
        wavelength = 1242.375 * 0.000001 / aTrack->GetTotalEnergy();

        if (wavelength < MPPC_efficiency_wavelength[0])
        {
            PDE = MPPC_efficiency[0];
        }

        else if (wavelength > MPPC_efficiency_wavelength[79])
        {
            PDE = MPPC_efficiency[79];
        }

        else
        {
            // method 1
            for (G4int i = 0; i < eff_num; i++)
            {
                if (wavelength < MPPC_efficiency_wavelength[i])
                {
                    array_number = i - 1;
                    break;
                }
            }

            slope = (MPPC_efficiency[array_number + 1] - MPPC_efficiency[array_number]) / (MPPC_efficiency_wavelength[array_number + 1] - MPPC_efficiency_wavelength[array_number]);
            PDE = slope * (wavelength - MPPC_efficiency_wavelength[array_number]) + MPPC_efficiency[array_number];
        }
        if (random_number < PDE)
        {
            fMPPC2counter++;
            man->FillNtupleDColumn(7, 0, 1.23984193e-3 / aTrack->GetTotalEnergy());
            man->FillNtupleDColumn(7, 1, aTrack->GetGlobalTime());
        }
    }
}
man->AddNtupleRow(7);
```

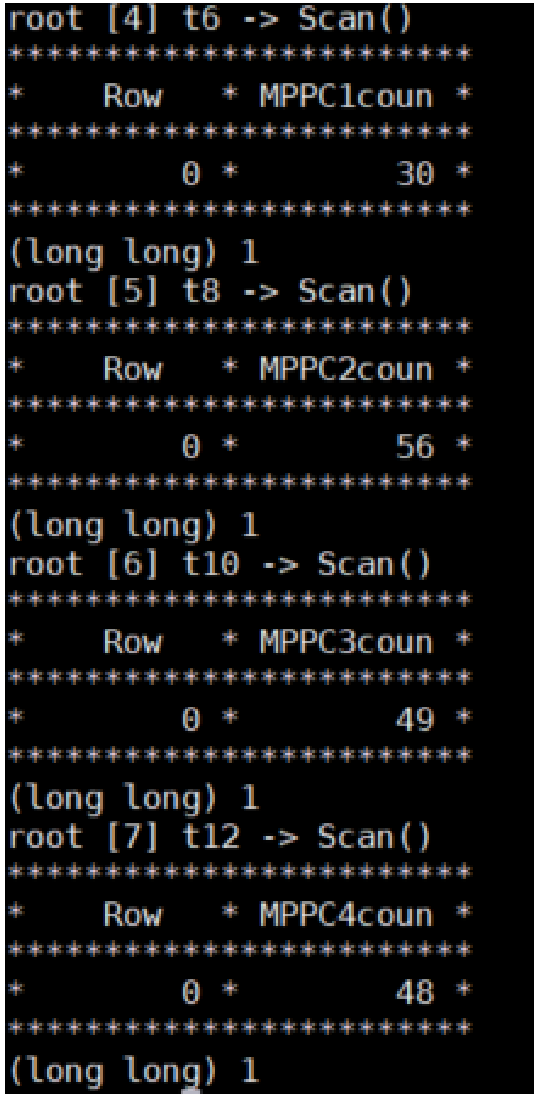
Energy deposit



```
root [2] t3 -> Scan()
************************************************************
*    Row   * Total_Ene * ScintCoun * Nelectron *     ratio *
************************************************************
*        0 * 0.7726016 *      7393 *         0 * 9568.9672 *
************************************************************
```

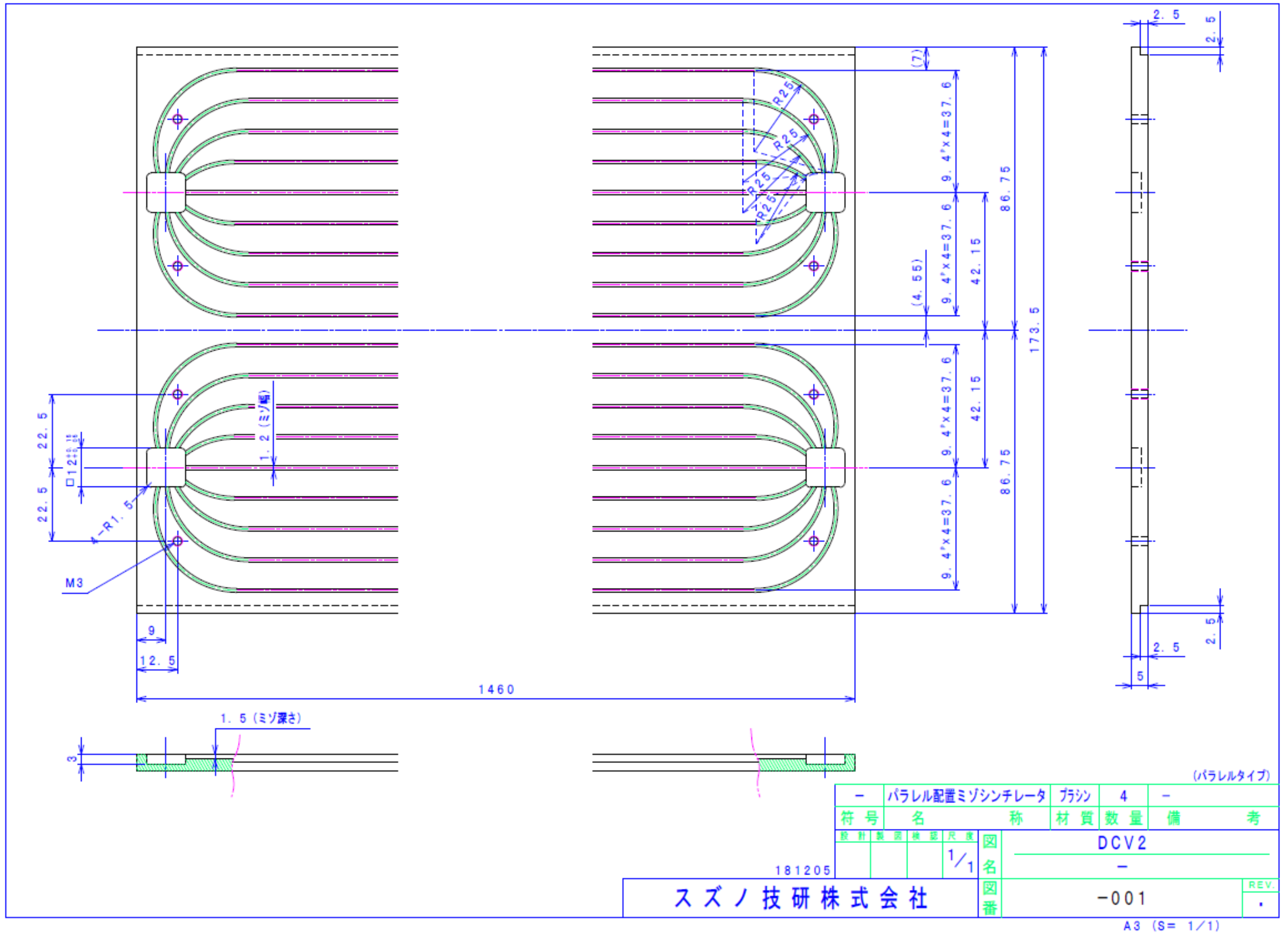


```
root [3] t1 -> Scan()
************************************************
*    Row   * Scintilla *  Cerenkov *       WLS *
************************************************
*        0 *      7393 *        64 *      8228 *
************************************************
```



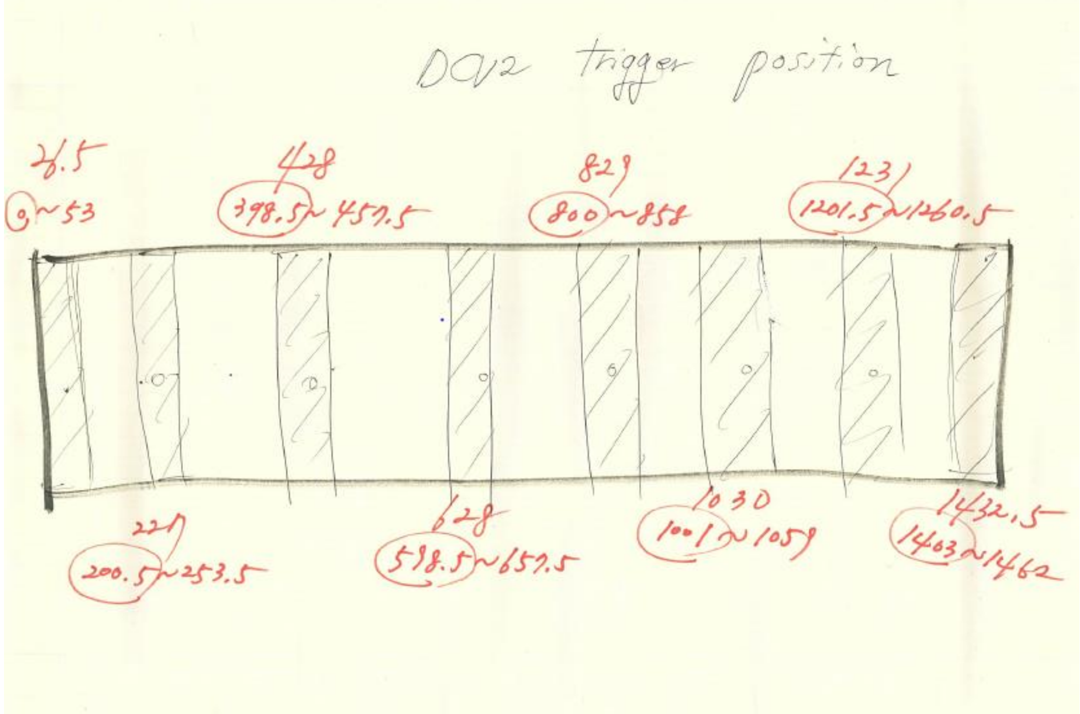
```
root [4] t6 -> Scan()
************************
*    Row   * MPPC1coun *
************************
*        0 *        30 *
************************
(long long) 1
root [5] t8 -> Scan()
************************
*    Row   * MPPC2coun *
************************
*        0 *        56 *
************************
(long long) 1
root [6] t10 -> Scan()
************************
*    Row   * MPPC3coun *
************************
*        0 *        49 *
************************
(long long) 1
root [7] t12 -> Scan()
************************
*    Row   * MPPC4coun *
************************
*        0 *        48 *
************************
(long long) 1
```

Blue print of DCV2



- Plastic scintillator for DCV2 is 1410 mm long with a cross section 5 mm X 171.5 mm.
- Each scintillator has 18 grooves for wavelength shifting fibers, which are guided to aluminum light collection boxes at both ends.

Checking point



- Checking point for position dependency of DCV is 8 point.
- According to DCV log note, each of checking width is written below.

0 ~ 53 mm, 200.5 ~ 253.5 mm,
398.5 ~ 457.5 mm, 598.5 ~ 657.5 mm,
800 ~ 858 mm, 1001 ~ 1059 mm,
1201.5 ~ 1260.5 mm, 1403 ~ 1462 mm

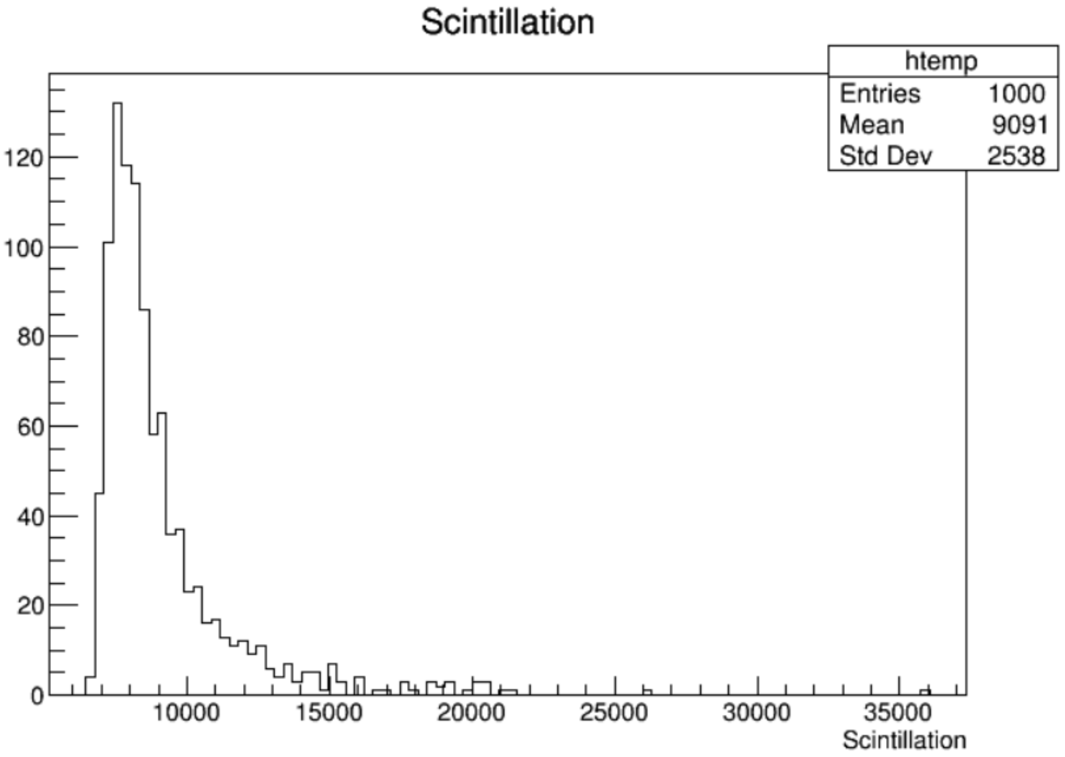
Step1

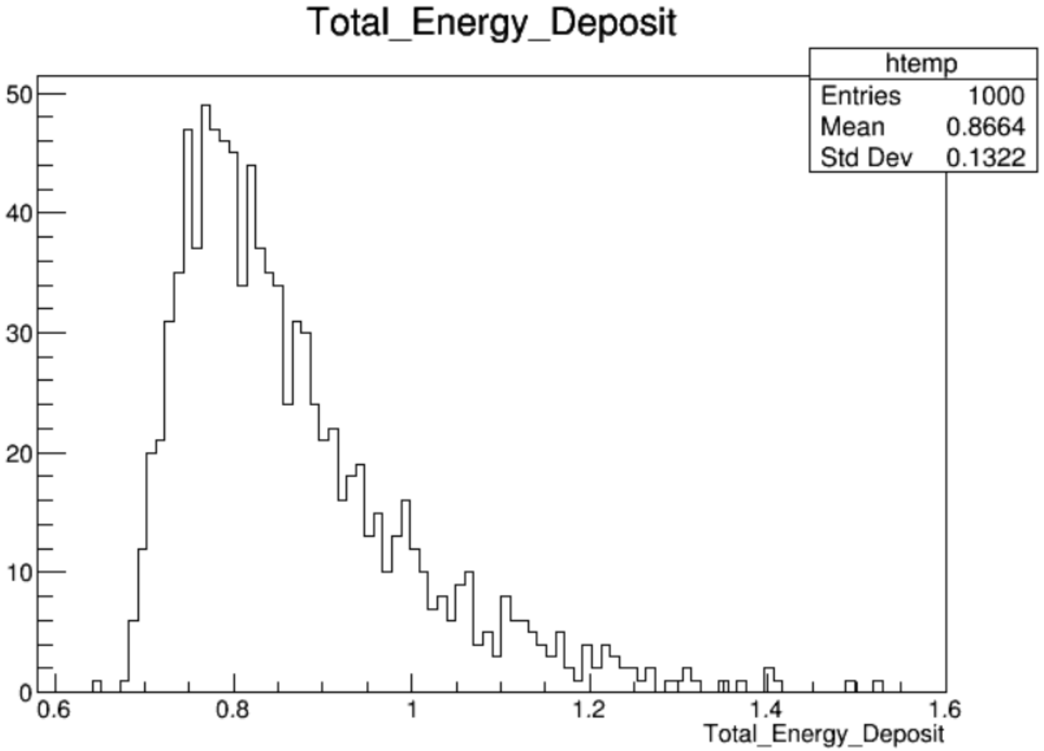
- Let assume that the point where the cosmic ray(muon+) fell is in the center of trigger area.
- In other words, the point where each cosmic ray falls is as follows.

26.5 mm, 227 mm, 428 mm, 628 mm, 829 mm, 1030 mm, 1231 mm, 1432 mm

- All y-axis position is 86.75 mm.

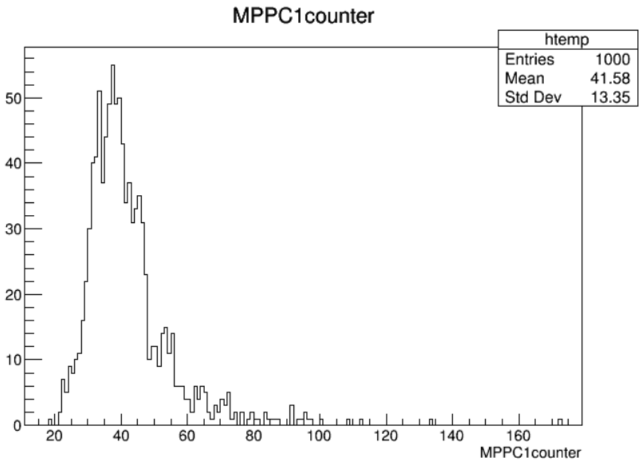
Results – 26.5 mm

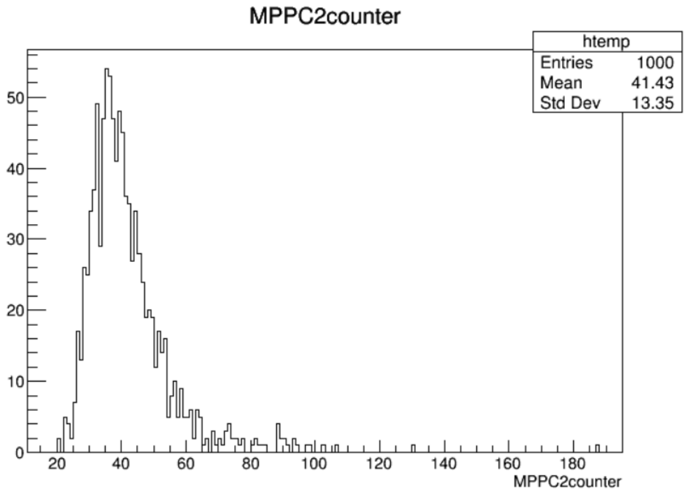


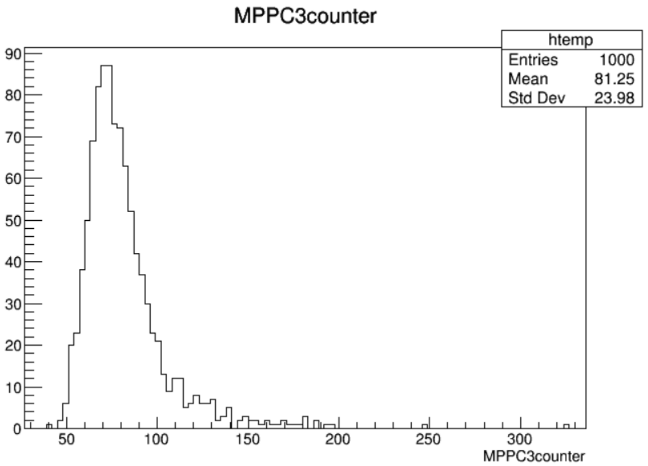


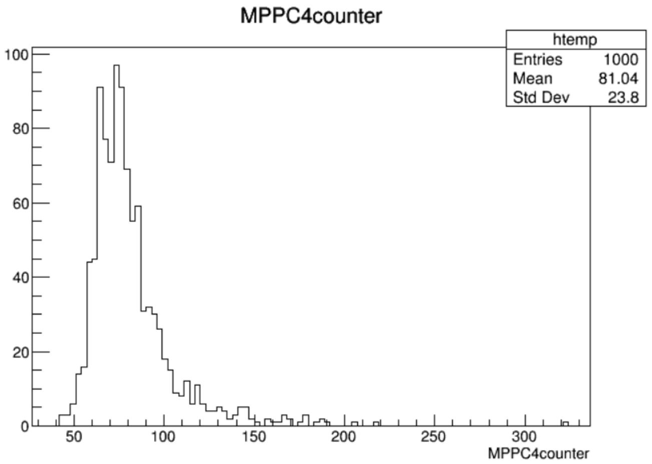
- The reason that there is difference between energy deposit and the number of scintillation photon is total energy deposit is collect energy deposit by muon+

The number of photons arrived MPPC

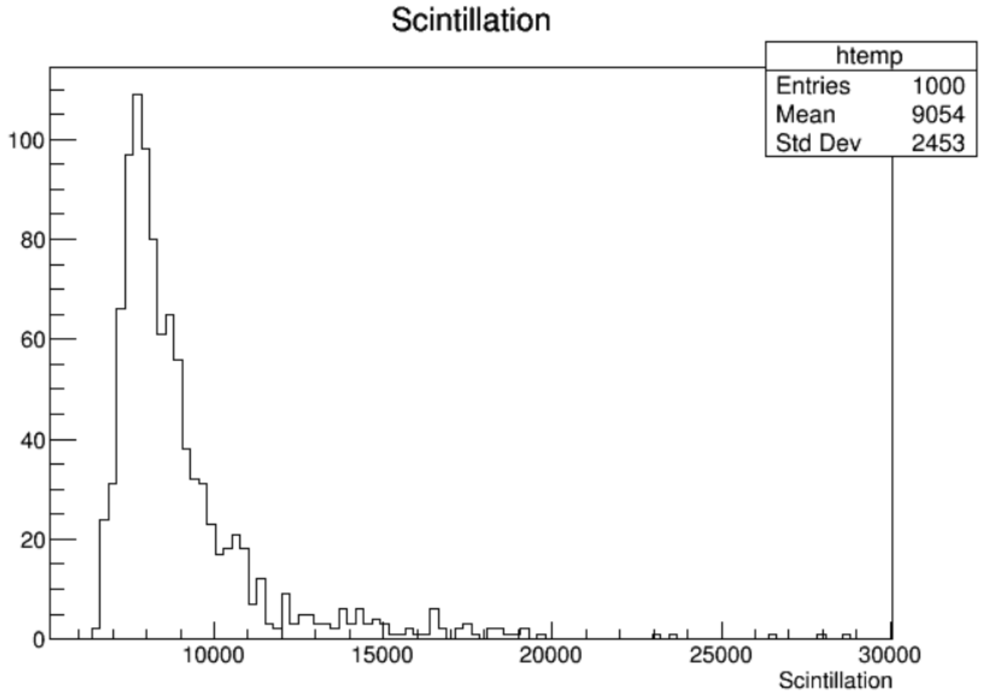


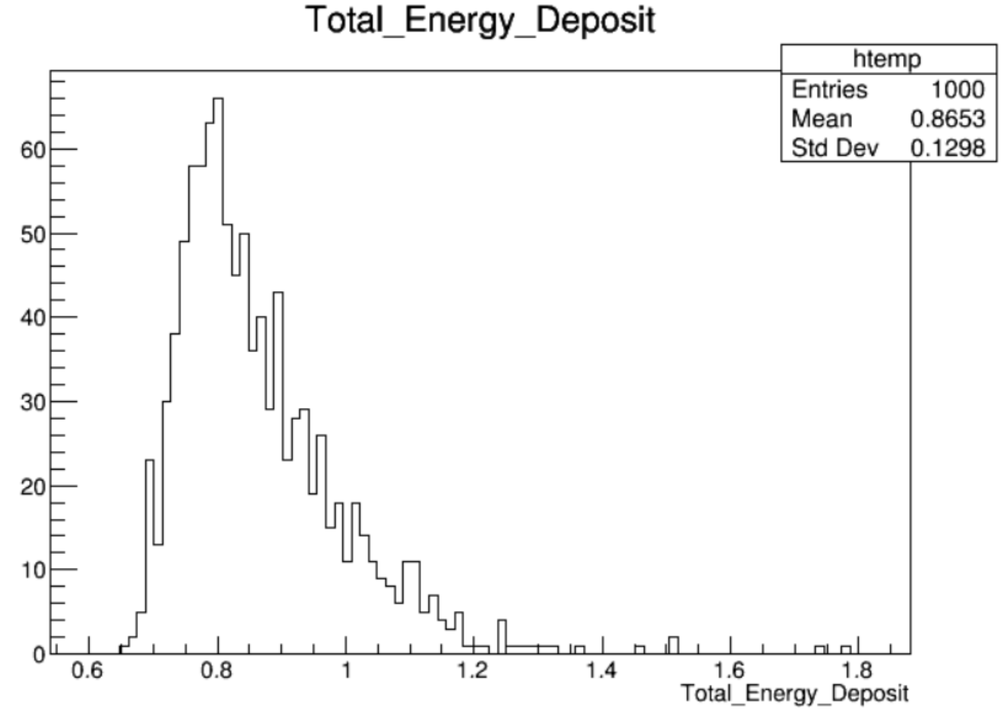




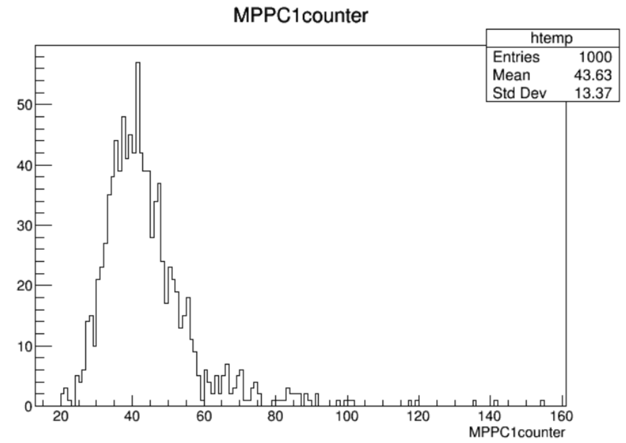


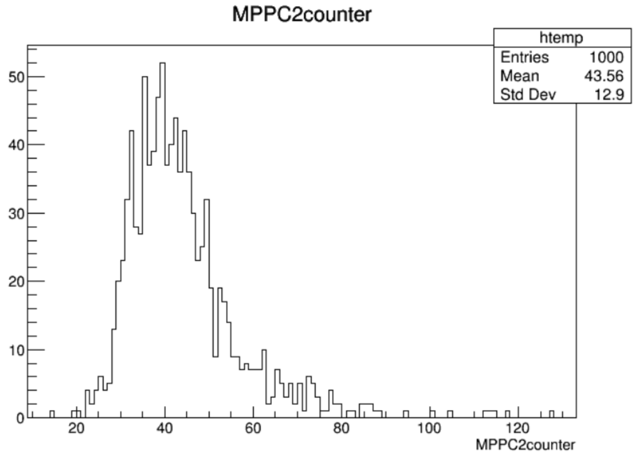
Results – 227 mm

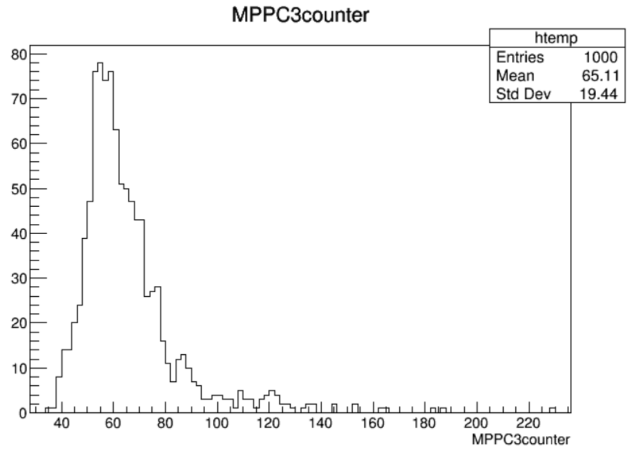


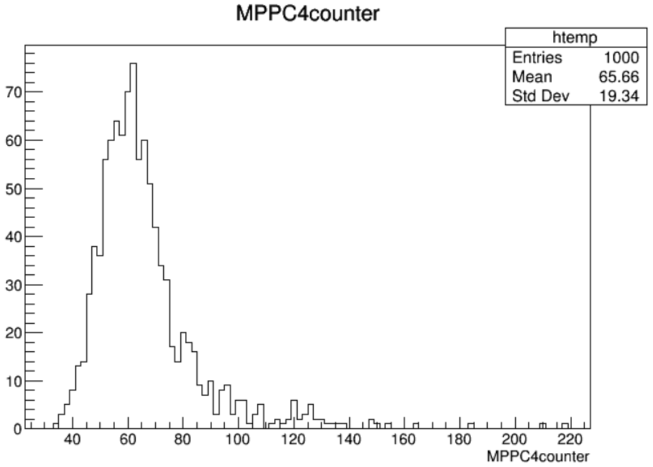


The number of photons arrived MPPC

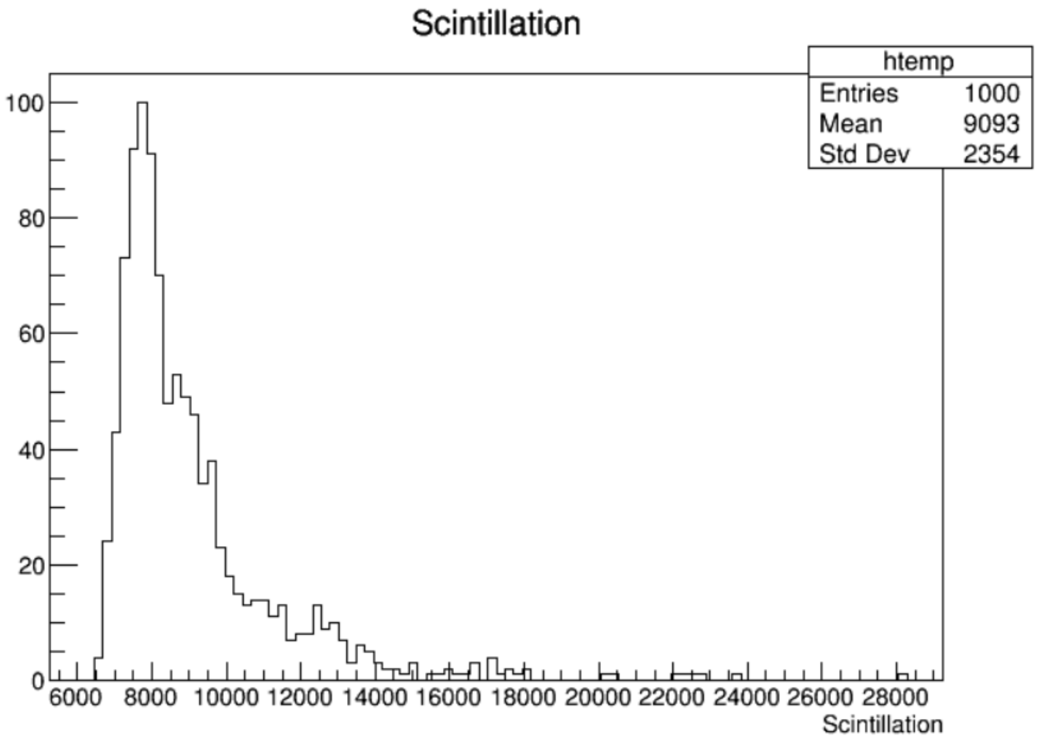


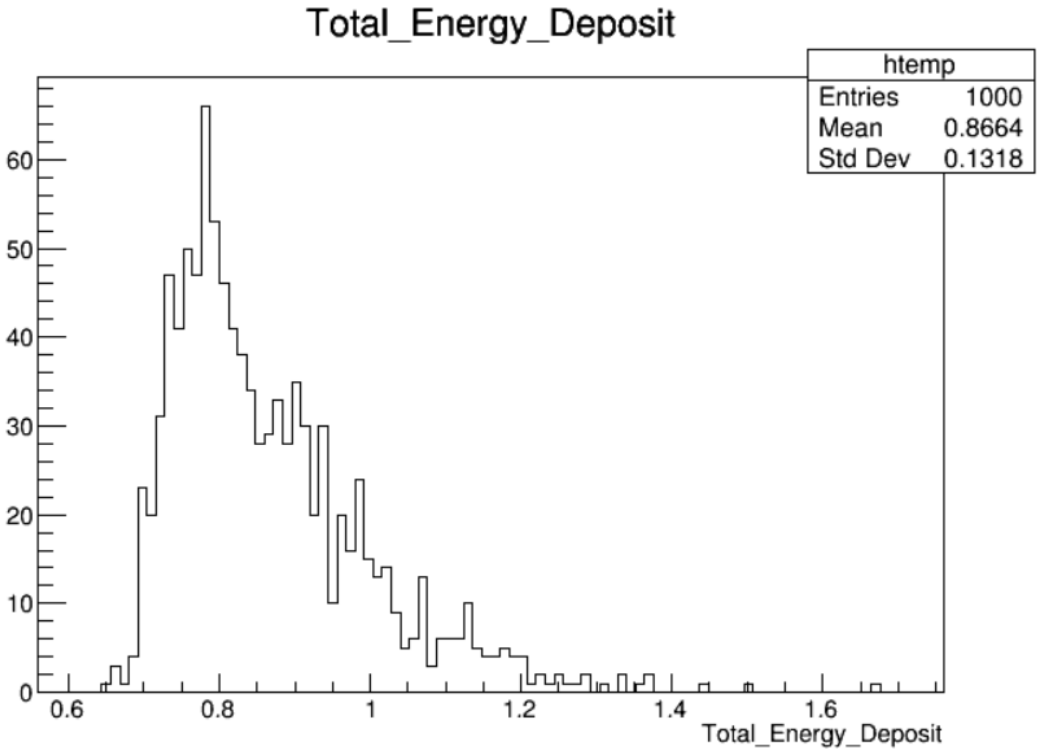




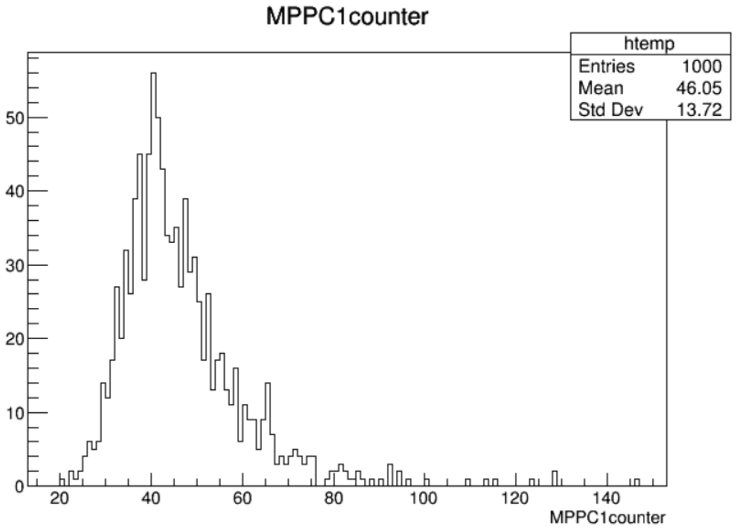


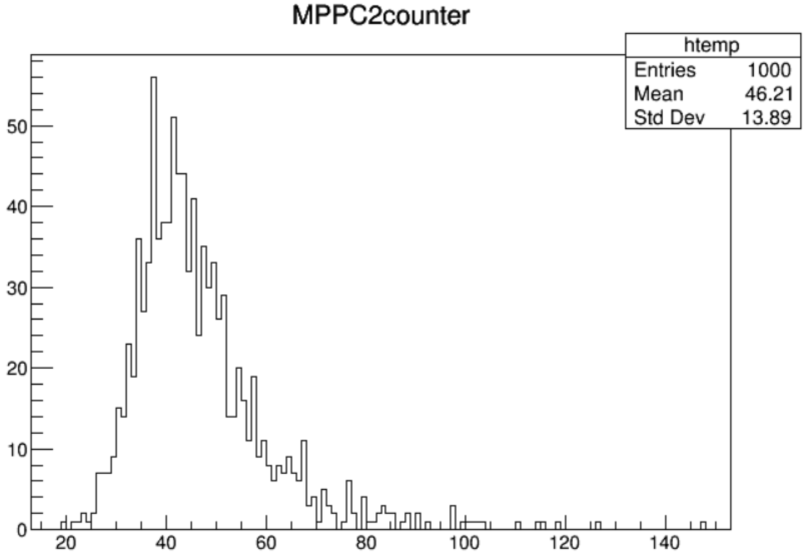
Results – 428 mm

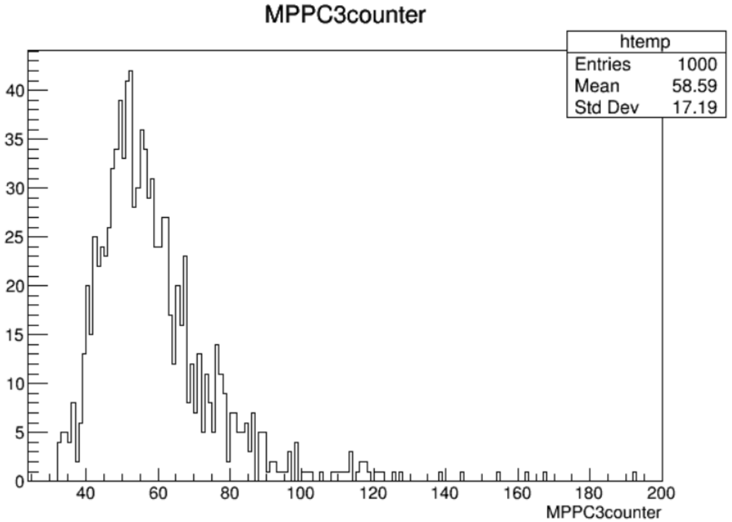


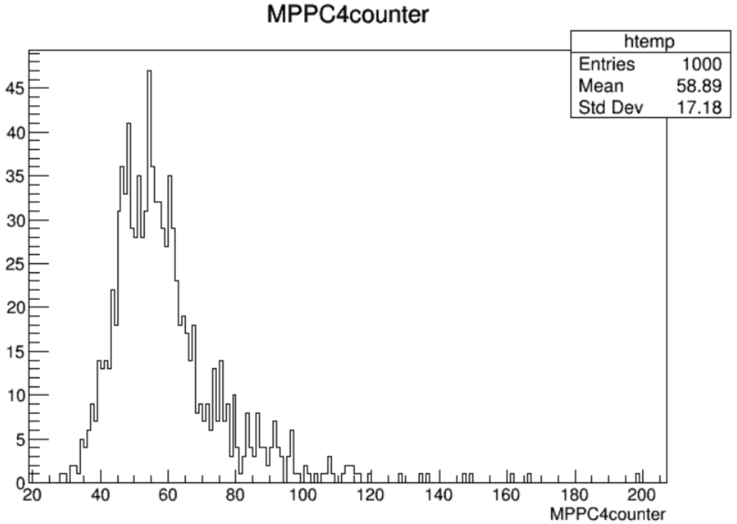


The number of photons arrived MPPC

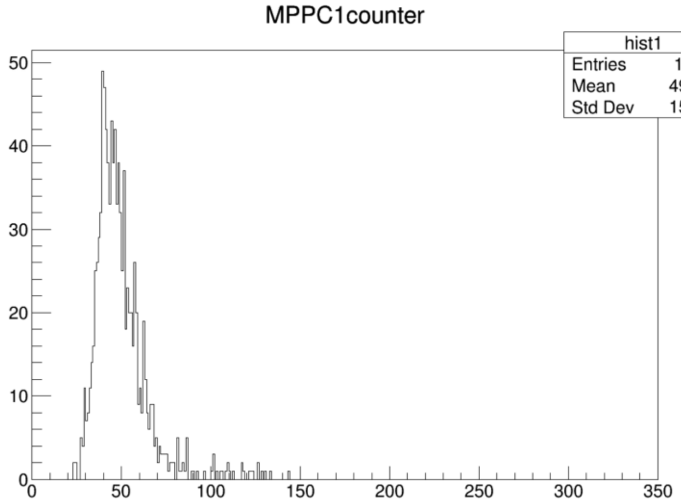


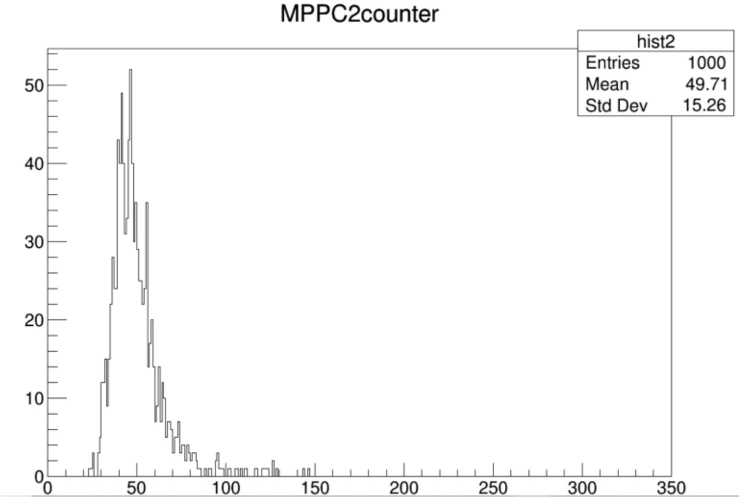


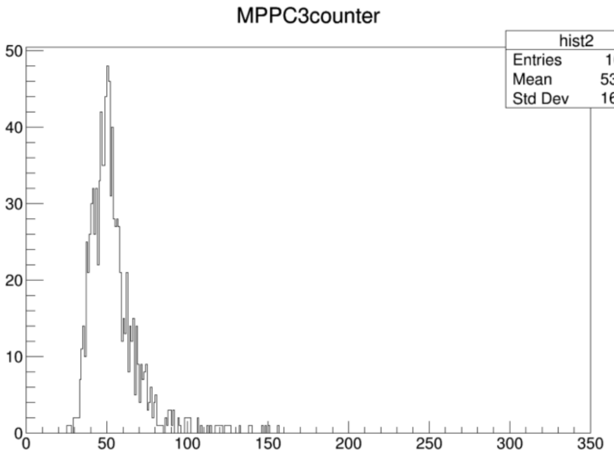


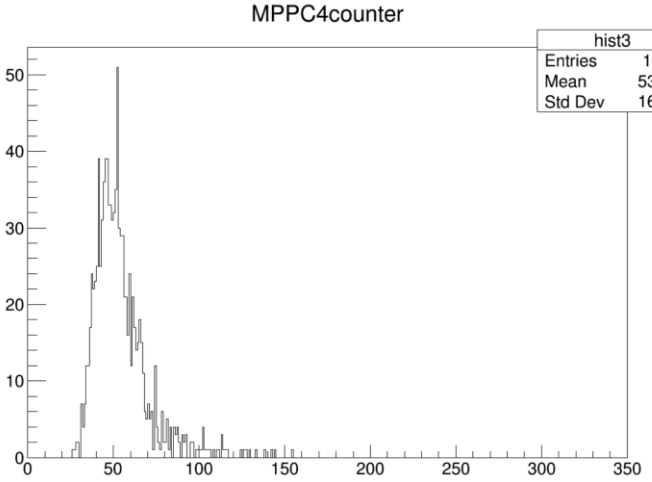


The number of photons arrived MPPC – 628 mm

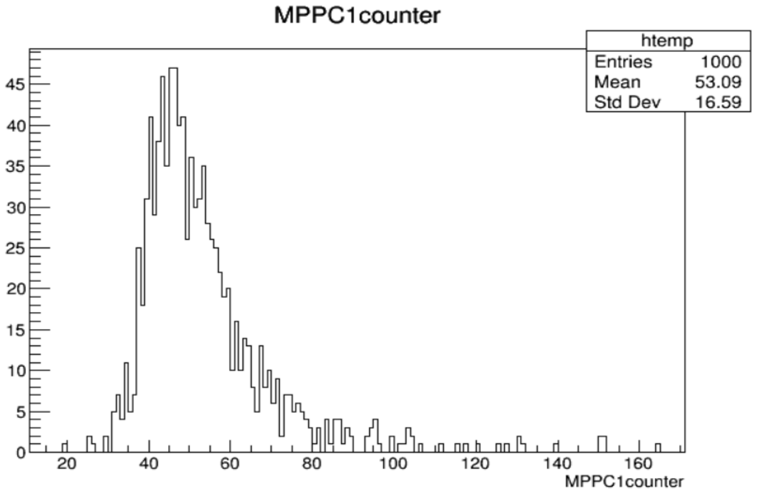


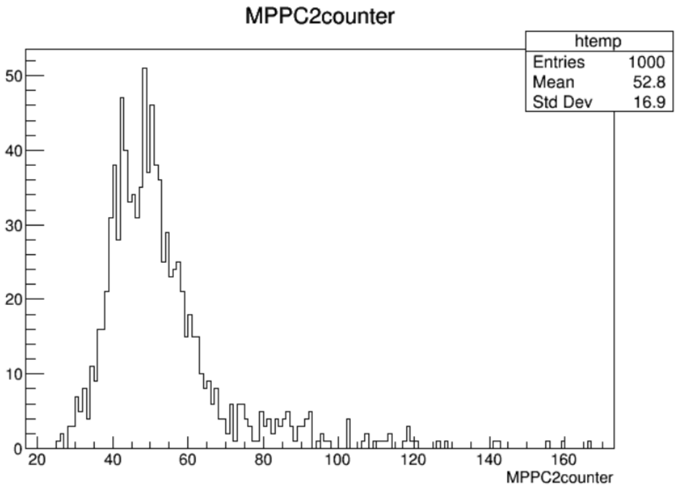


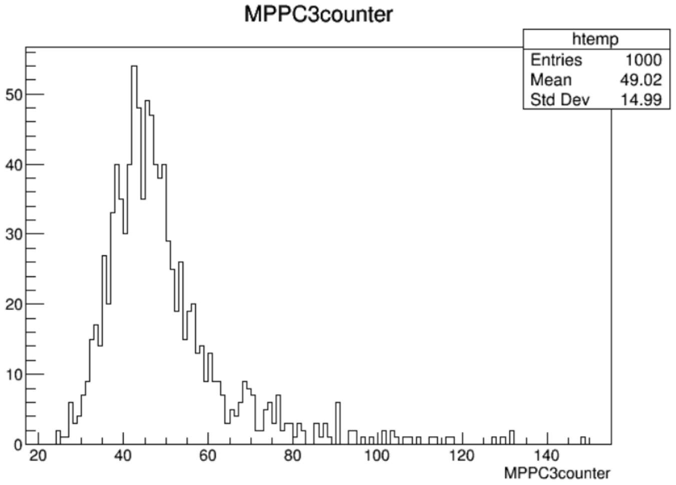


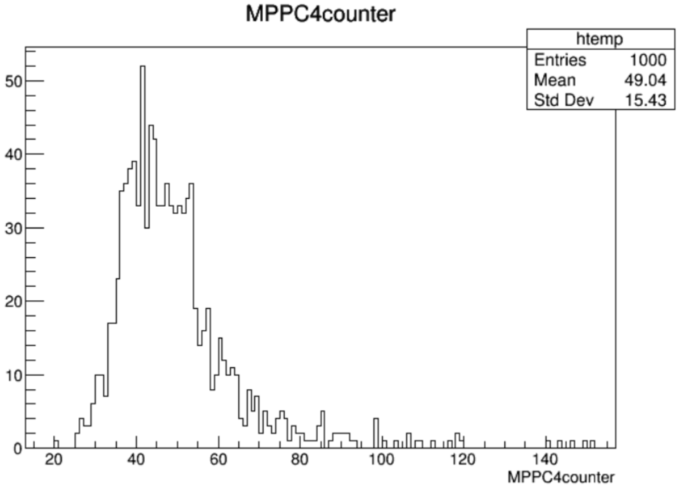


The number of photons arrived MPPC – 829 mm

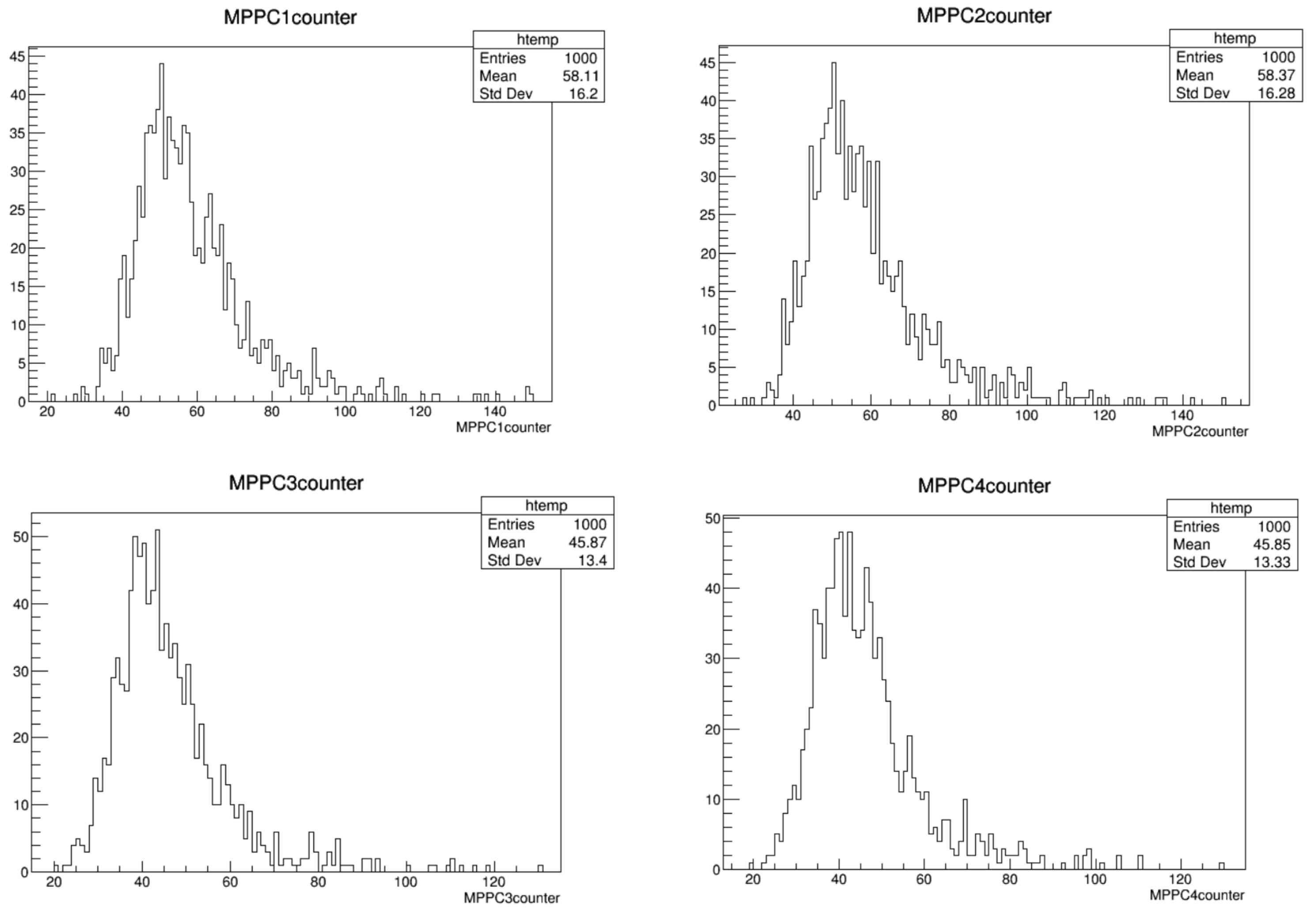




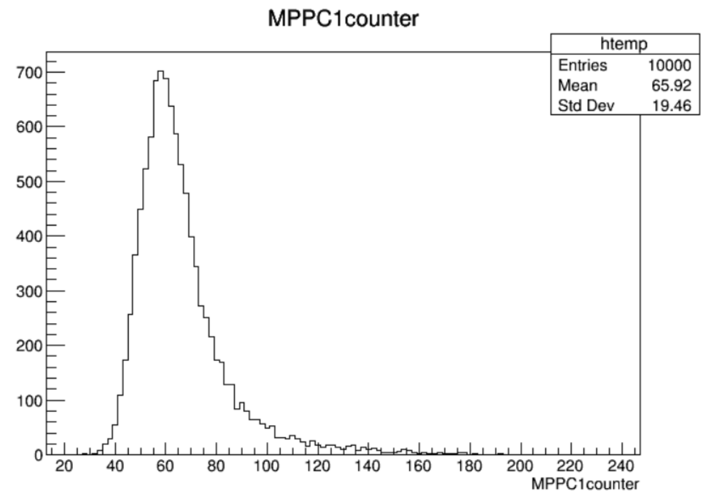


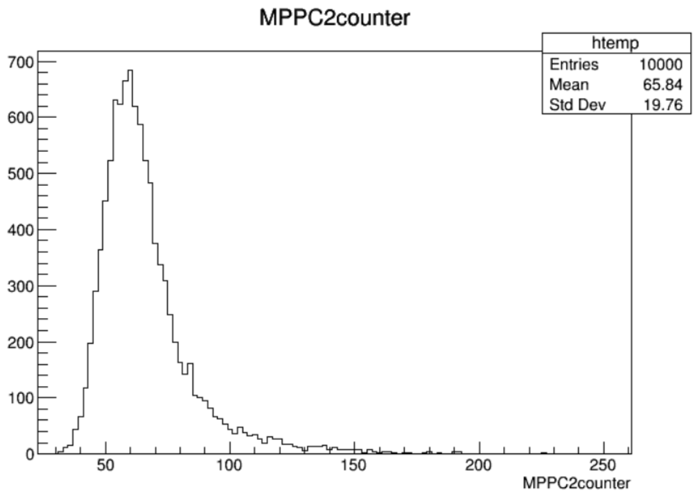


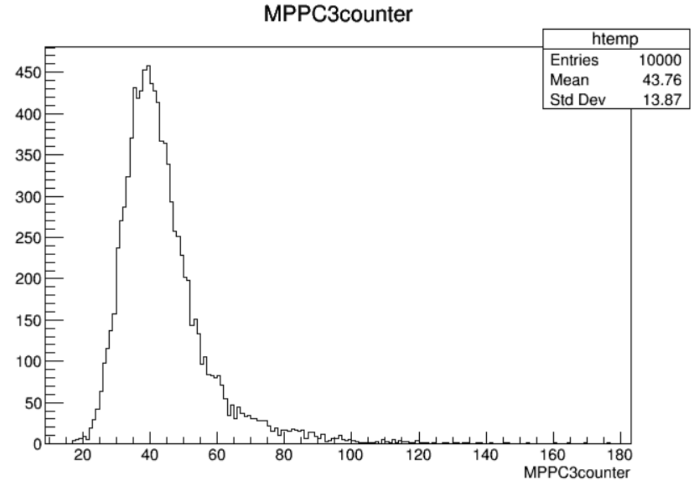
The number of photons arrived MPPC – 1030 mm

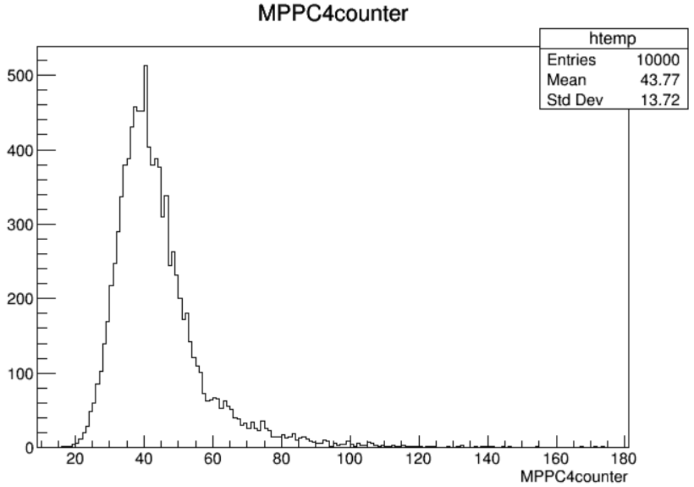


The number of photons arrived MPPC – 1231 mm

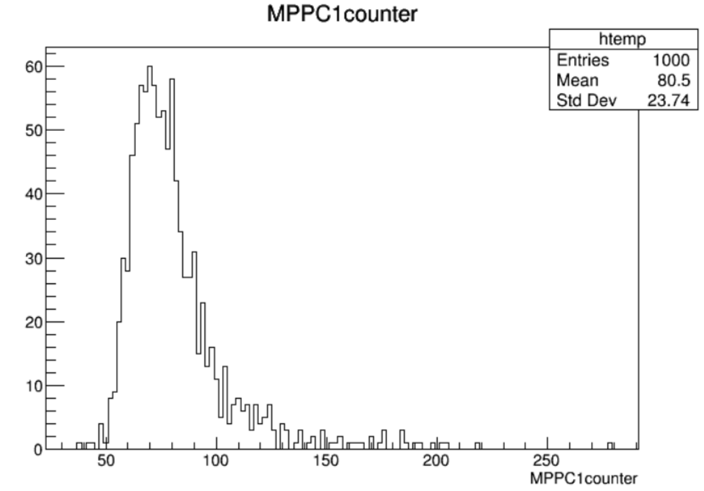


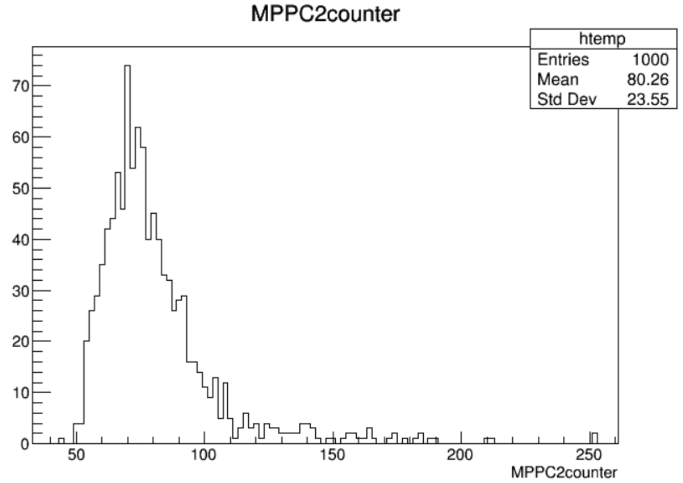


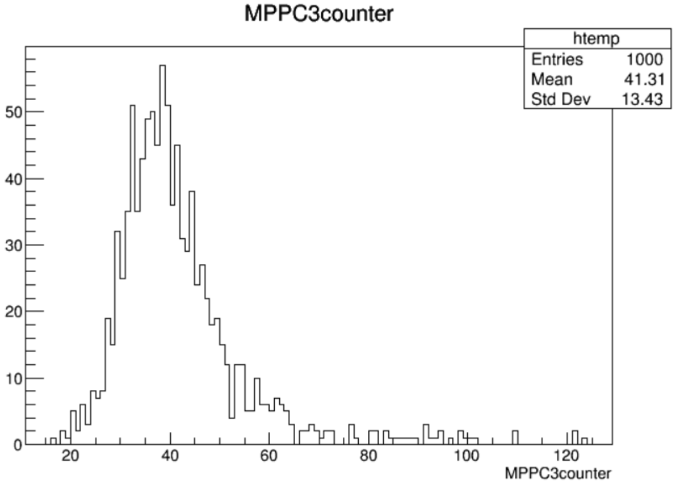


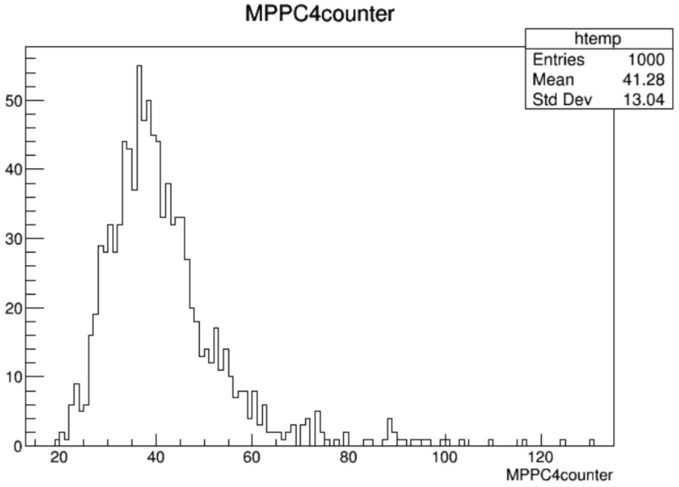


The number of photons arrived MPPC – 1432 mm





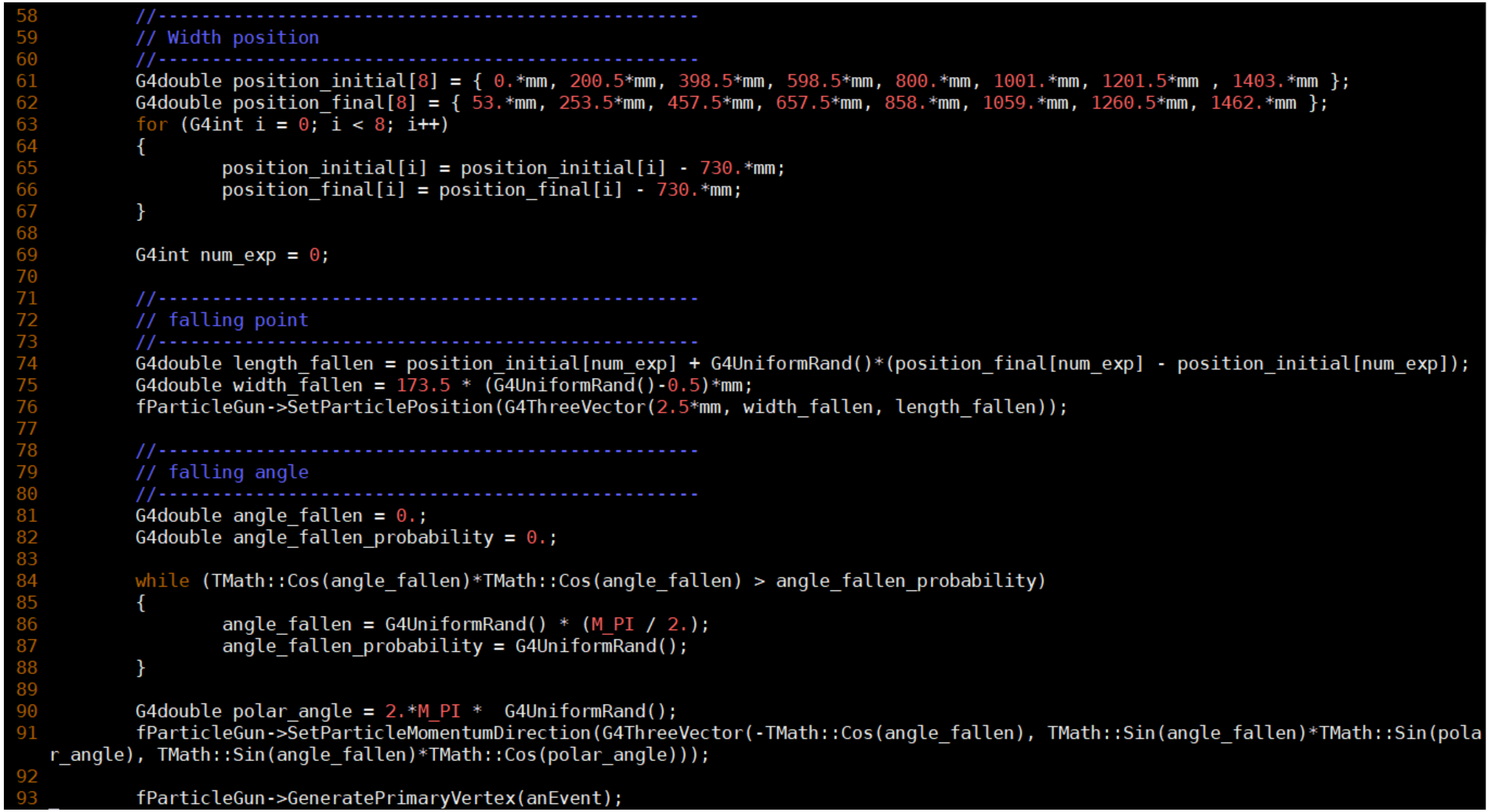




Future plan

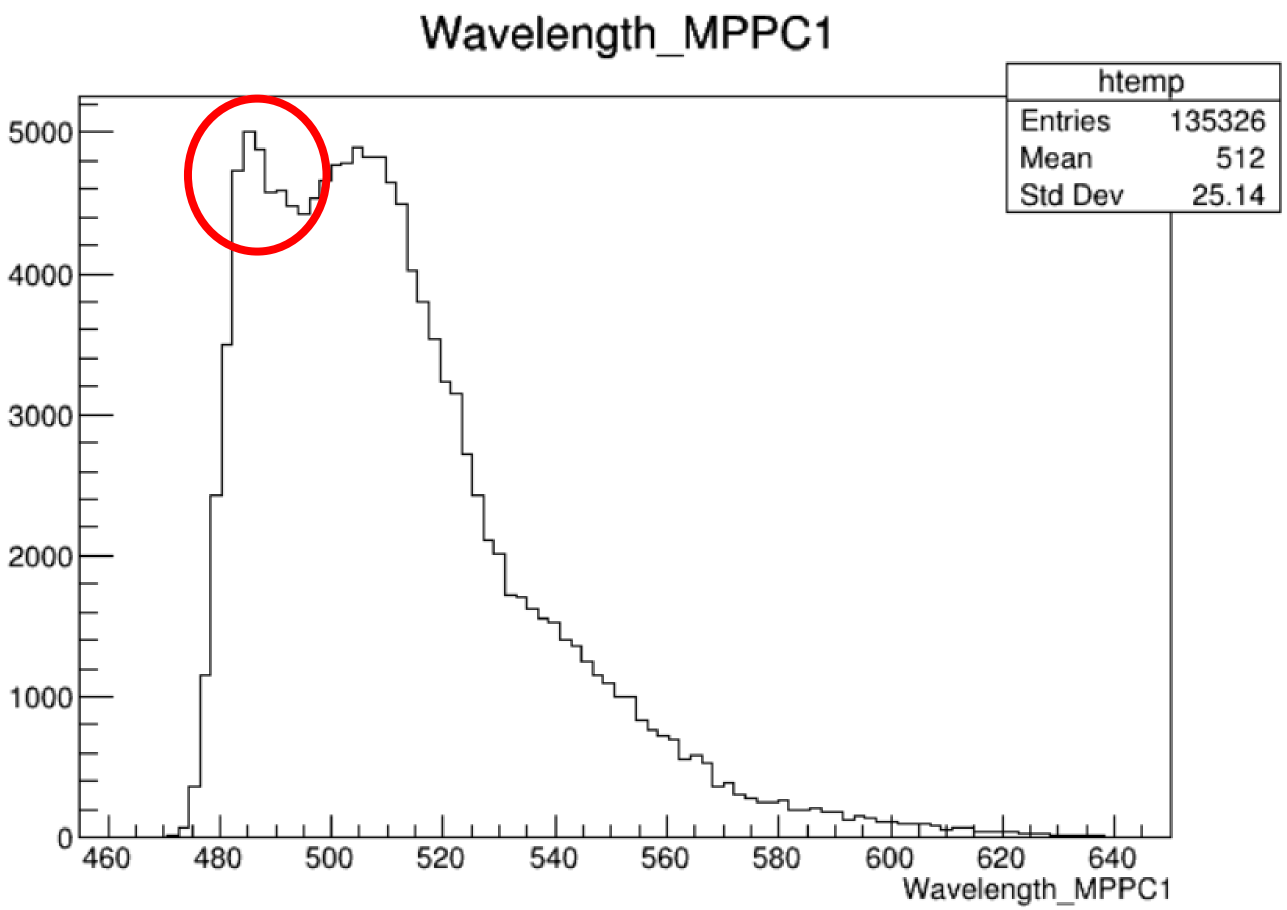
- Because the cosmic ray falls across all areas, next step is to make falling position random.
- Finally, the angle at which the cosmic ray falls is proportional to the square of the cosine. Therefore the angle of incidence is randomized. And the results will be compared with the actual results.

Code

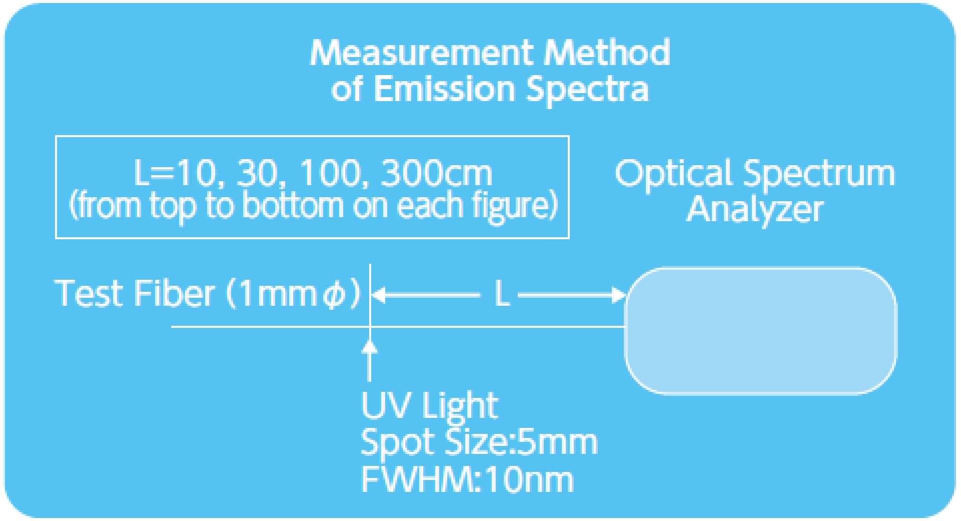


```
        //--------------------------------------------------
        // Width position
        //--------------------------------------------------
        G4double position_initial[8] = { 0.*mm, 200.5*mm, 398.5*mm, 598.5*mm, 800.*mm, 1001.*mm, 1201.5*mm , 1403.*mm };
        G4double position_final[8] = { 53.*mm, 253.5*mm, 457.5*mm, 657.5*mm, 858.*mm, 1059.*mm, 1260.5*mm, 1462.*mm };
        for (G4int i = 0; i < 8; i++)
        {
                position_initial[i] = position_initial[i] - 730.*mm;
                position_final[i] = position_final[i] - 730.*mm;
        }

        G4int num_exp = 0;

        //--------------------------------------------------
        // falling point
        //--------------------------------------------------
        G4double length_fallen = position_initial[num_exp] + G4UniformRand()*(position_final[num_exp] - position_initial[num_exp]);
        G4double width_fallen = 173.5 * (G4UniformRand()-0.5)*mm;
        fParticleGun->SetParticlePosition(G4ThreeVector(2.5*mm, width_fallen, length_fallen));

        //--------------------------------------------------
        // falling angle
        //--------------------------------------------------
        G4double angle_fallen = 0.;
        G4double angle_fallen_probability = 0.;

        while (TMath::Cos(angle_fallen)*TMath::Cos(angle_fallen) > angle_fallen_probability)
        {
                angle_fallen = G4UniformRand() * (M_PI / 2.);
                angle_fallen_probability = G4UniformRand();
        }

        G4double polar_angle = 2.*M_PI *  G4UniformRand();
        fParticleGun->SetParticleMomentumDirection(G4ThreeVector(-TMath::Cos(angle_fallen), TMath::Sin(angle_fallen)*TMath::Sin(polar_angle), TMath::Sin(angle_fallen)*TMath::Cos(polar_angle)));

        fParticleGun->GeneratePrimaryVertex(anEvent);
```

Problem


Wavelength_MPPC1
htemp
Entries 135326
Mean 512
Std Dev 25.14
5000
4000
3000
2000
1000
0
460
480
500
520
540
560
580
600
620
640
Wavelength_MPPC1

Comparison of simulation



Measurement Method
of Emission Spectra

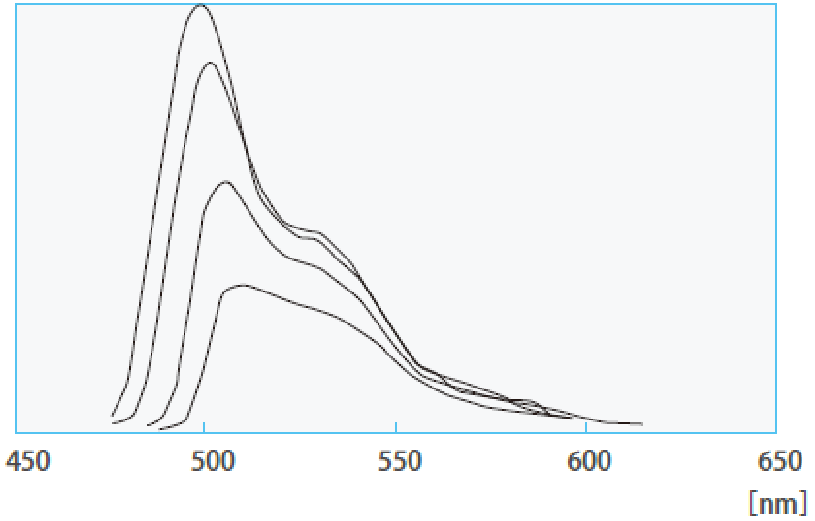
L=10, 30, 100, 300cm
(from top to bottom on each figure)

Optical Spectrum
Analyzer

Test Fiber (1mmϕ) ← L →

UV Light
Spot Size:5mm
FWHM:10nm

Y-11(200)



Exiting Wavelength:430nm

When photon of a wavelength of 430nm shoot in the end of fiber, the emission spectrum should be same as the right.

Absorbance

- Mathematically, probability of finding a particle at depth x into the material in calculated by Beer-Lambert Law

$$P(x) = e^{-x/\lambda}$$

- And λ is attenuation(absorption) length, and it depend on material and energy.
- Definition of absorbance is as follow.

$$\text{ABS} = k(\lambda)Cd = log_{10}\{\frac{I_0(\lambda)}{I(\lambda)}\} \qquad \text{when d = 10 mm}$$

- For reference, C is equal to 200 ppm and k_p(k at peak of abosorption) is equal to 0.00638 in Y-11 of Kurarary

Changing parameter used in Geant4

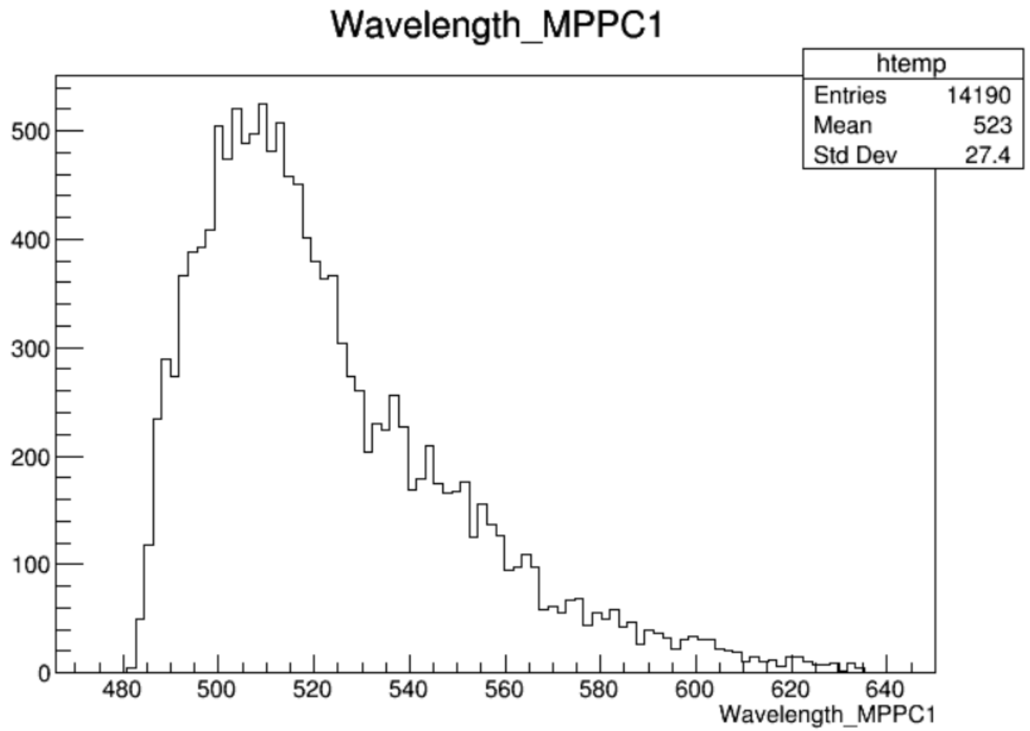
- If assume that y-axis of absorption spectrum is k, absorption(attenuation) length of Y-11 is as follow.

$$\lambda = \frac{10}{kC * ln10}$$

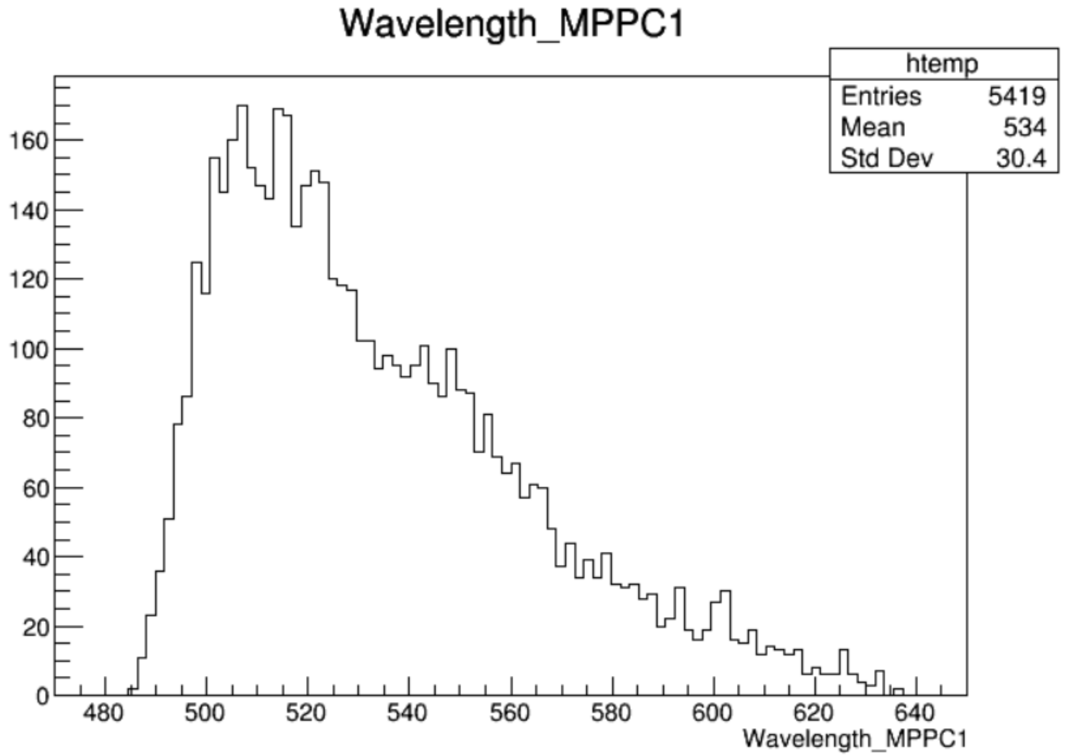
- C is concentration of dyne used in Y-11 and k is constant which is function of wavelength.

Comparison with reference

For 10 cm



For 30 cm



확인사항

- Photon 발생 제거했을 때 제대로 deposit 되는지 살펴보고 돌리기