Instruction to PMT Analysis with pulsed LED

- first of all you have to do the **measurements** of course, so you put LED in opposite of PMT and trigger on LED, save the PMT voltage over time values (best binary format) to hard drive (best name it Run00XXX AND FILL OUT THE RUNSHEET PROTOCOLL!!!)
- **copy** them to **hpc_data** (ftp server from ecap, you find it in finder at „All/hpc_data“, type your password from mac login) for storage and **copy folder with data to workspace** where you would like to make the further analysis (best hpc)
- the initial step for analysis is the generation of a histogram, therefore you use Tobi‘s programme and you have to perform the following steps for each run:
- ./Analysis -input [absolute path plus folder where data are, e.g. /home/vault/capm/sn0123/PMT_LED/Run00XXX/] -read -binary -sign -1
  - ➔ this generates you rootfile where all waveforms are stored in tree, so you can delete original binary files from oscilloscope after this step (input.root is new file)
  - ➔ you may have to use an older version for **generation of input trees** as otherwise trigger time may not be properly stored
- ./Analysis -input /home/vault/capm/sn0123/PMT_LED/Run00XXX/ -frames 900000 -datatree -bounds 15.0 35.0 generates **file** datatree.root with all necessary **information for further analysis**
  - ➔ integral, peak maximum, time of maximum, baseline mean and rms in branches
  - ➔ maybe rename it in histogram as datatree_900000.root so you can distinguish it from possible other trees you want to make
  - ➔ restriction of 900000 frames is important for next program, so that every tree has same number of events
  - ➔ you could also take another number, but I got 2 trees instead of 1 when I use -frames 1000000, so I chose 9000000
  - ➔ bounds defines you the integration interval
  - ➔ if you want to **check your tree**, you can either type root -l and then new TBrowser and check the branches and leaves, or you check with the following commands in the shell:
    - ➢ root datatree.root or root -l, Tfile *file = Tfile::Open(„datatree.root“) [Opens file]
    - ➢ file->ls() [shows all trees in the file]
    - ➢ treename->Print() [shows info to tree, number of branches, their size and so on]
    - ➢ treename→Draw(„peak_integral >> h1(400,0,10)“, „baseline_rms<0.2“) [you can perform already some „little analysis“, this draws you the integral into a histogram between 0 and 10 with 400 bins, but only if the baseline_rms is smaller than 0.2]

for the following analysis steps you need some **further small programs**:

- get_baseline_cuts_gaus_2.cpp
  - ➔ this is a root macro, so you have to open root firstly in the shell, e.g. type root -l and then .x get_baseline_cuts_gaus_2.cpp
  - ➔ before use, you have to modify the script to your files
  - ➔ it reads the datatree with 900000 entries and applies different cuts: 0-3 sigma to the left of the maximum of the baseline_rms histogram and saves tree with integral and baseline_rms to fit_tree_n_sigma.root for each cut
  - ➔ you may have to optimise parameters

- now you have the integral data you need for the **pe-fitting** and you can start to fit the integral distribution
  - fit_1pe_w_cuts.py if you want to fit just 1 run (for test reason or so)
  - call: python fit_1pe_w_cuts.py --file filepath/fit_tree_n_sigma.root
  - this is for the unmodified model (so in principle wrong and you should not use it)
  - fit_1pe_w_cuts_res.py is some further advanced (see program list), but also for unmodified model
  - for the modified model, it is:
  - → fit_1pe_w_cuts_res_mod.py (this also includes saving of fit parameters and histogram to plot later, call: python fit_1pe_w_cuts_res_mod.py --file filepath/fit_tree_n_sigma.root)
  - → there are 8 fit parameters, so you have a lot of possibilities to test, but I got the best fit results when taking the given parameters within their allowed limits and only vary Q1 and sigma1 a little, all other stuff deteriorates the fit too much
- when you have found the best bounds (the problematic bounds were the lower limit of Q1 but especially the upper limit of sigma1), you can use the script for **fitting all your runs at once**
  - fit_1pe_all_1sig_cut_mod_ufin.py
  - fit_1pe_single_1sig_cut_mod_ufin.py
  - script ‚all‘ calls script ‚single‘, here you have to modify the paths, datafiles and limits
  - the reason I have 2 scripts and not 1 is that root somehow (nobody could help me or explain why this is) ‚memorizes‘ the limits of the run before when I put this in one script and then the fits didn‘t work (except for the first one), so I made it with one script calling the other and this worked
  - you have to create the folders ‚residual‘, ‚histogram_data‘, ‚fit_data‘
  - all parameters are saved for all runs, so you can **plot different dependencies** if you want
- plot_fit_param.py should plot you some of the combinations in python
  - call: python plot_fit_param.py
- if you have measured different intensities, you may want to calculate the weighted mean
- if you have measured different intensities for different temperatures, you may want to calculate the weighted mean and extrapolate to -100°C
- I only did **weighted mean** and **extrapolation** together (also with determination of PDE), so you may have to select in the program what you want to do:
  - calc_weighted_mean.py is for different cuts
  - calc_w_mean_pde_one_cut.py is weighted mean, extrapolation and PDE for only one cut; it also saves you the important temperatures vs mean Q1 values with errors