Table of Contents

How to produce plots (as a shifter)

As a shifter, running the analysis code is remarkably simple, to a point. From any terminal, logged in as vertextb, just run the command "monitor run_no" where run_no is the number of the run you want to analyse. The viewer will appear when the analysis step has finished, allowing you to check the data quality.

At present the monitoring is set to run over all events, so will take ~8 minutes to produce plots when run.

More detail

Below is a little more detail on each algorithm, in case things don't quite go to plan...

The scripts which are being run are all located in /home/dhynds/tbcode/macros. Which algorithms are run is set by the monitoringPlots.sh script, which does some simple checks to see if the data needs to be converted etc. There is a specific folder structure where things are written to: converted data is placed in "rawDataLocal", root ntuples are placed in "rootDataLocal", alignment files are found in "cond", files to mask pixels are located in "masks" and the output histogram files are written to "monitoring". The scripts called by the monitoringPlots.sh are:

- Convert.sh: This script calls a converter program compiled as part of eudaq: /home/vertextb/eudaqSG/bin/MonitorEventsSG.exe. It takes as an argument the number of events to process (-e), the number of events to combine in a single output file (-g) and the location of the folder to write to (-p). For the analysis software converting this into an ntuple appears to run much faster when there are many files with few data, so a compromise value of 10 events/file should be used.

- Amalgamate.sh: This calls the analysis software (/home/dhynds/tbcode/bin/tpanal) for the first time, producing an ordered event file. It takes the folder containing the raw data (-r) and the output ntuple name (-z) as arguments, with an option quiet mode (-q) preventing the displaying of histograms. In order to save disk space on pcvertextb the amalgamate.sh script removes the converted data after checking that the ntuple was made.

- Analyse.sh: Not surprisingly this runs the analysis code over the ntuple produced and makes the output histograms. The other input required is an alignment file - by default if no alignment file exists it will take a copy of the "current" alignment, which is placed in cond/Alignment.dat after the telescope and DUT have been aligned. There are a host of command line options used for the analysis code, most of which are self-explanatory. The only one worth mentioning is the track window file, which restricts the clustering and reconstruction on the telescope planes to a specific area - chosen to overlap with the clicpix. This must be tuned each time something moves!

- Viewdata.sh: This simply launches bin/tpmon, a simple root gui which displays some straightforward diagnostics for the run. Useful plots to check are the differences x an y plots (correlations between one of the mimosa planes and all others) which should show sharp peaks around 0 if the telescope is aligned, and the residual x and y plots which show the difference between the cluster position and the fitted track position on each plane. The resolution of each of the telescope planes should ideally be less than 3 um.

In the end there are two files which contain all of the "interesting" output: histogramsX.root and logX.txt. These are both written to the monitoring folder and contain the analysis results for the run.

-- DanielHynds - 2015-05-12

This topic: CLIC > TestbeamMonitoring
Topic revision: r3 - 2021-08-12 - DominikDannheim