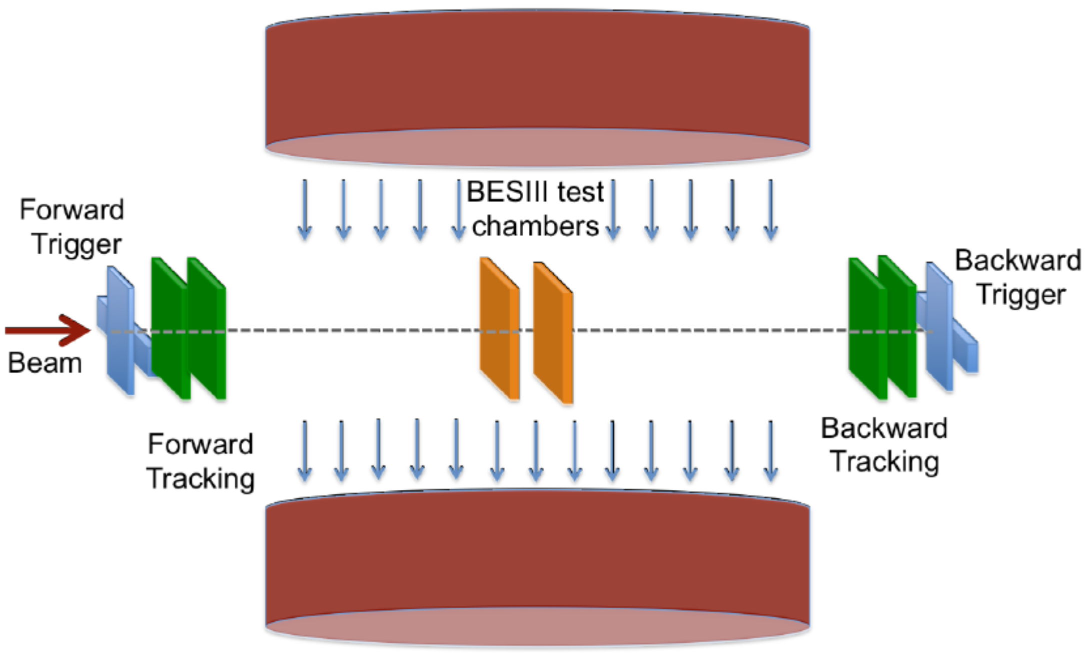
**SETUP TEST BEAM**



**ANALYSIS CODE:**

**Two steps:**

- **Reconstruction of 1D cluster**
- **Analysis of the detector performances: efficiency, space resolution, tracking, etc**

**Forward Tracking: 2- Triple GEM detector with 2D view (X-Y) <u>& the active area is 10x10 cm2</u>**

**in the reconstruction code are called plane 0 and plane 1;**

**in the analysis code are called TRK_A and TRK_B;**

**Backward Tracking: 2- Triple GEM detector with 2D view (X-Y) <u>& the active area is 10x10 cm2</u>**

**in the reconstruction are called plane 5 and plane 6;**

**in the analysis code are called TRK_C and TRK_D;**

**<u>The Trackers chambers are outside the magnetic field.</u>**

**The test chambers are:**

**LNF-BESIII Triple-GEM with a 2D view (X-Y) → in the reconstruction is called plane 2;**

**FE-BESIII Triple-GEM with a 2D view (X-V) → in the reconstruction is called plane 3;**

**RWELL with a 1D view (only Y) <u>& the active area is 5x5 cm2</u>→ in the reconstruction is called plane 4 and the in the analysis is called WELL;**

**<u>The magnetic bending is only in the Y-axis</u>**

**Main Directory: Test_Beam_2015**

## RUN DIRECTORY:

**Run**: contains all the acquired runs coming from SRS+APV system:

Each run is a rootpla where you can find information like:

- APV_id,
- APV_strip;
- APV_q;
- Etc;

You can download the runs from:

dropbox

Start with the runs from 670 to 679 (scan in HV & B=0) for the cluster mode.

Start with the runs from 907 to 911 (scan in Drift Field & B=0 & Angle=30) for the micro-TPC mode.

I also attached an excel file with the different scan in magnetic field and angle (WELL_H4_2015_logbook_online) and the log-book.

## RECONSTRUCTION CODE:

**Rec**: contains in the "src directory" several codes for the Hit & the Cluster1d reconstruction:

**parameter.h** there are some inputs of our setup:

const float pitchBen = 0.4; is the pitch of the WELL detector in mm

const int APVmap[20] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ,11, 12, 13, 14, 15, 16, 17, 18, 19}; is the number of AVP (we used 14th APV);

const int Planemap[20] = {0, 1, 0, 1, 5, 6, 5, 6, 3, 2, 3, 2, 3, -1, 4, 4, -1, -1, -1, -1}; is the number associated to a detector: 0 is the 1st tracker; 1 the 2nd tracker; 2 is the LNF-GEM test chamber; 3 is the Fe-GEM test chamber; 4 is the WELL test chamber; 5 is the 3rd trackers;6 is the 4th tracker;

const int Viewmap[20] = {1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 2, 0, 2, -1, 1, 1, -1, -1, -1, -1}; is the number associated to the view; 0= the X view; 1= the Y view; 2= the V view;

const float Zpos[7] = {0., 62., 3514., 3546., 3906., 7403., 7465.}; is the position of each chamber along the beam (z-direction). For example the 1st tracker is placed on 0.; following the 2nd tracker is in 62., LNF-GEM is in 3514, Fe-GEM is in 3546, WELL is in 3906, 3rd tracker is in 7403 and 4th tracker is in 7465.

const float AllCoeff_x[7] & const float AllCoeff_y[7] are the X & Y alignment coeff for the 7 planes following the same order of Zpos. The values should be changed for the system alignment.

**RecSelector.C** contains the code (CLUSTERING MODE):

**Digitizer.C** → takes the information coming from the SRS system + APV:

For each channel, it loops on the sample of the APV (in bin of 25 ns) searching for the maximum of the charge, and registering all the HIT information such as the coordinates, the max charge, the plane, etc.

**Reorder.C** -> ordering of the Hit_channels from the biggest to the lower. This code is needed only for the RWELL because we used a strange pattern to readout the chamber, and the output, before this code, is all the even channels and then the odd ones. This code is not written well but it works and allow to order channels. If you want, you could write in a better way.

**Clusterizer.C** → this code produce a cluster in a single view.

To process these codes you go to the "**rec directory**" and digit:

**make clean;**

**make;**

**./bin/recSelector run###.root**

P.S<u>. if you change somethings in the file.h, you need to digit always the above commands</u>.

For the micro-TPC mode is needed to change the Digitizer.C, introducing the Fermi-DIRAC fit for the RWELL and not for the tracking clambers, and saving the several information and of course the Clusterizer.C.

**<u>RECONSTRUCTION REPOSITORY</u>**

Now you produce a file like **"rec_run###.root"** that you can find in the "<u>well directory</u>" which can be open with root, showing all the information about the hit and the cluster for all the chambers (trackers + test chambers).

## <u>ANALYSIS CODE:</u>

**Code**: contains in the "<u>src directory</u>" some codes for the analysis (CLUSTERING MODE):

**ana.h** → contains some fine adjustment for the alignments of all the chambers plus a cut for a spatial search window of the WELL cluster with respect to a track, and the declaration of all the histograms.

**anaSelector.C** → contains the analysis of the WELL detector.

Since the WELL active area is lower than the active area of the trackers, I put a cut on the X-Y views to perform an efficiency measurement. Then I ask to have only 1 cluster (in both views) for all the trackers (<u>the tracker C view Y which corresponds to the plane 5 view Y doesn't work well so I ask its cluster >=0</u>)

Then I perform some studies of the trackers for the evaluation of the tracking error.

Finally, I perform the analysis when the WELL test chamber have more than 1 cluster in an event or only one cluster.

To process these codes you go to the "**code directory**" and digit:

**make clean;**

**make;**

**./bin/anaSelector rec_run###.root**

P.S. If you change somethings in the file.h, you need to digit always the above commands.

## ANALYSIS REPOSITORY

Now you produce a file like **“ana_run###.root”** that you can find in the “ana directory” which can be open with root a showing all the information about the histograms filled in the analysis.