A Telescope Using CMS PSI46 Pixels and the CAPTAN for Acquisition and Control over Gigabit Ethernet

Ryan A. Rivera, Marcos Turqueti, Lorenzo Uplegger

Abstract–The Electronic Systems Engineering department of the Computing Division at the Fermi National Accelerator Laboratory has assembled a pixel test beam telescope for high energy physics detector research and development. The telescope features CMS PSI46 readout chips and a data acquisition and control system known as the Compact And Programmable daTa Acquisition Node or CAPTAN. The CAPTAN is a flexible and powerful system that meets the readout and control demands of a variety of pixel and strip detectors for high energy physics applications. The CAPTAN functions in a gigabit Ethernet network, which facilitates the coordination of the multiple pixel planes of the telescope. Through the use of the CAPTAN hardware, a unified telescope system is attained encompassing both the CMS PSI46 pixel components and a device under test. This paper discusses results from the telescope project including mechanical design, alignment procedure, and attainable precision.

I. INTRODUCTION

During a two week period in June of 2009, a test beam run was completed for a tracking telescope at the Meson Test Beam Facility of Fermi National Accelerator Laboratory (FNAL). The telescope used CMS PSI46 readout chips and the CAPTAN [1], [3] for data acquisition and control over gigabit Ethernet. A CMS PSI46 readout chip, coupled with a crystal diamond sensor, was the device under test (DUT). The purpose of the run was to characterize the PSI46 functionality when paired with a diamond sensor. The run results are presented in this paper, along with the details of the telescope hardware and analysis software.

This was the first run for any telescope using the CAPTAN control and data acquisition system and so, as might have been expected, some unforeseen problems arose. Due to a data labeling bug, that was only realized after the fact, the alignment and analysis became highly labor intensive. After considering that the next test beam run for the telescope is scheduled for December of 2009, and the techniques devised to recover the data were uniquely applicable to this initial run, it was decided to reconstruct just a partial set of the data, and then shift focus to the future run.

The results presented herein are from a six minute data gathering window, where 321,142 events were collected.

This pixel test beam telescope project was executed by the Electronic Systems Engineering department of the Computing Division at FNAL.

The mechanical assembly was completed by the Mechanical Department of the Particle Physics Division at FNAL.

The alignment and analysis software was created by the Electronic Systems Engineering department of the Computing Division at FNAL.

II. MECHANICAL DESIGN

The foundation for the telescope was a large mechanical table provided as part of the Meson Test Beam facility. The table is capable of translations and tilts via remote control. The remote control is necessary for positioning the telescope within the beam enclosure while the beam is active and a radiation hazard. In the control room, the table control software was used, in conjunction with the telescope software [2], for the mechanical alignment of the telescope.

![Carbon Fiber](Image)

Fig. 1. The images above are views of a pixel station used in the telescope. There are two carbon fiber pieces, one on each side of the aluminum support, where the pixels are mounted. The CAPTAN is also shown, which is the interface between the pixels and the Ethernet network.

Mounted on top of the table was a dark box that housed the telescope. Within the dark box, there were two telescope pixel stations and one DUT pixel station. The backbone of each
station was a 15” x 12” aluminum support shown in Fig. 1. The station electronics all were screwed to the aluminum support.

The two telescope pixel stations had two carbon fiber pieces mounted to the aluminum. Each carbon fiber piece was 10 cm x 10 cm square, and the CMS PSI46 module was glued to the center. One of the 2 x 4 modules is shown in Fig. 2 in a test bench configuration that was used for early debugging.

The DUT station only had a single pixel test card which was screwed to the support. The DUT was a single CMS PSI46 readout chip with approximately a quarter of the pixels mated to a single, crystal diamond sensor.

There are a grand total of 28 readout chips that make up the two telescope pixel stations – the chip positions are shown in Fig. 3. Each of the two stations has one PSI46 module that is 2 x 4, and another that is 2 x 3. The two modules are rotated by 90 degrees, with respect to each other, to give precision in the X and Y dimensions. They have roughly a 2 x 2 chip overlap. Each chip is approximately 1 cm x 1 cm with pixels having dimension 100 µm x 150 µm.

Each of the three CAPTANs was connected to a standard gigabit Ethernet switch to form a local network. Data acquisition and control were handled over a single gigabit Ethernet connection between the CAPTAN local network and a quad-core PC.

The PC ran the data acquisition and control software, and stored the telescope data to a local drive. The data could then be accessed through the Fermilab network. The complete system layout is depicted in Fig. 4.

III. ALIGNMENT AND ANALYSIS

The alignment and analysis was conducted by a Visual C++ software application for Windows shown in Fig. 5. The software utilizes the OpenGL library for three dimensional viewing of results and progress.

The software has several key functionalities: event building, track fitting, alignment, and track simulation. Additionally, the software can place filters on the events to search for particular data features. It also allows users to change the number of chips that make up the telescope planes, or add and remove planes altogether. This flexibility should maintain the software’s utility for future configurations of the telescope.

The external accelerator clock and scintillator trigger arrived over LEMO cables to the central DUT station. The clock and trigger lines were distributed from the DUT station, over SATA cables to the outer two planes. The power was supplied by three channels, each at 3.3 V.

The power arrived over 2-pin, 100 mil connectors individually to the telescope stations.
A. Event Building

After data collection, the data from each individual telescope pixel plane is stored in its own directory. Within each directory, the runs are segmented into many partial-run files where the binary data resides.

The first step in building the events is to merge all the temporally correlated data from every pixel plane into software structures that represent complete events. To accomplish this, the individual directories are scanned sequentially and pixel data, with matching event ID tags, are pulled from the files and merged.

Locating data with matching event ID tags turned out to be the biggest obstacle to successfully reconstructing the telescope data. The problem occurred in the handling of the readout token by the data acquisition system. When enough hits were recorded by a chip in response to a trigger, the chip could hold the readout token for an extended period of time. If this period of time lasted until the next arriving trigger, then the chips deeper in the chain would receive two triggers and only one token. These mismatches could accumulate randomly and independently throughout the run for every chip in the system.

In the CMS experiment at the Large Hadron Collider, where these readout chips are also used, this scenario is handled by the Token Bit Manager (TBM) module, which awaits return of the previous token before initiating a new token. The CAPTAN system attempted to remove the TBM, but neglected to include this token stacking feature.

To resolve the mismatched ID issue, the data was partially assembled for many possible chip ID combinations, and then the ID relationships with the highest correlation were chosen. In this way, blocks of events were recovered - one by one. When a relationship started to demonstrate less correlation, a relationship boundary was created, and the software would begin searching for a new relationship.

In the six minute telescope run presented in this paper, 952 different chip relationship blocks were detected.

B. Track Fitting

Once all of the events are completely assembled, the analysis software can be used to locate the best track within an event. To accomplish this, the software loops through every possible combination of hits such that one hit is selected in each of the four planes. For each combination, a line is fit using a least squares minimization approach. Whichever line has the lowest error is recorded as the track for the event. The steps of the process are depicted in Fig. 6. The C++ fitting algorithms and matrix operations were verified using MATLAB.

The error is calculated by intersecting the line equation for the track, with the plane representation of the PSI46 module. The distance between the point of intersection and the raw pixel data position is taken as the error. The goal is to minimize the sum of the four errors at each pixel plane to find the best fitting track.

![Raw Pixel Data from Single Event](image)

![Track is Fit to Event Data](image)

![A Good Track Intersects Pixels](image)

Fig. 6. The steps for line fitting are shown above. First the raw pixel data is assembled into an event, then tracks are fit for every combination of data amongst planes, and finally the best track is chosen.

C. Alignment

Before any analysis of the data can begin, an accurate alignment of the telescope planes and DUT plane is necessary. Hand measurements of the actual plane locations within the setup can only give a rough estimate of the relative positions in free space – for useful analysis, the planes must be aligned with precision on the order of µm.

The alignment algorithm attempts to attain this precision by minimizing the aggregate error between the predicted tracks and the raw data. The algorithm to accomplish this is described here:

1. Since all of the planes were exposed to the same beam, they must share a common beam center. The initial plane alignment guess makes use of this common reference point. The beam center is found by filling
histograms with the accumulated hit locations and then curve fitting as shown in Fig. 7.

Fig. 7. Histograms for the X and Y dimensions of each of the four telescope pixel planes are shown above. The peaks of the fits to these histograms were used as the initial guess for the telescope alignment.

2. A finer alignment was then reached by considering only data near the beam center, thus making it easier to isolate highly correlated data as shown in Fig. 8.

Fig. 8. Shown above are the event hits within a small radius from the center of the beam. These hits are expected to be highly correlated in time.

3. Finally, all the data was re-introduced and an iterative alignment procedure, only over the X and Y position of the planes, was conducted to minimize the aggregate error. It was determined, empirically, that even large changes in rotation (e.g. 30 degrees) or in Z-depth had little impact on the overall error of the system. So optimization, only over X and Y, was taken as a good enough final alignment.

The final alignment can be seen in Fig. 9. The alignment procedure was able to achieve less than 100 µm of error per plane, which corresponds with the active pixel dimension of 100 µm.

Fig. 9. Shown above is the accumulated beam spot with the telescope planes and DUT aligned.

It is believed the error could be even further minimized by considering charge sharing in a more intelligent way. For this first run, when a cluster of pixels were detected, the position of the particle passing through the plane was taken to be the geometric center of the cluster. However, since the pixels report an analog charge value that is digitized with 10-bits of accuracy, that information could be used to weight the pixels of the cluster, and infer a more accurate cluster centroid. This approach will be taken in the December run’s analysis.

Once the planes are in their final alignment, the tracks from the telescope data can be projected onto the DUT to check that the DUT responds when and where the telescope tracks predict.

D. Track Simulation

The analysis software has the ability to emulate a beam and generate simulated data from the telescope pixels as if the beam actually interacted with the telescope. The parameters, such as the origin, size, and shape, of the beam can be controlled. An example beam simulation is shown in Fig. 10. The emulator can even add noise to the pixel readout to add distortions to the intersection points.

Fig. 10. A snapshot of the beam emulator is shown above. The emulated tracks are shown in green bisecting the telescope planes. The pixels intersected by the tracks are shown in orange.

This functionality was useful in developing the alignment techniques. To test the validity of the alignment algorithm, the planes were arbitrarily placed in space, and then the beam emulator created tracks and generated data for the planes. Finally, the iterative alignment algorithm was run on the data to check that the final alignment matched the initial arbitrary plane positioning.

IV. RESULTS

From the single six minute run, the telescope system acquired 321,142 events. Of these events, 180,686 (56.3%) events had at least one hit in the critical region where the telescope planes overlapped. 32,398 (10.1%) events had at least one hit in each of the four telescope planes with those four hits forming a low-error track.

A low-error track, here, refers to a track with an aggregate error of less than 1000 µm. The aggregate error is defined as the sum of the distance between the predicted hit location and
the actual hit location for each of the four telescope planes. Therefore the 1000 µm threshold implies an average leeway of 250 µm per plane.

On average, the 32,398 low-error tracks had 48.7 µm of error at each plane. This error makes sense because the pixel size is 100 µm x 150 µm, and any particle passing through the pixel area is considered to have passed through the center of the pixel. So, as the worst case, if a particle passes through the very edge of a pixel, and was only recorded by that single pixel, then the assumption that the particle passed through the middle of the pixel adds 50 µm of error in one dimension and 75 µm in the other.

The histograms of the error after the final alignment for the 32,398 low-error tracks are shown in Fig. 11-14. The iterative alignment procedure attempted to move all of the histogram centers to zero, but could not fully reach zero. The full widths at half maximum for the Gaussian fits are in the range of 45.9 µm to 93.3 µm, which is consistent with the pixel dimension limitation.

Of the 32,398 events with low-error tracks, 5,593 (17.3%) events had a track that intersected the DUT sensor, based on the final alignment.

Of these 5,593 Events, 5,121 (91.6% Efficiency) events also had some amount of DUT data associated with the event. 4,983 (89.1% Efficiency) events had low-error DUT data associated with the event.

Low-error DUT data is defined as a hit response on the DUT within 400 µm of the predicted intersection point of the track and the DUT plane. The histograms of the error, after the final alignment, at the DUT are shown in Fig. 15.
Fig. 14. X and Y Error Histogram for low-error track events on Plane 3.

Fig. 15. The X and Y error histogram for low-error tracks and low-error DUT data events on the DUT plane.

V. CONCLUSION

The results, overall, were encouraging considering the small window of test beam time allotted this project in its first run. Several issues did arise, and a lot was learned from them that will be applied to the next run beginning in December, 2009.

The goals for the next run are to fix the several bugs in the system, and to include charge sharing in the analysis once the analysis is made simpler by having reliable data tags.

It is expected that once these issues are resolved, the telescope should be able to achieve a pointing resolution of near 5 µm, and the analysis will be able to place more confidence in the efficiency numbers of the DUT.

REFERENCES

