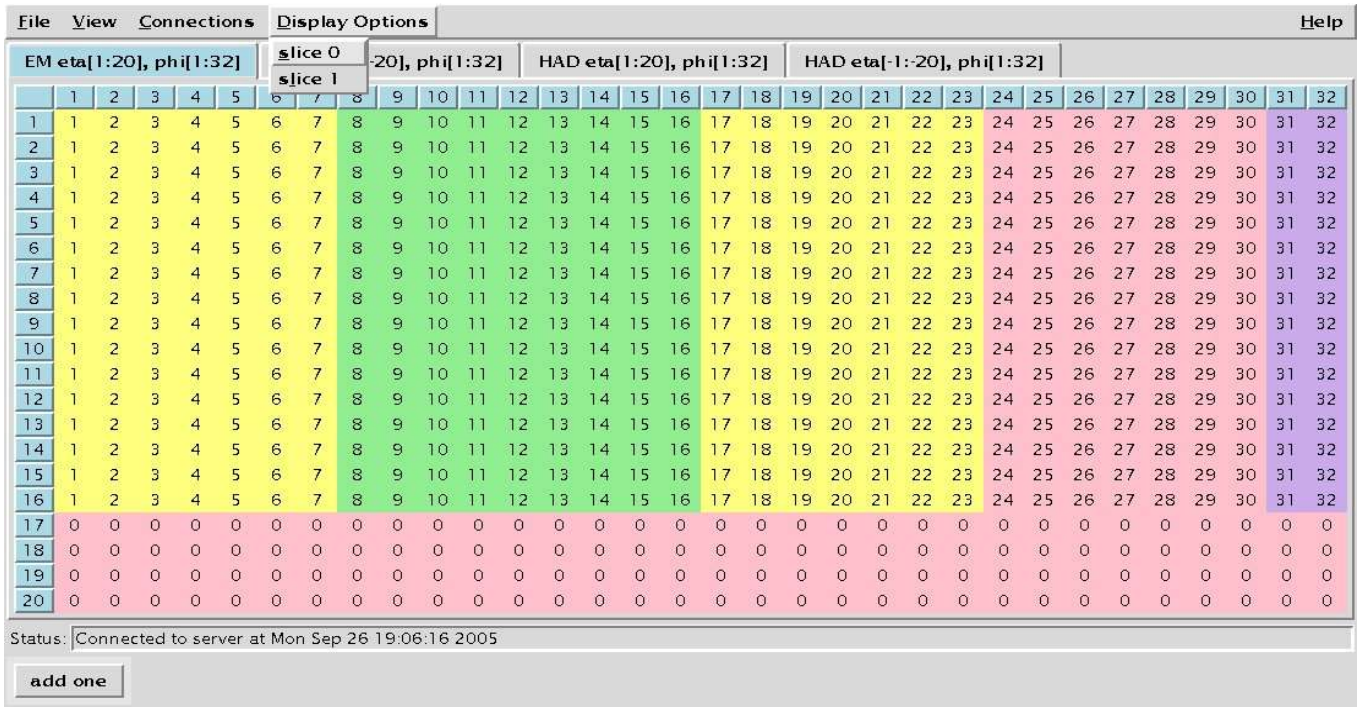
September 26, 2005
Adam Roe
Northeastern University

L1 Cal Run IIB Control Room Monitoring GUI

There is now a functional yet incomplete GUI to monitor data from the L1TCC. The purpose of this document is to present current functionalities, and open discussion regarding furhter options and functions. The GUI functions in a basic state: it dispalys all energies in all Trigger Towers (both EM and HAD portions) for a given timeslice. This GUI has two main purposes. The short-term purpose is to serve as a diagnostic tool during preparation for commisioning and commisioning. The longer-term goal is to help troubleshoot during IIB. The functionalty of this program is soley based on the TT energy readouts; seperate L1Cal IIB GUI's are still in the works for other information. For instane I am beginning work on a GUI to display Latched Bit Errors from within the TAB/GAB sytem, but that is seperate.

The GUI follows the general visual format of control room GUI's:



This is displaying a test pattern which I am receiving from a mock server at MSU. I have set the color display levels to be what they are, x<1 is red, 1<x<8 is yellow, 8<x<17 is green, and so forth.

Each page dispalys phi horizontally by eta vertically. There are two pages for each portion, corresponding to +/- eta. Thus there are four pages, which are selected by tabs at the top of the GUI.

Functions and Options

Approximately every 5 seconds, the ADF system captures the raw TT data for eight consecutive ticks, and holds it for readout by the L1TCC. Upon request, this GUI recieves a message with these data from the L1TCC. With this low frequency of random sampling, the GUI is not a full out diagnostic tool to catch every single problem. At 8 data points every five seconds, an intermittent issue would probably not be caught once, let alone twice. It can, however, see repetetive issues.

The message from the L1TCC contains 8 consecutive TT energies (each being one tick apart) for all TT's. Thus, a given timeslice for all TT's will give the energy values from a given collision.

It is worthwhile to note the stage at which the L1TCC reads data. This data is acquired in parallel to the information being sent to the TAB; the data seen here is the same data that the TAB sees.

There are many options to consider for analyzing and presenting the TT data. In theory, any number of options can be programmed in, and they can all be user-specified. In reality, we want (as always) to spend the most time programming the most important options. Here are some ideas of variations on the current display:

-- user specifices which of the eight timeslices to display.

-- instead of displaying all values for a given timslice, display the average over the eight timeslices for each TT. Rather than seeing a display pointing you towards a jet or other paricles, this gives you longer term information. It could point to a noisy TT, for instance: if a given TT has a mean of 9 (instead of 8) ADC counts over the eight timeslices over several messages, it could mean there's unaccounted for noise. Another choice here would be to display the mode instead of the mean. For any TT, this should be 8 ADC counts (the pedestal). If it is 9 or 7, there could be a consistent miscalibration.

-- instead of displaying all values for a given timslice, display for each TT the max (or min) value among the eight time slices. Again, instead of seeing a display pointing you towards particles, this could give a better idea of long-term activity, like noise level fluctuation.

-- display values in GeV instead of ADC counts.

-- ideas?

Color Displays

Another aspect of the display to consider is color levels. The program follows a silimar setup to the color displays in the control room. The basic ideas is that a different background color shows up for a given value of text, in this case energy.

For instance, if the user has chosen to display all energies for a given timeslice, it can be green for counts 7-9 (pedestal), and purple for counts lower than 7 (something is wrong). Yellow and Red can correspond to appropriate energy levels.

If we are taking an average (as a mode), the levels could be green for 8, yellow for 7 or 9, and red for anything else, or higher tolerances for mean.

As a whole, there is a question of how closely the GUI should emulate the sliding windows algorithms. We could emulate the algorithms very closely, with a red display as the center of a jet, and yellow as its sorrounding 5x5 (in eta-phi space) region. The question here is whether or not it would be useful to do this; this would be a non-trivial task.