Concrete Instructions to Running L2CAL Pulsar Simulation and Monitoring Code

Michael A. Schmidt, University of Chicago

December 15, 2006

This document describes how to run the (offline) simulation and monitoring code for the level-2 calorimeter trigger Pulsar cards. The code was written by Vadim Rusu, Wojciech Fedorko, Giorgio Cortiana and Simone Pagan Griso, and is comitted to the cvs repository.

It is assumed that the user will be running the code on the machines fcdflnxN.

1. Make sure you have sourced the cdf2 .cshrc file:

   ```
   source ~cdfsoft2/cdf2.cshrc
   ```

2. Enable the use of shared libraries:

   ```
   setenv USESHLIBS yes
   ```

3. Setup the cdf software:

   ```
   setup cdfsoft2 6.1.2
   ```

   Note: You may want to put instructions 1 through 3 in your .cshrc file

4. Make a new test release:

   ```
   newrel -t 6.1.2 trigmon
   ```

5. Go into the folder named trigmon that you just made:

   ```
   cd trigmon
   ```

6. Add Packages to your release:

   ```
   addpkg Consumer
   addpkg -h TriggerObjects
   addpkg -h PulsarMon
   addpkg -h L2GlobalTrigger
   ```

7. Remove the PulsarMon directory:

   ```
   rm -r PulsarMon
   ```

8. Checkout the L2CAL development tagged version of PulsarMon:

   ```
   cvs checkout -r L2CAL_DEV PulsarMon
   ```

9. Compile it, but don't make the executable:

   ```
   gmake nobin
   ```

10. Build the executable:

    ```
    gmake Consumer.trigmon
    ```

    At this point, you have the executable built and linked. Before you run it though, you'll want to edit the .tcl file so that it knows which dataset you'd like to analyze.

11. Edit the .tcl file:

    ```
    xemacs -nw PulsarMon/tcl/pulsar_trigmon_offline.tcl
    ```

    Edit the line beginning with 'include files' so that it points to your dataset. I know it's stupid, but don't forget to save your edits!

12. Run it and re-direct the output into a .log file:

    ```
    TrigMon PulsarMon/tcl/pulsar_trigmon_offline.tcl >& my_log_file.log &
    ```

NOTES:

- You can see what's going on while the program is running by doing:

  ```
  tail -f my_log_file.log
  ```

- Currently, the monitoring code only monitors one card at a time. This will eventually be updated such that all 18 L2CAL Pulsars can be monitored.

- The results of the program are output into the file:

  ```
  ErrorLog_LEVEL2_PULSAR_CAL_0_224521.log
  ```

- Since there are currently no real TP2D data with which to compare the simulation, the columns labeled 'REAL TP2D' and 'SIM TP2D' in the above log file are identical, and hence their XOR difference is always zero.

- The order of printouts in the .log file is:

  `CONTROL, DATAIO1_1 (input), DATAIO1_2 (output), DATAIO2_1, DATAIO2_2`

- The .tcl script can be edited to select:

  - Which cables are disabled. To disable a cable, set its 'L2CAL_CablesInputList' parameter to -1.
  - Which input cables are connected to each pulsar card. The .tcl script parameters 'L2CAL_CablesInputList' determine the cables connected to each mezzanine. Each cable corresponds to 2 eta/phi towers. The way the eta/phi tower information is encoded in the cables is:

    $$N_{cable} = (\eta + 24 * \phi) \quad div \quad 2$$

    Where N_{cable} is the cable number and div indicates integer division. Recall that each cable carries two information from the two trigger towers:

    $$(\eta, \phi) \text{ and } (\eta + 1, \phi)$$

    Thus, given an input cable number, we can determine its (η, ϕ) coordinates:

    $$\eta = 2 * N_{cable} \quad mod \quad 24$$
    $$\phi = 2 * N_{cable} \quad div \quad 24$$

    Where div indicates integer division and mod indicates the 'modulo' operation (remainder from integer division).

    The first input cable of the first mezzanine card of the first pulsar board corresponds to the first number in the 'L2CAL_CablesInputList'. This corresponds to entry '0' in C-style arrays. In this way, the first input cable to the second mezzanine card in the first pulsar board would be the 5th number (entry 4 in the array) in the list. For example the following setting:

    ```
    L2CAL_CablesInputList set 0 1 2 12 23 -1 -1 25
    ```

    Tells the code that the first input of the first mezzanine of the first pulsar board is cable 0, which corresponds to the (0, 0) and (1, 0) eta/phi towers. The second input of the first mezzanine of the first pulsar board is cable 1, etc. Note that the second and third inputs of the second mezzanine card of the first pulsar board are disabled (set to -1). Finally, the fourth input cable to the second mezzanine of the first pulsar board is number 25, which correpsonds to:

    $$\eta = (25 * 2) \quad mod \quad 24 = 2$$
    $$\phi = (25 * 2) \quad div \quad 24 = 2$$

    Thus, this cable carries the information from the (η, ϕ) towers: $(2, 2)$ and $(3, 2)$.

– MORE TO COME HERE!! STILL NEED TO INSTITUTE SOME EDITS!!