

# MONITORING

version monitor-v4.4

*Note that the system described here is not for a full monitoring system, the purpose is to have a monitoring system for tests only. The monitor and presenter tasks can handle only one FEC at a time.* The monitoring system is briefly shown in figure MON-1. The readout starts three threads, the raw event readout, and two monitoring threads. The readout is storing an event in the shared physical memory, this event is accessed by the online monitoring thread using a locking mechanism to avoid the readout to overwrite an event currently sent by the monitoring. The playback thread is reading a file, independent of the physical event buffer. An event is sent on request from the monitoring task.

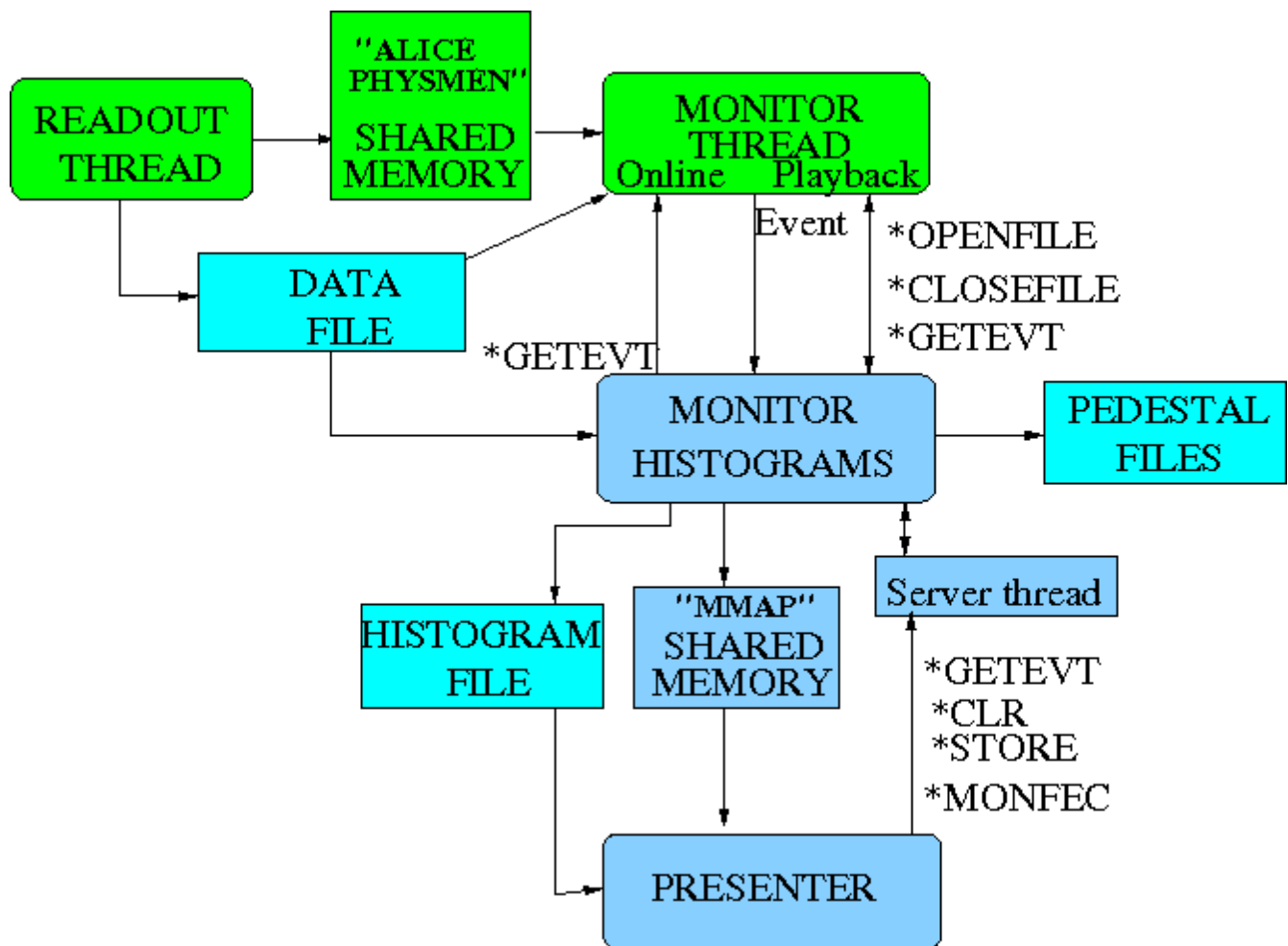


Figure MON-1: Brief overview of monitoring system

The communication between the monitoring threads and the monitoring program is through network commands, sent from the monitoring task.

\*OPENFILE <filename>            open file for playback  
 \*CLOSEFILE                        close playback file  
 \*GETEVT                            get an event, the event is sent though the network.

The event handling loop in the readout is shown in figure MON-2. The monitoring threads listen to the network for requests. The online monitoring thread is using the built in locking mechanism in pthread to have exclusive usage of the event buffer. No such mechanism is needed for playback. If there is no run active is the monitoring thread sending back a data buffer with a status message instead of a true event.

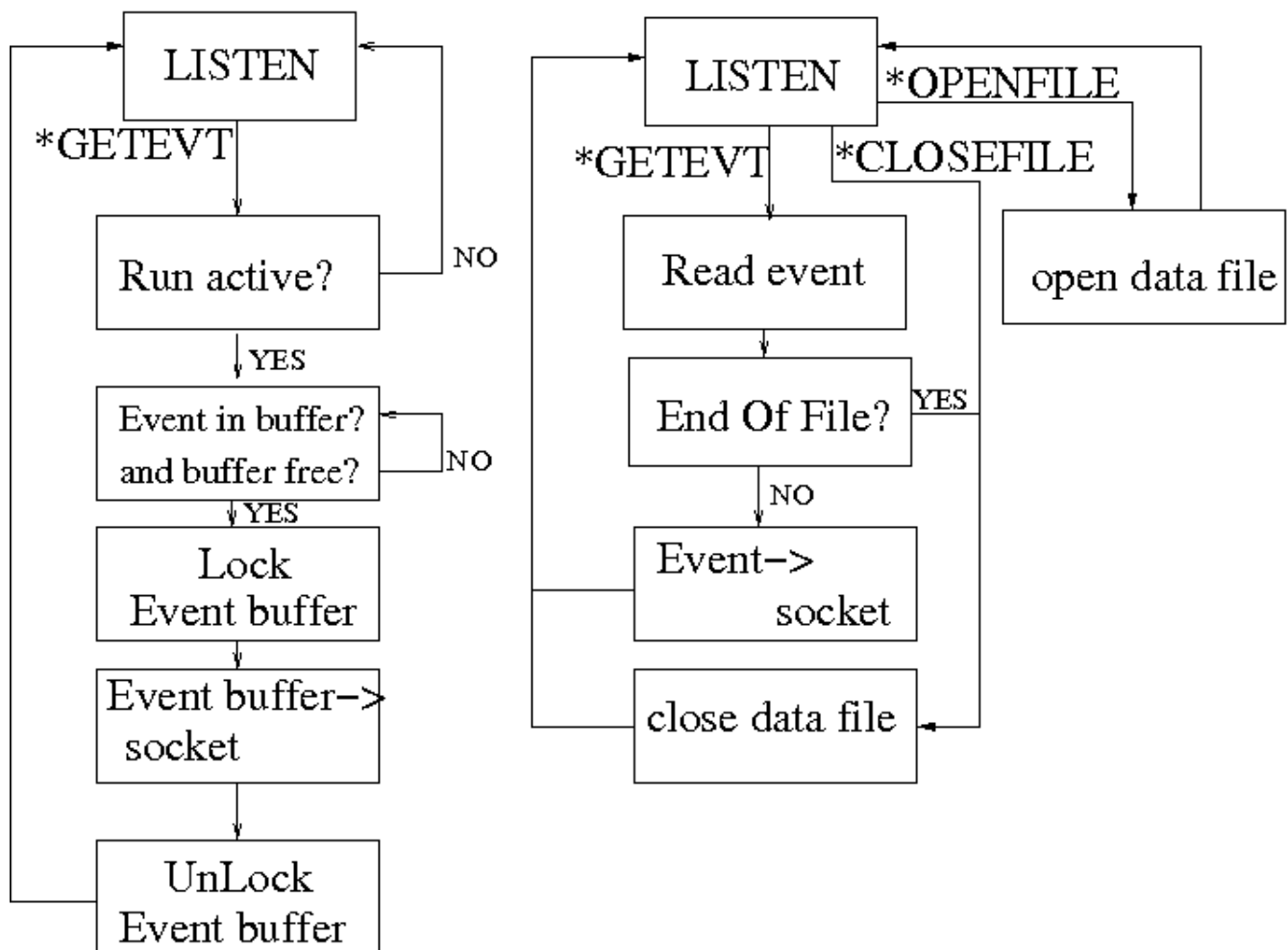


Figure MON-2: Event monitoring loop

The monitoring task can also read events from a data file directly, the data file is expected to be in DATA\_DIR as defined in monitor.h, or it can be set on the command line (--indir). It fills histograms and stores them in a ROOT format in a file (online.root) and in mmap shared memory for access by the histogram presenter. The communication with the presenter is described in the document of the

presenter. The monitor task starts up a thread for listening for commands from the presenter. All files are stored with respect to the path `ROOT_DIR` which is defined in `monitor.h`, or can be set on the command line (`--outdir`).

In a special running mode can the monitoring task be used to read a raw data file and produce pedestal files. It is also possible to do pedestal files online for special pedestal runs. The pedestal files are placed in `ROOT_DIR/pedestals/pedestals-<id>-<pcasettings = %X-%d,shiftreg,dac>.new`. They are in a format understood by the readout task to be downloaded to the ALTRO. The pedestal file must manually be copied to the configuration directory on the DAQ computer as the name `pedestals-<id>-<pcasettings = %X-%d,shiftreg,dac>.dat`. Since pedestals depend on the `pca16` settings must there be a pedestal file for each `pca16` setting used.

The histogram configuration is read from the file *config.dat*:

```
# What to histogram - overridden by command line options
```

```
#HIST.PH      pulse heights
```

```
#HIST.TIME    time history
```

```
#HIST.CHARGE  charges
```

```
#HIST.POSITION hit positions
```

```
#HIST.PED     calculate pedestals
```

```
#HIST.MAXCH   maxadc vs channel/time
```

```
#
```

```
HIST.PH      1
```

```
HIST.TIME    1
```

```
HIST.CHARGE  1
```

```
HIST.POSITION 1
```

```
HIST.PED     0
```

```
HIST.MAXCH   1
```

The file *histos.dyn* contain booking information for histograms where the x/y-axis scales can be chosen when the program is started.

```
0,Event Display,Event Display,128,-0.5,127.5,128,-0.5,127.5
```

```
1,Time Display,Time Display,128,-0.5,127.5,128,-0.5,127.5
```

Where event display is a 2-dimensional histogram where the ADC sum of a channel is filled at the (X,Y) location given the mapfile described below. In Time display is the time of a hit filled at the (X,Y) location. The presenter (described in another document) is showing this as a color coded box. The title and name of the histogram may not be changed, without changing it in the presenter as well.

The file *mapchvxy.txt* contains a table that maps a FEC to X,Y position. This is used in the histograms for event display, and time display:

```
#      RCU  FEC  CH   X           Y
```

|   |    |      |        |       |
|---|----|------|--------|-------|
| 0 | 0  | 0    | 0.00   | 0.00  |
| 0 | 0  | 1    | 0.00   | 1.00  |
| 0 | 0  | 2    | 0.00   | 2.00  |
| 0 | 0  | 3    | 0.00   | 3.00  |
| 2 | 31 | 4093 | 122.00 | 85.00 |
| 2 | 31 | 4094 | 122.00 | 86.00 |
| 2 | 31 | 4095 | 122.00 | 87.00 |

The monitoring program understands the command line arguments:

|                  |                                                                                                                         |
|------------------|-------------------------------------------------------------------------------------------------------------------------|
| .f, --fec:       | select which FEC to monitor on program startup                                                                          |
| -i, --infile :   | input file in the input directory                                                                                       |
| -I, --indir :    | input data directory overrides default (DATA_DIR in monitor.h)                                                          |
| -O, --outdir :   | output data directory overrides default(ROOT_DIR in monitor.h)                                                          |
| -p, --playback : | get event from monitor server in playback mode                                                                          |
| -a, --all :      | change file when EOF encountered in playback mode                                                                       |
| -s, --server :   | monitor server IP address                                                                                               |
| -e, --events :   | make rootfile for each event (only when reading data file)                                                              |
| -P, --pport :    | port used for communication with presenter                                                                              |
| -w, --wait :     | wait for a run from online server in seconds - then retry<br>wait for presenter to request an event when reading a file |
| -o,--online:     | take events online                                                                                                      |
| -d,--pedestal:   | allow program to calculate pedestals during online running, only for special pedestal runs.                             |
| --wait 0/1       | if reading file, 1 = wait for *GETEVT from histogram presenter to read next event                                       |
| -r,--reversed    | reversed data format expected                                                                                           |

## FILES

|                               |                                        |
|-------------------------------|----------------------------------------|
| monitor.cxx, monitor.h        | main program                           |
| JPed.cxx,JPed.h               | pedestal calculations                  |
| JMonSocket.cxx,JMonSocket.h   | network connection with monitor thread |
| JMonHist.cxx,JMonHist.h       | histogramming routines                 |
| JMonFile.cxx,JMonFile.h       | reading of raw data file               |
| JMonEvent.cxx,JMonEvent.h     | unpack an event                        |
| JMonDisplay.cxx,JMonDisplay.h | communication with presenter           |

|               |                                                                                          |
|---------------|------------------------------------------------------------------------------------------|
| Makefile      | compile and link                                                                         |
| config.dat    | which histograms to fill – read at program start                                         |
| histos.dyn    | a few histogram where limits can be changed dynamically                                  |
| mapchvsxy.txt | mapping of channels to x,y coordinates                                                   |
| mon-file.sh   | script to monitor with events read from a data file                                      |
| mon-online.sh | script to monitor events from online readout with the option to do pedestal calculations |
| mon-ped.sh    | script to run monitor to make pedestal files from a data file                            |
| readme.txt    | changes done                                                                             |

The ROOT package (<http://root.cern.ch/>) must be installed for the presenter to be compiled and executed.

## ***CHANGES since version 2.3***

2.4 histograms for max adc vs channel, histograms for time of max adc vs channel

4.0 new RCU firmware included in unpacking for online monitoring

4.3

pedestal file used as: pedestals-<id>-<pcasettings = %X-%d,shiftreg,dac>.new, Added runnb as line in file instead.

can handle more than one RCU: FECMON = ((rcu << 5) | fec)

correct bug in masking only 0x3FF in pos=3 for clf.altro\_address = clf.index0 & 0x3FF; which made channel numbers wrongly decoded.

changed format of mapchvsxy.txt to allow many RCUs

4.4

reversed data format

channel exist histograms

**Note:**

***Playback mode not working***

***When program exits it exits with a crash. Most likely due to memory allocation bug.***