

Petr Gorbounov  
FCAL meeting, CERN 8 April 2003



# Data Acquisition for the FCAL Beam Test in 2003

- Status of hardware and software work
- MiniRODs etc
- Raw Data format
- Trigger handling, CIRQ module
- Dispatcher
- Run Control

# Status of Hardware and Software

Understood, tested, debugged, done...

- Bit3 (PC-VME)
- CBD (VME-CAMAC)
- VIC (VME-VME)
- MiniRODs (VME part)
- CIRQ (trigger/busy module)
- Basic run control
- Dispatcher
  - data/command transfer
  - inter-process communication

Not tested yet, to be understood, debugged...

- SPAC master
- FEB init/configuration
- Calibration board
- MiniRODs (TTC & Link parts)
- PC -PC link
- PC - RAID
- PC – CERN Network

## Responsibilities (\*)

- P. Gorbounov:
  - online r/o software, VME and CAMAC h/w, miniRODs, run control, “producer” part of the on-line monitoring
- P. Loch:
  - online monitoring, data archiving, networking, CDR, AFS, Linux support, SPS / H6 data

## DAQ Software Design (\*)

- discard the idea of using “generic” or HEC/EM DAQ frameworks: adaptation efforts are comparable with writing one from scratch
- pick-up and adapt relevant fragments of the hardware r/o code from all possible sources, avoid any potentially dangerous techniques (interrupts, threads, asynchronous DAQ tasks etc)
- “barebone” framework, with trigger polling and unbuffered r/o

(\*) as discussed by P.L. and P.G. on February 4, 2003

## MiniRODs

- almost no documentation on the hardware, learn by adapting the s/w written for RIO
- initialization problems: presumably, due to missing `volatile` declarations for miniROD registers
- 3 miniRODs were successfully tested by downloading & reading the FIFOs from VME

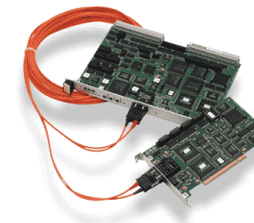


## Bit3, VME, VIC, CAMAC, memory-mapping

- low-level libraries, test programs
- no FCAL-specific read-out code written yet, but expect no problems, as the hardware is fully memory-mapped.

Typical r/o "code":

```
int *p=buffer; for (i=0; i<N; i++) *p++ = *list[i];
```



# Raw data format

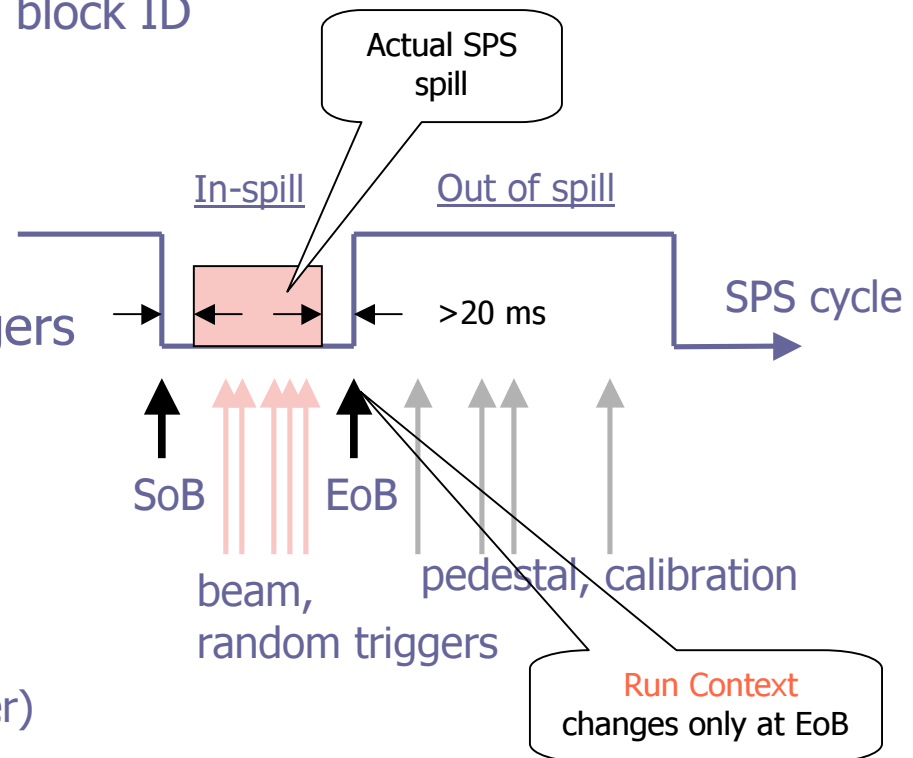
Binary UNIX file, 1 record = 1 event (readable on any platform)

- event := header, {block}, trailer
- block := header, data
- header := block\_length, block ID

## Trigger Handling

Up to 6 different types of triggers

- Start of Burst (SoB)
- Beam event
- End of Burst
- Pedestal or Random event
- Beam ch. Calibration (pulser)
- FCAL calibration (pulser)



# Event loop, Run Control: Final State Machines

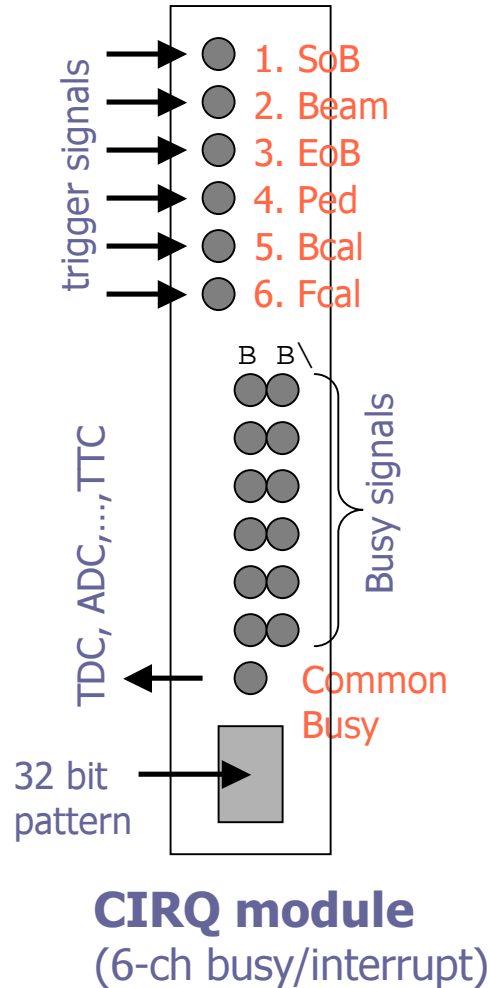
RunContext: {Starting, Running, Paused, Stopping, Stopped}

BurstContext: {In, Out} derived from SoB and EoB signals

TriggerType: {SoB, Beam, EoB, Random, BeamChCalib, FCALCalib}

*TriggerMask [RunContext][BurstContext]*

*TriggerAction [RunContext][BurstContext][TriggerType], e.g.:*



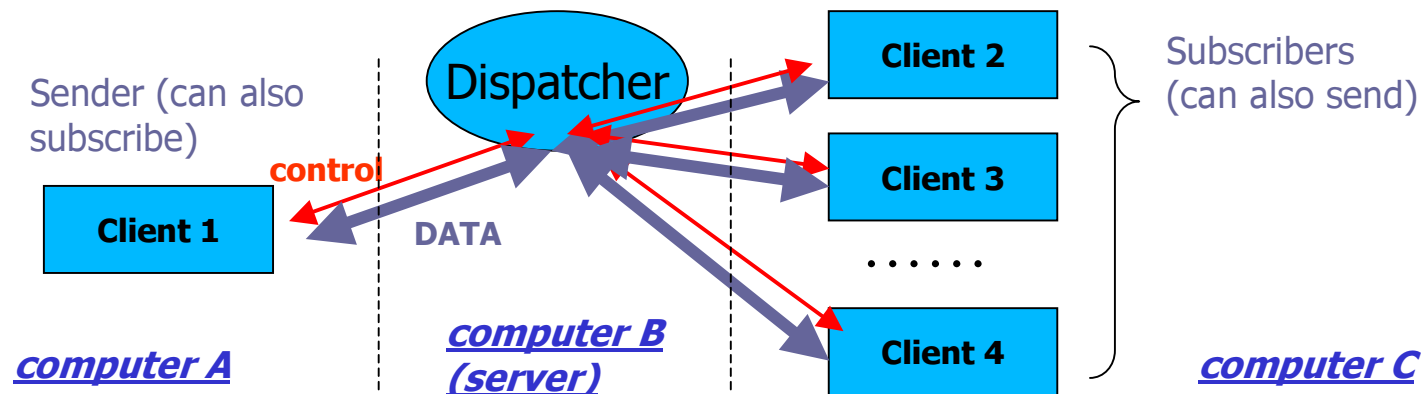
|                    | <i>Sob</i> | <i>Beam</i> | <i>Eob</i> | <i>Ped</i> | <i>Bcal</i> | <i>Fcal</i> | <i>TrigMask</i> |
|--------------------|------------|-------------|------------|------------|-------------|-------------|-----------------|
| <i>RunStarting</i> |            |             |            |            |             |             |                 |
| <i>InBurst</i>     | -1         | 0           | 3          | 0          | 0           | 0           | 0x05            |
| <i>OutBurst</i>    | 11         | 0           | -13        | 14         | 15          | 0           | 0x1d            |
| <i>RunPaused</i>   |            |             |            |            |             |             |                 |
| <i>InBurst</i>     | -21        | 0           | 23         | 0          | 0           | 0           | 0x05            |
| <i>OutBurst</i>    | 31         | 0           | -33        | 0          | 0           | 0           | 0x05            |
| <i>Runnning</i>    |            |             |            |            |             |             |                 |
| <i>InBurst</i>     | -41        | 42          | 43         | 0          | 0           | 0           | 0x07            |
| <i>OutBurst</i>    | 51         | 0           | -53        | 0          | 0           | 56          | 0x25            |
| <i>RunStopping</i> |            |             |            |            |             |             |                 |
| <i>InBurst</i>     | -61        | 0           | 63         | 0          | 0           | 0           | 0x05            |
| <i>OutBurst</i>    | 71         | 0           | -73        | 0          | 0           | 0           | 0x05            |
| <i>RunStopped</i>  |            |             |            |            |             |             |                 |
| <i>InBurst</i>     | 81         | 0           | 83         | 0          | 0           | 0           | 0x05            |
| <i>OutBurst</i>    | 91         | 0           | 93         | 0          | 0           | 0           | 0x05            |

# Dispatcher (or "Control Host")

by A.Maslennikov, R.Gurin (CASPUR,Rome)

**Functionality:** buffer manager, event handler,  
inter-process communication engine

- command transfer between RC Panel (Tcl/Tk) and the DAQ program (C)
- data transfer between the DAQ PC and the Data saver PC
- data transfer between the Saver program and distributed monitoring tasks
- custom interfaces, e.g. command/data transfer between control panels, histogram presenters and the monitoring applications
- Unix: server/client; Windows: client only (e.g. **slow control**); C, Fortran, Tcl/Tk



*(A and/or B and/or C can be the same computer)*

## Run Control: draft (demo) version is available

- Visual Tcl/Tk
- Linked to Dispatcher
- Demo with a running DAQ: later this week

The screenshot shows the 'FCAL DAO Controls' window. At the top, there are 'Daq' and 'Settings' buttons. The 'Settings' button is highlighted with a callout: 'Enabled in Run-Stop context'. The main area is divided into several sections:

- Run Title:** Run: 363, Run Type: Physics, Beam: e- 10 GeV/c, Config. File: Physics.v2, Start date/time: 21-June-2003 12:45. A callout points to the 'Config. File' dropdown: 'Run type-dependent configuration menu'.
- Run Control:** Stop after: 1000. Options:  Writing to disk,  Run Autostart. A callout points to the 'Bursts' and 'Events' checkboxes: 'Toggle'.
- Run Status:** Start, Pause, Stop, and a green 'Running' button. A callout points to the 'Running' button: 'Dynamic color'.
- Run Information:** Run Progress: 25% (with a progress bar). A table of statistics: Stat., Scalers, Beam, Recording. Bursts: 95, MBytes sent: 210.1. Total Triggers: 15885. A table of counts: SoB: 95, EoB: 95, Beam: 12345, Ped: 200, BmCal: 3000, FCal: 150.

The window title bar says 'FCAL DAO Controls' and the top right corner has window control icons. The bottom right corner of the window says 'FG v.0.1'.