THE MRF TIMING SYSTEM. THE COPLETE CONTROL SOFTWARE
INTEGRATION IN TANGO

How it works

- The EVG creates and sends out the events to the EVRs. The optical event stream transmitted is phase locked to the RF clock reference.
- EVRs decode events from the EVG. Each event code can be associated to different actions, amongst which the most common ones:
  - Trigger a pulse in any of its outputs (up to 10 outputs per EVR).
  - A configurable delay of the trigger since the event got can be set, as well as the pulse width. The delay/width can be set in 8 ns steps.
  - Set a given output to high. A delay is configurable.
  - Log the event. Events can be stored with globally distributed timestamps into a log memory.
- Forward the event downstream.
- EVRs also have 2 inputs: an event can be sent upstream when activity is detected, and then it can be redistributed downstream to the rest of EVRs.
- Fanouts/concentrators distribute 1 up-link Rx port to 8 Tx downstream ports and concentrates 8 Rx downstream to 1 upstream Tx port.

Some numbers

- 100 EVRs.
- 86 cPCI control machines.
- 418 output signals.
- 49 interlock input signals.
- 10 km of fibre cable.

Timming key functions

- Synchronously distribute timing signals to the whole machine, locked to RF frequency.
- Fast interlocking of the whole machine in 4.2 μs.
- Logging of events with globally distributed timestamps is available for machine postmortem analysis.
- Permit top-up and arbitrary filling pattern of the machine.
- Fast interlocking of the whole machine in 4.2
- Synchronously distribute timing signals to the whole machine, locked to RF frequency.
- The optical event stream transmitted is phase locked to the RF clock reference.
- EVGs create and send out the events to the EVRs. The event code can be associated to different actions.
- EVGs have 2 inputs: an event can be sent upstream when activity is detected, and then it can be redistributed downstream to the rest of EVGs.
- Fanouts/concentrators distribute 1 up-link Rx port to 8 Tx downstream ports and concentrates 8 Rx downstream to 1 upstream Tx port.

Software

- Drivers for Linux kernel 2.6 developed both for EVG and EVR.
- Pythonized API for EVG, EVR and fanouts.
- Tango device servers written in Python (using the PyTango binding).
- Tango device servers for controlling all the hardware:
  - EventG gives full control of an EVG.
  - EventR gives full control of an EVR.
  - Tango device servers for globally managing/monitoring the system:
    - TimingManager allows to set/retrieve the configuration of all the EVGs and also configure/retrieve the logs.
    - TimingMonitor is continuously monitoring the status of the EVGs, EVRs and fanouts to find any problem.
    - Tango device servers for filling any top-up:
      - Filling device server for controlling filling of the storage ring.
      - Top-up device server (under development).

Graphical User Interfaces

- EVG and EVR expert GUIs for controlling all their features.
- TimingManager GUI is the users main interaction point with the system. It is the front end for the TimingManager device server.
- TimingMonitor GUI is the front end for TimingMonitor device server, allowing a human readable display of its warnings.
- Filling GUI is the front end for Filling device server.
- Top-up GUI (to do) will be the front end for top-up device server.

Contacts: J. Moldes (jmoldes@cells.es), D. Beltrán [on leave], D. Fernández (dfernandez@cells.es), J. Jamroz (jjamroz@cells.es), J. Kiora (jkiora@cells.es), O. Matilla (omatilla@cells.es), R. Suné [on leave]