Controlling DAQ Electronics using a SCADA Framework

Luis Granado Cardoso, Joao Barbosa, Clara Gaspar, Federico Alessio, CERN, Geneva, Switzerland

LHCb is an LHC experiment and collects data from the collisions. In order to acquire the data, LHCb has an infrastructure composed of several sub-detectors to record different parameters of the events.

LHCb will be undergoing an upgrade in order to cope with the increase of luminosity delivered by the LHC accelerator. Control and monitoring of the upgraded electronics will be done via the a radiation hard chipset – GBT and SCA chips.

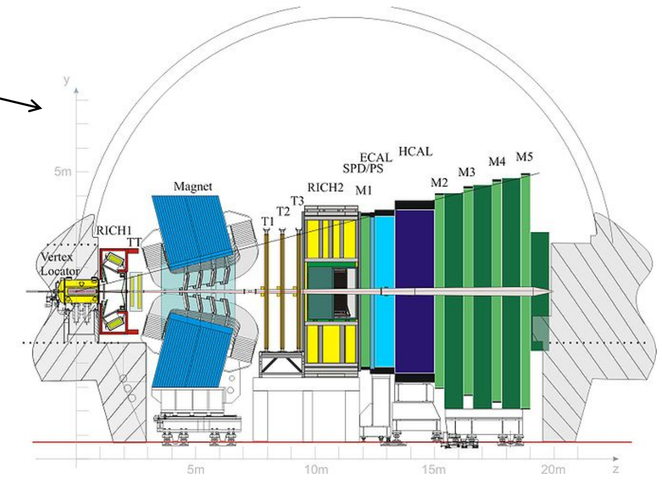
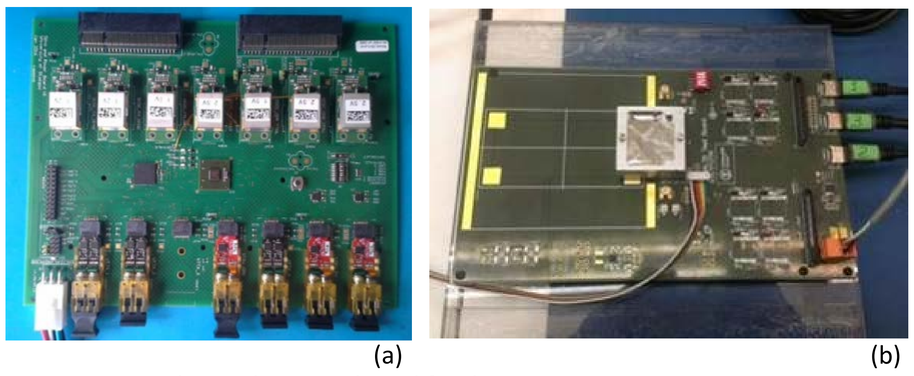


Fig 1. LHCb and its sub-detectors



(a) (b)

Fig 2. Custom demo electronic board for data acquisition:
(a) Velo OPB, (b) UT Salt8

Each sub-detector has specialized custom electronic boards and devices, which use the GBT chipset, to acquire the data from the events produced in the detector. These devices need to be integrated in the LHCb Experiment Control System (ECS) in order to be easily controlled and monitored.

A set of tools was developed to implement the communication with the new devices and integrate them into the ECS:
. GBT Server (GbtServ)
. GBT Client (fwGbt)
. Abstraction tool (fwHw)

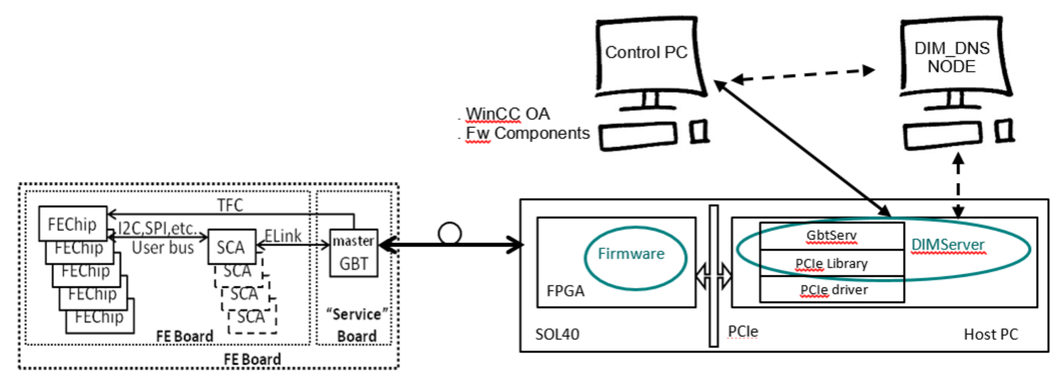


Fig 3. Control System architecture

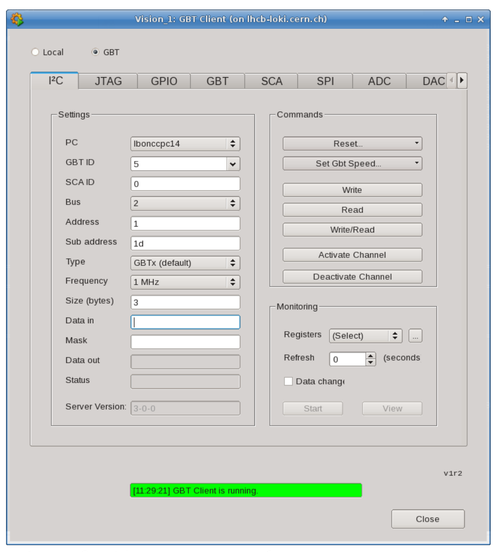


Fig 4. fwGbt User Interface

The Gbt server (GbtServ) implements the lower level communication with the FPGA board devices and firmware, the master GBT, and the SCA chips. It also implements the connection to the Control System, acting as the bridge between the Control System and the hardware devices and firmware (via DIM).

The Gbt client (fwGbt) provides the base communication between the SCADA and the GbtServ. It provides also a User Interface to easily check the connection and debug communications via all the available field-buses

The abstraction tool (fwHw) allows the abstraction of the electronic devices and create models of the hardware. This hides the complexity of accessing the hardware and allows for an easy integration into the ECS.

From an XML file containing the description of the hardware and all the settings necessary to access the hardware registers, a model can be created in the control system. Registers are then able to be accessed via their defined names. These settings are transmitted to the GbtServ that interfaces the hardware and is aware of all the different protocols to communicate with the hardware.

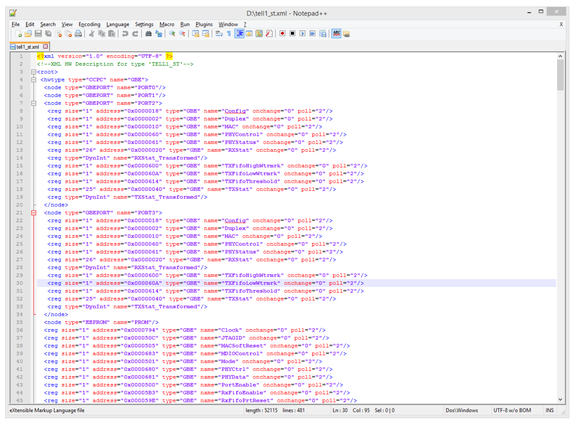


Fig 5. XML File with the hardware description

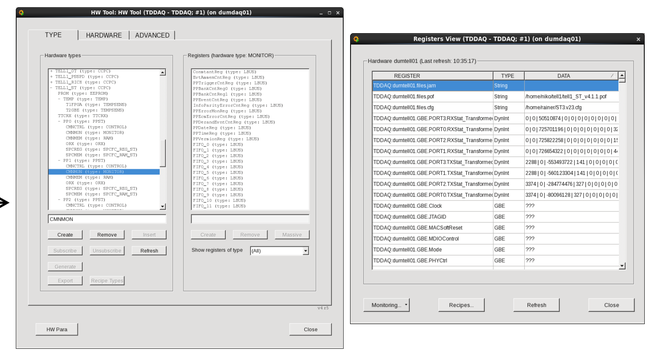


Fig 6. HW Tool with the imported model from the XML File and some of the defined named registers

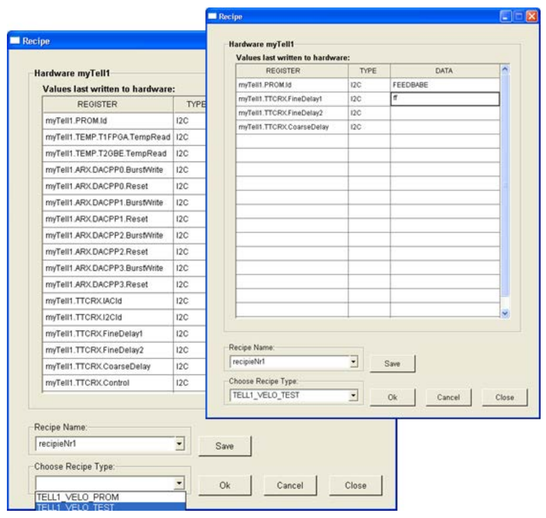


Fig 7. HW tool recipes for some registers of a device

The abstraction of the hardware into models, also allows for the usage of recipes (named sets of configurations) which can be applied to the registers of the devices. This allows to easily configure the devices for various running modes of the LHC.

Subdetectors can now create their own User Interfaces and control trees. They will use libraries provided by the fwHw tool to access their hardware via the registers defined in the models.

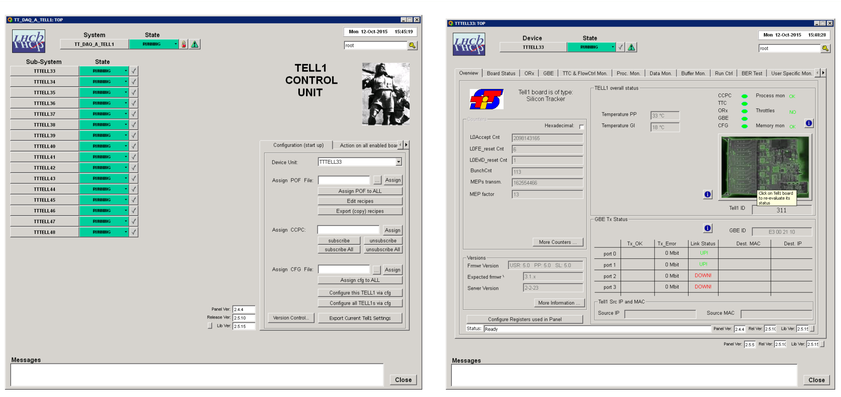


Fig 8. Control Tree and Electronic Device panel for one of the LHCb sub-detectors

The tools developed provide a complete solution to control and monitor the new electronic devices and to easily and reliably connect them to the Experiment Control System.
The Gbt server (GbtServ) and client (fwGbt) provide the interface with the electronics and easily usable low-level debugging tools, and the abstraction tool (fwHw) removes complexity from the system, by abstracting the models of the devices into the Control System. This hides the complexity of the hardware access, allowing the communication with the hardware registers to be done just using the names defined in the models.
This framework provides libraries that sub-detectors can use to easily create the specific panels and control trees for their specific devices. It also allows for the usage of recipes for easier configuration of the devices and as a base for automatic actions on the ECS.