Integration of custom DAQ Electronics in a SCADA Framework

CHEP 2019

Luís Granado Cardoso
Clara Gaspar
João Vitor Viana Barbosa
Federico Alessio
Beat Jost
Niko Neufeld
Markus Frank
Rainer Schwemmer
Paolo Durante

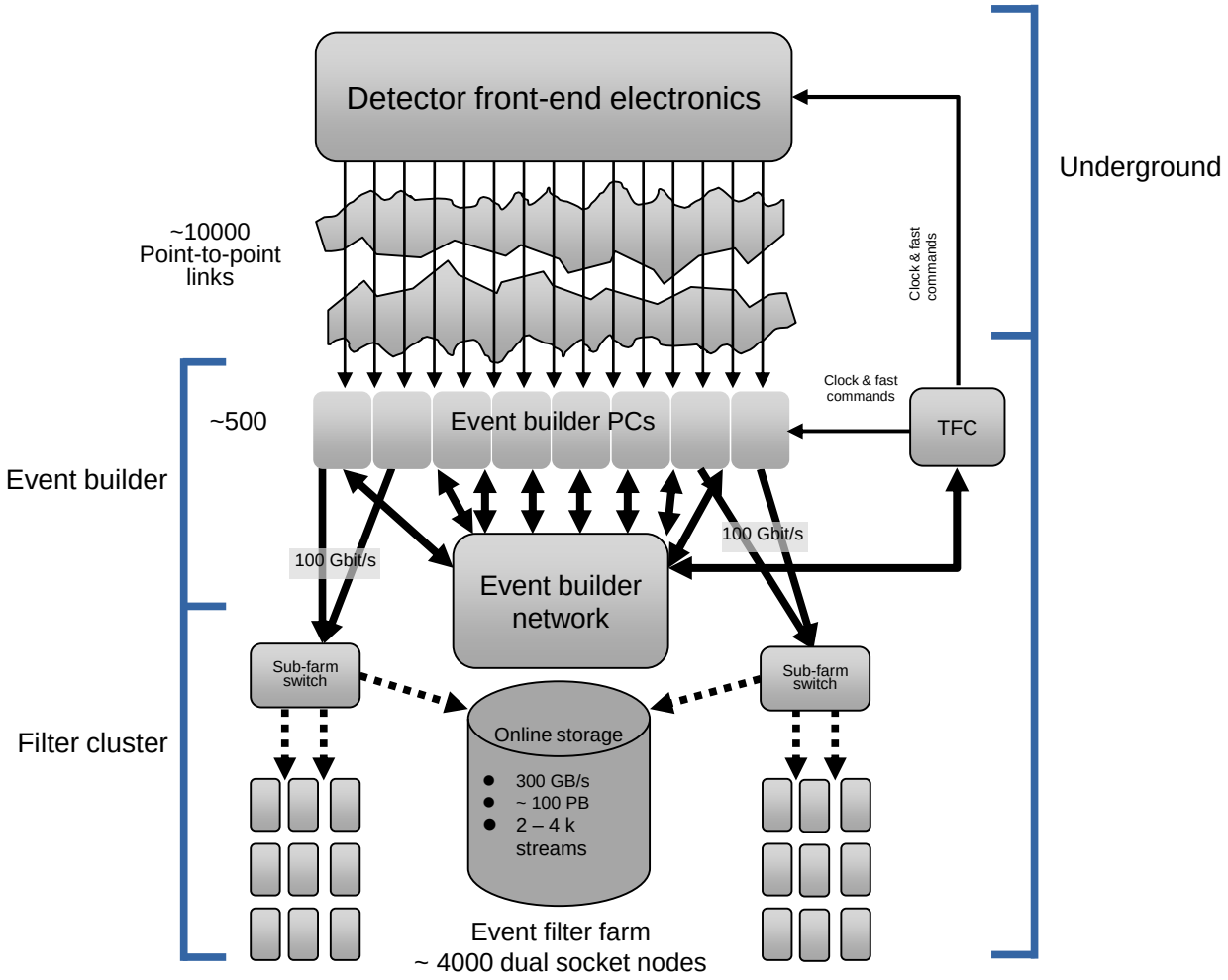
Outline

- Motivation
- Experiment Control System (ECS)
- Architecture
- Gbt Server
- Gbt Client (fwGbt)
- Hw abstraction tool (fwHw)
- Summary

Motivation

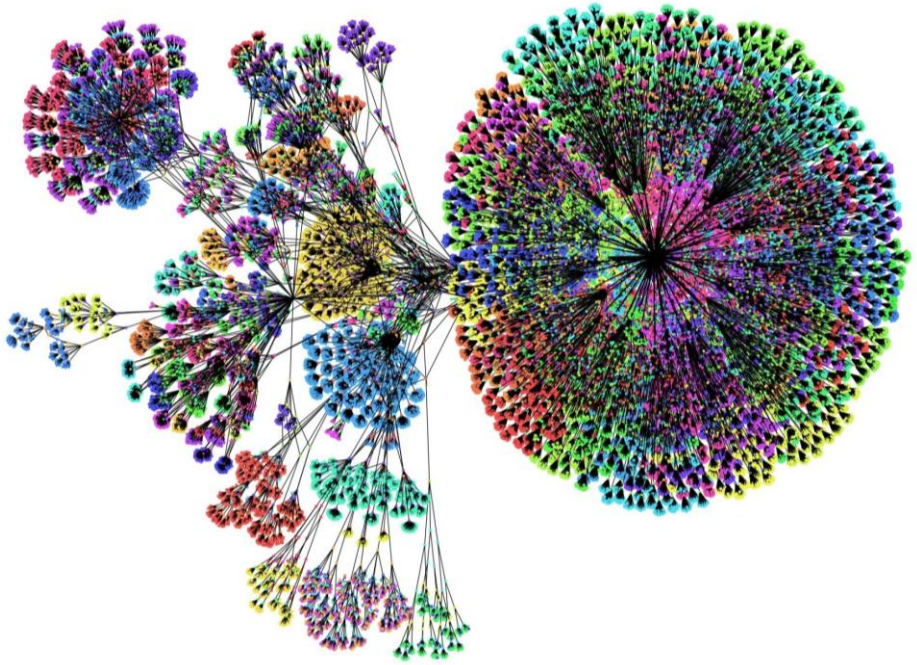
LHCb upgrade (starting 2018)

- Bottleneck L0 trigger will be removed
  - Reduced the data from 40MHz to 1.1MHz
- Trigger-less readout
- Full software trigger
- New readout system
- New/upgraded sub-detectors
  - New electronics to cope with the new data rate



Experiment Control System

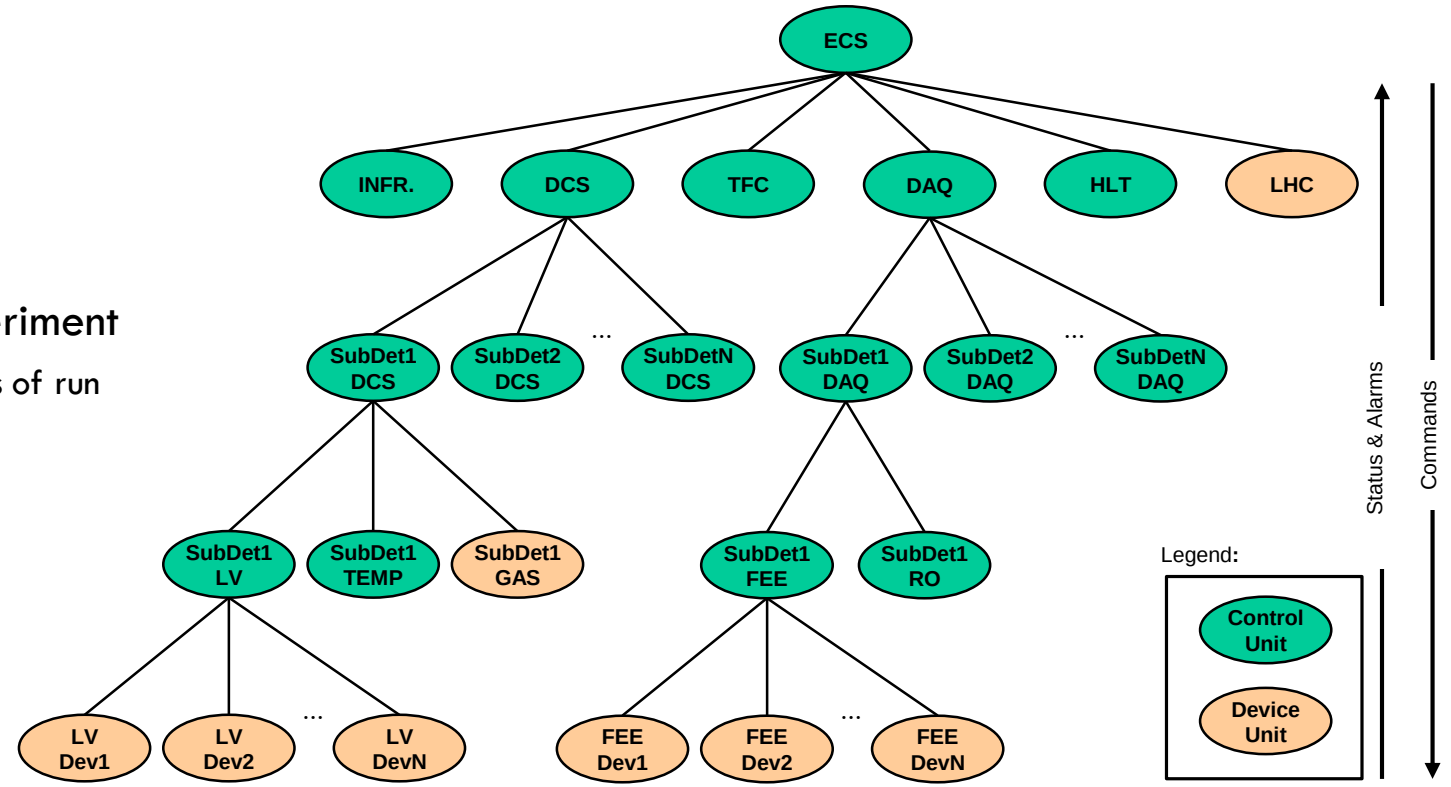
- LHCb has designed and implemented a coherent and homogeneous control system
- Controls the complete experiment
  - Run Control, DCS, …
- Is operated by only 1 person
- Is almost completely automated
- Based on WinCC OA 3.16 SCADA
  - Provides the UI, archiving, drivers, alarm-handling, …
  - Allows for custom developments
- CERN JCOP Framework
  - Provides a set of common WinCC OA components



LHCb Control System Size
Picture: Courtesy of the CMS DCS Team

ECS
Finite State Machine (FSM)

- LHCb Control System modelled with an FSM tree
- The ECS controls the whole experiment
  - DAQ Infrastructure
  - Front-End Electronics Configuration
  - DAQ processes
  - ....
- Run Control configures and the whole experiment
  - Dynamically configure according to the needs/types of run



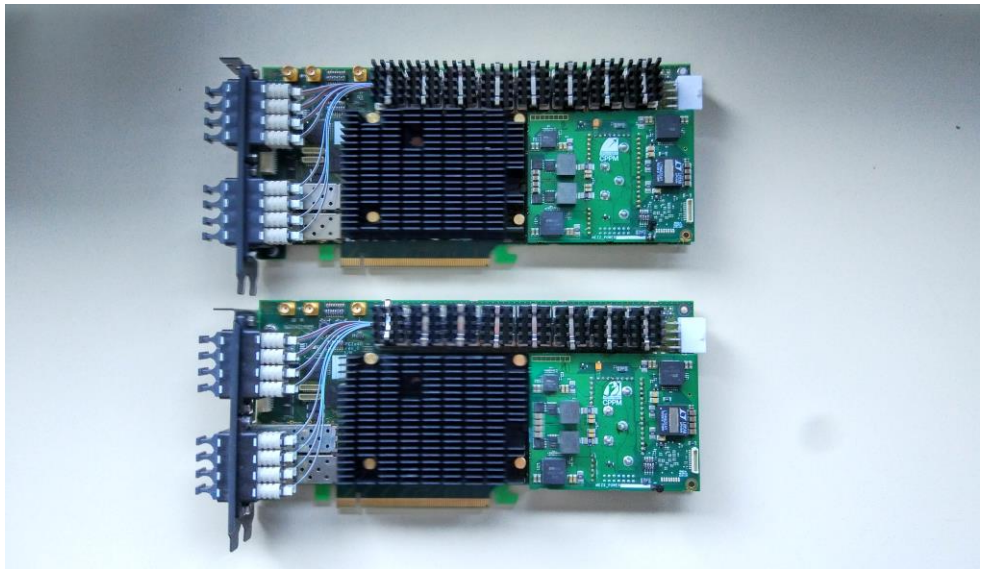
ECS
Run Control

- LHCb is composed of multiple Sub-detectors
- Each Sub-detector can be run independently in a partition
- All the Sub-detector FSMs are integrated in the Global Run Control
- Front End Electronics need to be well integrated in it the ECS
  - Eases the running of the experiment
    - Configure according to the needs
    - Allow for easy error detection/recovery
    - Allows for operation automation

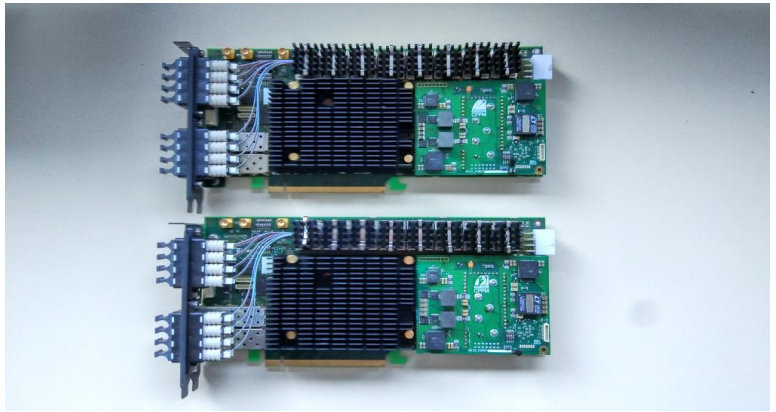
All the Sub-detector Front End electronics must be integrated in the ECS for seamless operation

ECS
Front End Electronics control

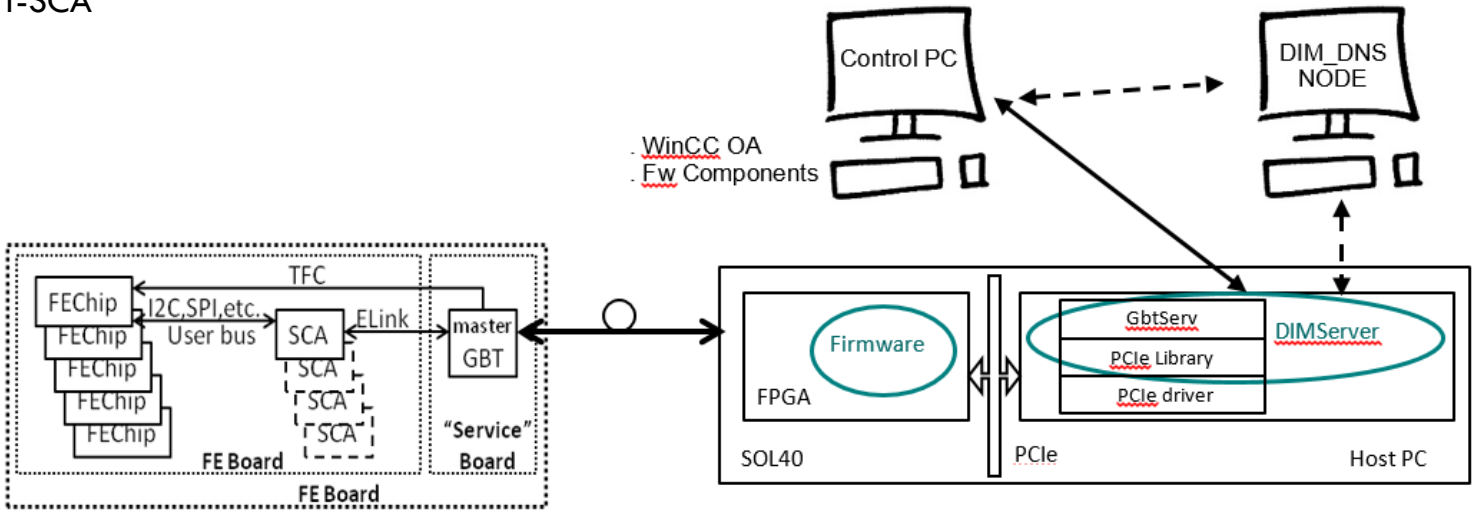
- Based on the GBT-SCA radiation hard chipset
- GBT-SCA provides multiple user buses
  - I^2C, ADC, DAC, GPIOs…
- Common chipset for all the sub-detectors
- Common framework for integration
  - SOL40-SCA firmware
  - Gbt Server
  - Gbt Client (fwGbt)
  - Hardware abstraction tool (fwHw)



Architecture

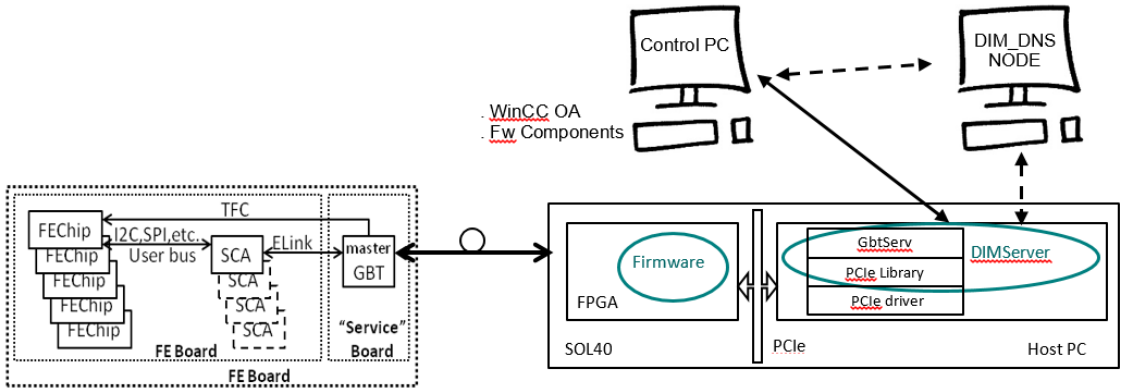


- Front End Electronics
  - Connected to the PCIe40 boards via GBT Links
- PCIe40 has the SOL40-SCA firmware
  - Provides the lower-level interface to the GBT-SCA
- Gbt Server
  - Runs on the same server the PCIe40 board
  - 1 per board
- Gbt Client
  - Runs on the Control System PC
  - Provides the basic communication between the SCADA and the Gbt Server



Gbt Server

- Runs on the server hosting the PCIe40
- Implements the low level communication to the SOL40-SCA firmware (via PCIe bus)
- Implements all the different protocol communications
  - Also custom “protocols” for easier operation
    - E.g. communications that require multiplexing
- Implements complex actions as single commands
  - E.g. low-level reset and reconfigure in 1 command
- Implements connection to the control system (via DIM)



Gbt Client (fwGbt)

- Developed as WinCC OA component
- Provides the basic communication between the SCADA and the Gbt Server
- Implements all the different protocols provided by the Gbt Server
  - all settings of the different protocols easily accessible
- Provides a interface for testing and debug of low-level communication with the FEE
- Provides User Interfaces for debugging
- Requires deep knowledge of the hardware
  - Power dependencies
  - Register addresses and settings

HW Abstraction Tool (fwHw)

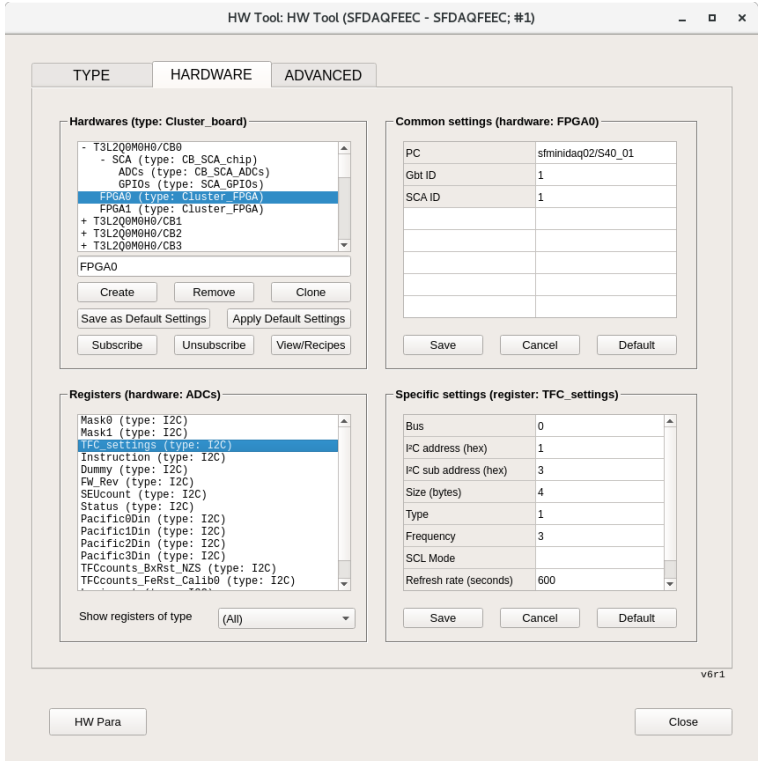
- Developed as WinCC OA component
- Supports several interfaces and their supported protocols
  - SPECS, CCPC, GBT
- Allows for the modelling of the Front End Electronics
  - Create models with blocks and sub-blocks that group common registers
  - Define named registers
    - For each protocol define all the necessary settings to communicate with the HW
    - Use the named register to access the HW
- Removes complexity from the system
  - No longer required to know all the settings of a register by heart
  - Communicate via the register name

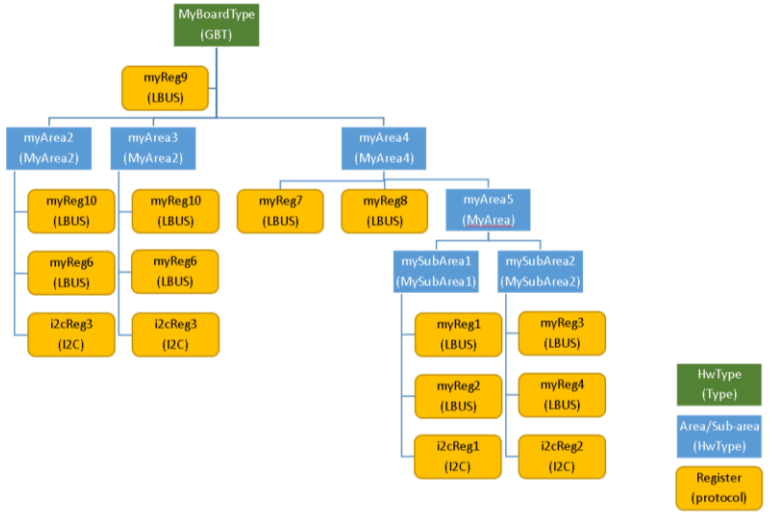
    e.g.

    ```
    fwGbt_read(T3L2Q0M0H0/C00.FPGA0.TFC_settings);
    ```

    instead of

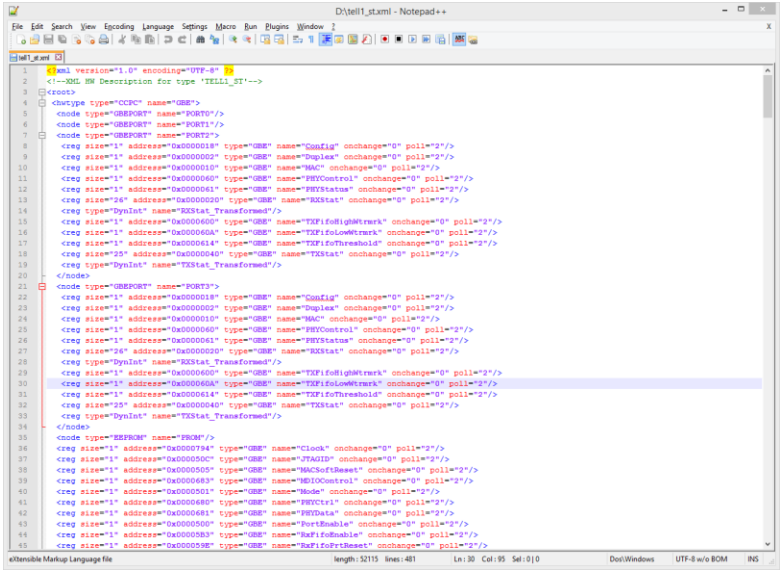
    ```
    fwGbt_I2Cread(sfminidaq02/S40_01, 1, 1, 0, 1, 3, 4, 1, 3);
    ```

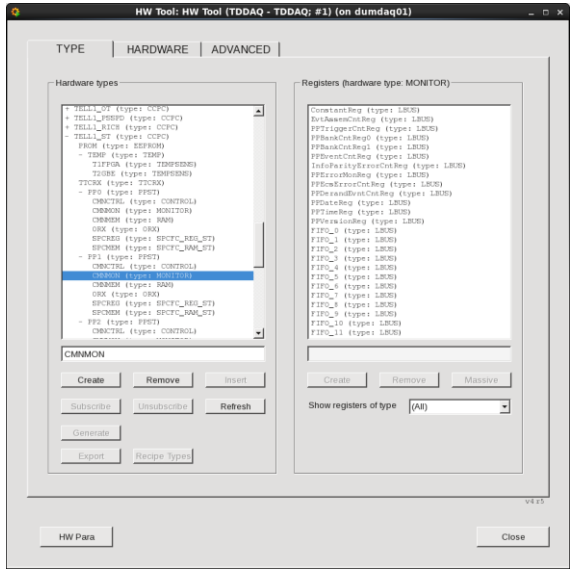


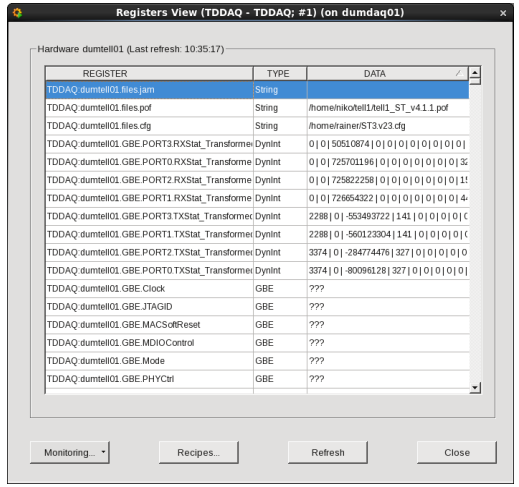


HW Abstraction Tool (fwHw)

- Easy to create models using provided tools
  - User Interface
  - Scripts
  - XML files
- Easy to change/update models and model instances
- Increases performance
  - The models can define the granularity and allow for better parallelization
    - (For instance switch on all the blocks of a given type before configuring other blocks which required this action)
  - Define one model and create as many instances as necessary
  - Allows for parallel operation for each instance

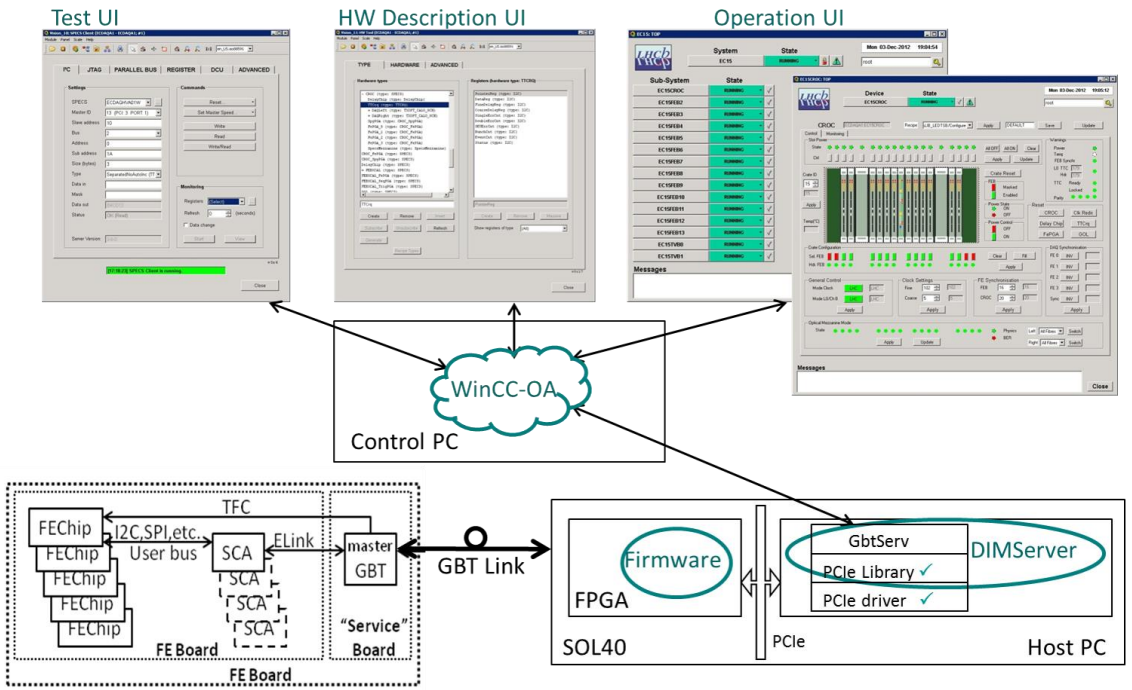






Integration

- Common set of tools libraries for all new electronics integration developments
  - Common look and feel
  - Usage familiarity
- Re-use of user code
- All the features of the SCADA available
  - Archiving and history
  - Alarms and messaging
- Good integration with the Run Control
  - Allows usage of all the available Run control features
    - Usage of configuration recipes
    - Error handling
    - Automatic actions



Summary

- Complete solution to control and monitor the new electronic devices
- Easily and reliably integrate the new Electronics 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
- fwHw removes complexity from the system, by abstracting the models of the devices into the Control System.
- Common set of tools to interact with the FEEs and for Sub-detector developments
- Homogenization of the ECS operation and “look and feel”
- Allows the implementation of useful ECS features
  - Recipes, error handling,…
- Allows for the increased automation of the LHCb ECS