Contribution ID: **5** Type: **Poster**

# Web-Based DAQ Systems: Connecting the User and Electronics Front-Ends

*Wednesday 28 September 2016 17:59 (1 minute)*

Systems designed to control and monitor particles detectors often make use of a back-end server to interact with the front-end electronics and to deliver the DAQ interface and real-time information to the user. We propose to port the functionalities of the back-end server to web interfaces and to the front-end electronics by implementing them on a dual soft-core processor : one core running an HTTP and WebSocket server to deliver the DAQ interface and establish real-time communication with the user for monitoring activities, and one core tightly connected to the electronics for control purposes.

## Summary

Web technologies are quickly evolving and are gaining in computational power and flexibility allowing for a paradigm shift in the field of DAQ systems design. Modern web browsers offer the possibility to create intricate user interfaces and are able to process and render complex data. Furthermore, new web standards such as WebSockets that bring the sockets technology to web browsers or WebWorkers that enable web applications to use multi-threading allow for fast real-time communication between the server and the user with minimal overhead. Those improvements make it possible to move the control and monitoring operations from the back-end servers directly to the user and to the front-end electronics, thus reducing the complexity of the data acquisition chain. Moreover, web-based DAQ systems offer greater flexibility, accessibility, and maintainability on the user side than traditional applications which often lack portability and ease of use.

As proof of concept, we implemented a simplified DAQ system on a mid-range Spartan6 FPGA development board (SP601) coupled to a digital front-end readout chip (VFAT2). The system is connected to the internet and can be accessed from any web browser. It is composed of custom VHDL code to control the front-end readout and of a dual soft-core Microblaze processor to communicate with the client.

The first Microblaze core is running an HTTP server that delivers static web content to the client upon connection. It makes use of a Memory File System to store files in the RAM and is able to handle a wide variety of HTTP request. It also sends and receives WebSocket requests to and from the users in order to broadcast data or receive commands. Using the later, clients are able to control the front-end readout chip and get feedback in real-time. Whenever an event is recorded, data can be transmitted to all connected clients and then displayed in their browser. This core acts as monitoring interface.

The second Microblaze core is coupled to the custom VHDL code and handles the communication with the front-end readout chip. It receives commands from the first core through an inter-processor communication system and parses the requests for the VHDL entities. This core acts as control interface.

Using the system described here-above users are able to connect directly to the front-end electronics, effectively by-passing the need for any intermediate server, to control the system and to get real-time monitoring information about it.

We will report on the latest web technologies and developments (WebSockets, WebWorkers, ⋯) that allow for advanced web-based DAQ system design and on the performance of such systems by presenting the results obtained with the system we implemented.

**Primary author:** LENZI, Thomas (Universite Libre de Bruxelles (BE))

**Presenter:** LENZI, Thomas (Universite Libre de Bruxelles (BE))

**Session Classification:** POSTER

**Track Classification:** Systems