JRA1 – Data acquisition system

D. Haas, E. Corrin, M. Pohl[1]

January 6, 2007

Abstract

This memo describes the JRA1 data acquisition system, giving an overview of actual hardware and software developments. Existing readout solutions from IN2P3 Strasbourg are described and an overview of the new readout board from INFN Milano/Ferrara is shown. Also, the current software framework is sketched.

[1] Université de Genève, Geneva, Switzerland

1 Introduction

To profit from the test beam infrastructure designed within the responsibilities of JRA1, a competitive data acquisition system (DAQ) needs to be designed. The DAQ will need to readout and combine data not only from the different planes of the pixel telescope, but eventually also from the device under test (DUT), which will be most of the time another pixel-sensor like device. Also the DAQ needs to handle external information of the test environment, for example run/beam related data. In addition, users of the test beam infrastructure should be able to integrate their DUTs into the existing framework in a simple and easy way, either at the hardware level, or at the software level.
Because the demands to the DAQ are relatively lightweight compared to other experiments, the design of the acquisition system can be held relatively simple. Still, the design needs to be able to scale at a later stage of the development.
Within JRA1, the University of Geneva is coordinating the DAQ efforts, combining hardware from INFN Milano/Ferrara, IN2P3 Strasbourg and University of Bristol together with software input from Universität Bonn/Universität Mannheim and IN2P3 Strasbourg.
In the following, we will present a general overview of the upcoming DAQ, as well as detailed information about the ongoing hardware and software developments.

2 DAQ Overview

To integrate a device under test into the data acquisition of the JRA1 beam telescope, different methods have been evaluated:

1. Integration at hardware level: This needs a special purpose hardware interface that should be able to read out the telescope sensors and the DUT as well. While the final read out board for the telescope implements this possibility, we can probably use this approach only for very dedicated DUTs. Also this approach gives a relatively large overhead to external groups who want to perform initial tests anyway with their own lab equipment for the readout.
2. Integration at software level: The DUTs will provide their own DAQ hardware, but the data will then be treated by common DAQ software. This approach puts most of the development effort on the side of the JRA1 working group and thus can also not be applied all the time. Still we foresee to use this approach eventually at the demonstrator level for dedicated DUTs.

3. Integration at data level: Both the beam telescope and the DUT use their own dedicated readout hardware and software, and the separate data streams are combined by inter process communication of the different DAQ systems. While this looks simple at a first glance, lots of problems can be expected because it is not clear, how to properly synchronize events and configure the different devices during start-up phase.
4. Integration at trigger level (Figure 1): This will be the default scenario for most of the applications. Again completely different hardware and software can be used for the beam telescope and the DUT. The synchronization of the events will then be performed using simple Trigger, Busy and Reset signals and the events will be combined off-line. Run control and configuration are also not necessarily easy in this case, but can usually be well-controlled using Busy signal. As an additional safety measure to avoid slippage of event numbers between DUT and beam telescope, the trigger unit can provide a dedicated event number, that can easily be read out by the DUT as well, thus guaranteeing a perfect match between an event from the telescope and the DUT.

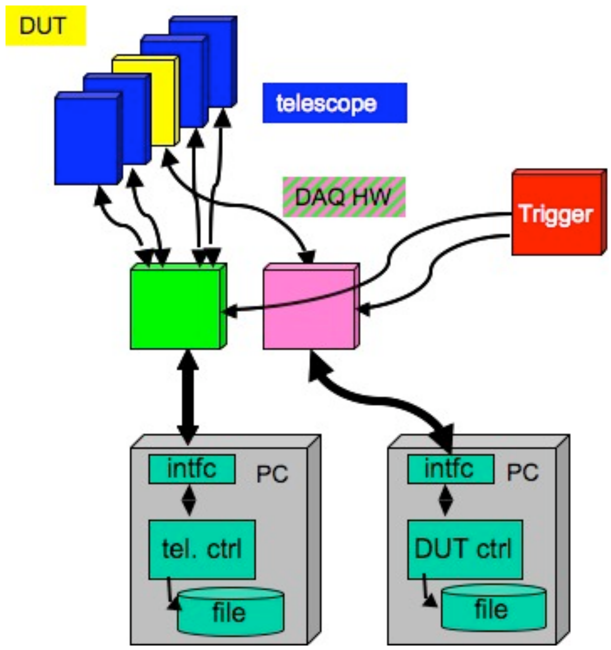

Figure 1: Integration of the DUT into the DAQ at trigger level

The following sections will look in more detail at the necessary hardware and software for the realization of the above scheme.

3 DAQ hardware development

Dedicated hardware for the data acquisition of the beam telescope is currently under development. For early tests, IN2P3 Strasbourg is providing their existing hardware for the readout of the Mimostar 2 and Mimostar 3 pixel detectors, which will be described in 3.1. Based on the experience of the Strasbourg group, INFN Milano/Ferrara is currently developing a more powerful and generic readout board, which will overcome some of the current limitations of the Strasbourg solution (see 3.2). In Section 3.3, we will describe the Trigger Logic Unit (TLU), which has been developed by the University of Bristol to provide the integration of DUTs at the trigger level.

3.1 Existing readout hardware of IN2P3 Strasbourg

To qualify and test MAPS pixel detectors, IN2P3 has developed there own readout system, which is in use successfully since a few years [1]. The pixel detector (e.g. Mimostar2) is directly bonded on a front-end board. This board is then controlled by an imager board that can be read out via a standard USB 2.0 interface (Figure 2).

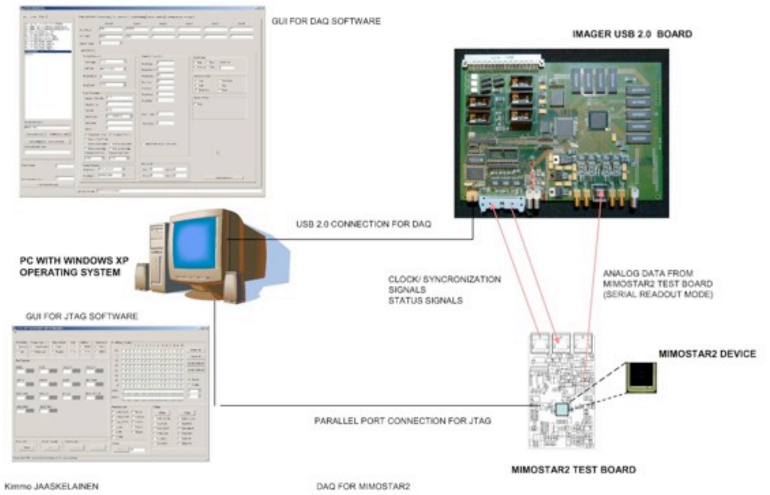

Figure 2: Readout of a Mimostar2 chip via the Imager USB 2.0 board of IN2P3 Strasbourg.

The existing solution has been designed for portable use and for sensor qualification and is good for initial developments of the beam telescope, but for EUDET, it has been decided to go for a new design, based on ideas implemented in the imager board, but also adding the following features:

- A VME bus in addition to USB 2.0 for faster readout speed while taking data without zero-suppression.
- A modular input interface, which can be adapted to different chip technologies using daughter cards, needed to accommodate different DUTs.
- The overall readout speed in the Imager board is limited to about 40 Hz, when reading 6 layers of the telescope in parallel. Currently, the Strasbourg group is implementing zero suppression on their boards. This could boost up the possible event rate to around 100 Hz, for a Mimostar3 maybe even to roughly 300 Hz. The new solution aims to achieve rates of the order of 1 kHz with onboard zero suppression.

For these reasons, INFN Milano/Ferrara has been assigned to design a new readout board for the JRA1 beam telescope, the Eudet Data Reduction Board (EUDRB)

3.2 Eudet Data Reduction Board (EUDRB) of INFN Milano/Ferrara

The EUDRB [2] has the following features:

- 20 MHz readout of 4 parallel input chains
- Altera FPGA running at up to 80 MHz
- SRAM memory with space for 1 million 48bit-long words. This allows the readout of up to 3 frames in succession
- Readout either by USB 2.0 or VME64x, offering maximum flexibility

A block diagram of the EUDRB is shown in Figure 3. The key component of the board is an Altera FPGA that controls the analogue and digital daughter cards, the input and output ports (trigger, RS232, USB 2.0 but mainly VME64x) and has access to the memory for event storage. The Altera can be configured via a microcontroller unit (Altera Nios).

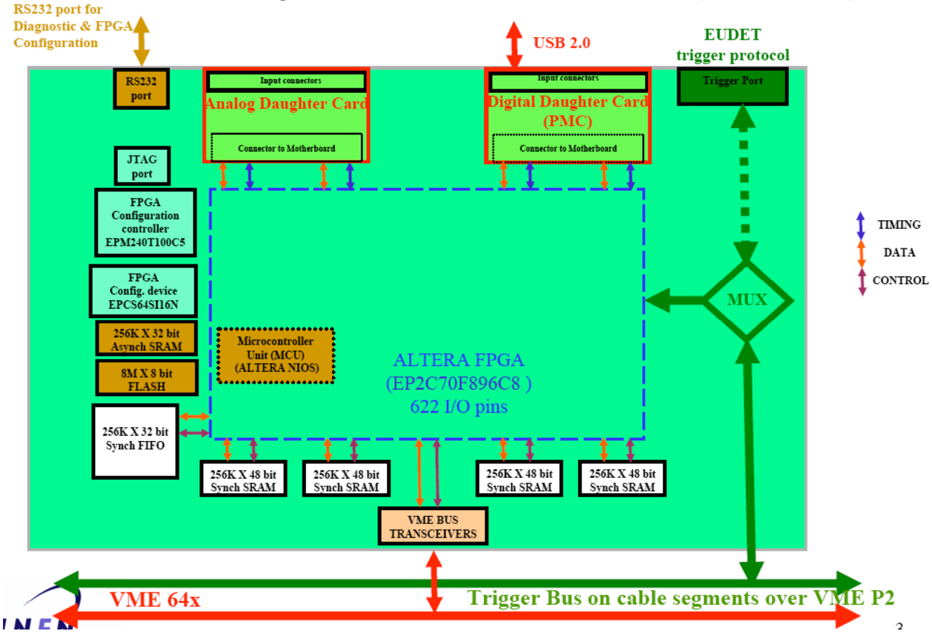

Figure 3: Block diagram of the EUDRB

The EUDRB will support two different readout modes: A zero suppressed readout, thus minimizing the dead-time while in normal data taking and a non-zero suppressed readout of multiple frames. The second option can be applied also to hardware compatible DUTs. A picture of the first prototype and the according daughter boards is shown in Figure 4. The board will be available for integration into the DAQ beginning of 2007.

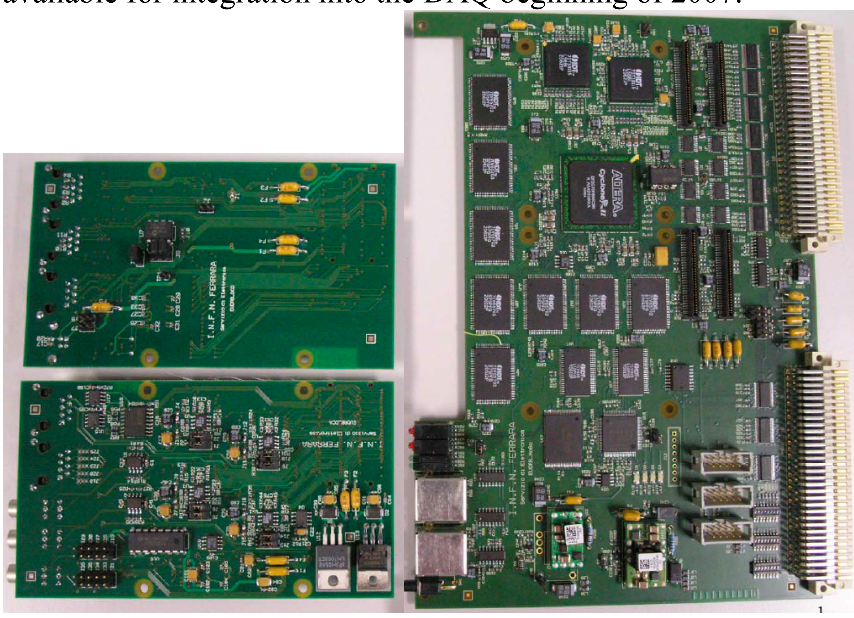

Figure 4: Prototype of the EUDRB – analogue and digital daughter cards (left), motherboard (right)

3.3 Trigger Logic Unit (TLU)

For simple integration of DUTs into the telescope, a trigger logic unit has been developed by the University of Bristol [3]. The unit replaces a 'classic' solution of hand cabled NIM-modules and implements at the same time additional features like an event number and a timestamp to individual events. The design is based on a commercially available breadboard (ZestSC1) from Orange Tree Technologies. The TLU provides the following interface:

- A USB 2.0 connection for configuration and readout of the TLU timestamps and event numbers via a PC
- Four Lemo input connections for the beam trigger (photomultiplier or similar). The input type can be configured via daughter cards, if needed
- Six RJ45 connectors providing trigger, busy and reset signals
- As an alternative two of the six interfaces can be configured as a TTL interface via Lemo connections

In the easiest case of integration, the TLU provides a simple handshake as shown in Figure 5a. On request, the TLU provides a more sophisticated trigger data handshake (Figure 5b)

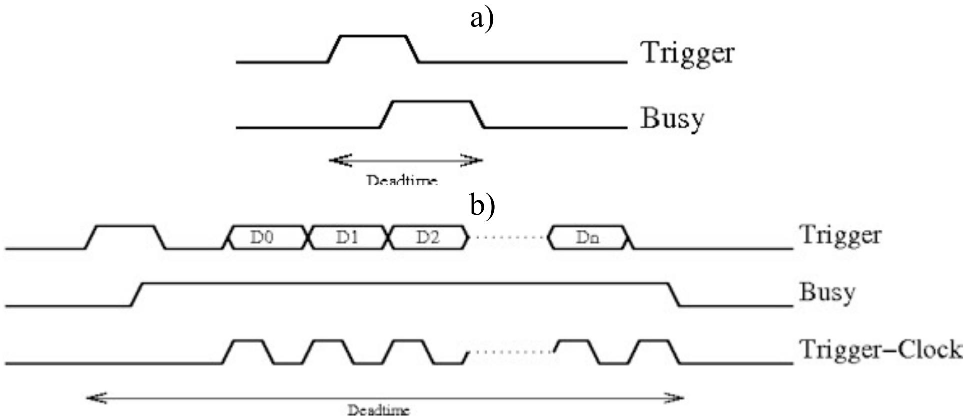

Figure 5: a) Simple handshake: Trigger goes high, waits for busy to go high before releasing trigger and is ready for new triggers when busy goes low. b) Trigger data handshake: As before, but during busy high, the DUT can clock out data from the TLU.

Pictures of the first unit (motherboard and assembled box) are shown in Figure 6. Currently we are already using 2 units in Bristol and Geneva. 3 more units are under production and will be distributed to other groups for test preparations.

Figure 6: The motherboard of the TLU inside the box and the assembled box (without PMT daughter boards)

4 DAQ software development

To quickly achieve results without blocking further developments, the following approach for the DAQ software has been chosen: To develop first a quick solution, combining existing software packages from Universität Bonn/Universität Mannheim, IN2P3 Strasbourg and University of Bristol as a proof of principle. Then, redesign the DAQ software from scratch, using most of the concepts in a generalized form.

4.1 Description of existing software

Before the EUDET project, IN2P3 Strasbourg was already using their own DAQ software to test and qualify their pixel sensors under Windows. Universität Bonn/Universität Mannheim had also their own DAQ for the test of their DEPFET devices, based on a classic producer/file writer approach, also running under Windows (Figure 7). In addition to this, small perl-scripts to access the TLU under Linux have been provided by the University of Bristol.

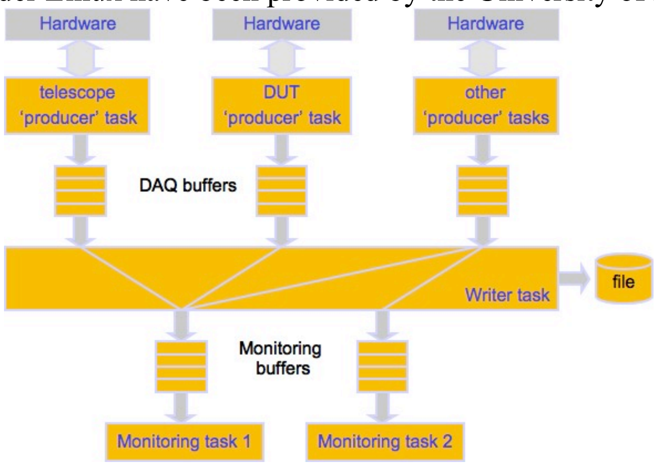

Figure 7: The existing DAQ scheme from the DEPFET group

4.2 JRA1 data acquisition software, version 1

To quickly come to a running DAQ, which is able to readout the JRA1 beam telescope as well as possible DUTs and the TLU, it has been decided to combine the existing DAQs into one. The skeleton from Figure 7 has been used. The Strasbourg code and the TLU readout have been implemented in the form of independent producers. This system is currently up and running, but will be used only as a proof of principle. It lacks some essential functionality, like running on more than one computer, so a clean redesign, based on the above scheme, is currently under study.

4.3 Improvements for DAQ software, version 2

The existing DAQ solution still has disadvantages:

- The communication between the different software packages is only rudimentary; no real exchange of information is taking place.
- The system is not scalable and all processes have to run on the same machine under Windows
- The output format of the raw data still needs to be addressed: We are currently studying, if the use of a common data format like LCIO [1] can be implemented easily into our DAQ system
- The graphical user interface has been developed with a proprietary software (Borland), where the future is unclear
- The existing software is not platform-independent

The new software clearly needs to overcome the first three issues. Platform-independence and the choice of the software suite for the GUI are optional, but will be addressed as well. The DAQ software version 2 will be based on the scheme shown in Figure 8. This scheme enhances the one from Figure 7:

- Global Run Control takes cares of all inter process communication.
- The concept of shared buffers will be extended for data exchange across machines. Thus, producers and file writer processes do not need to run on the same machine and not even under the same operating systems. This will help users of the beam telescope to easily integrate their DAQ, if they wish to do so.

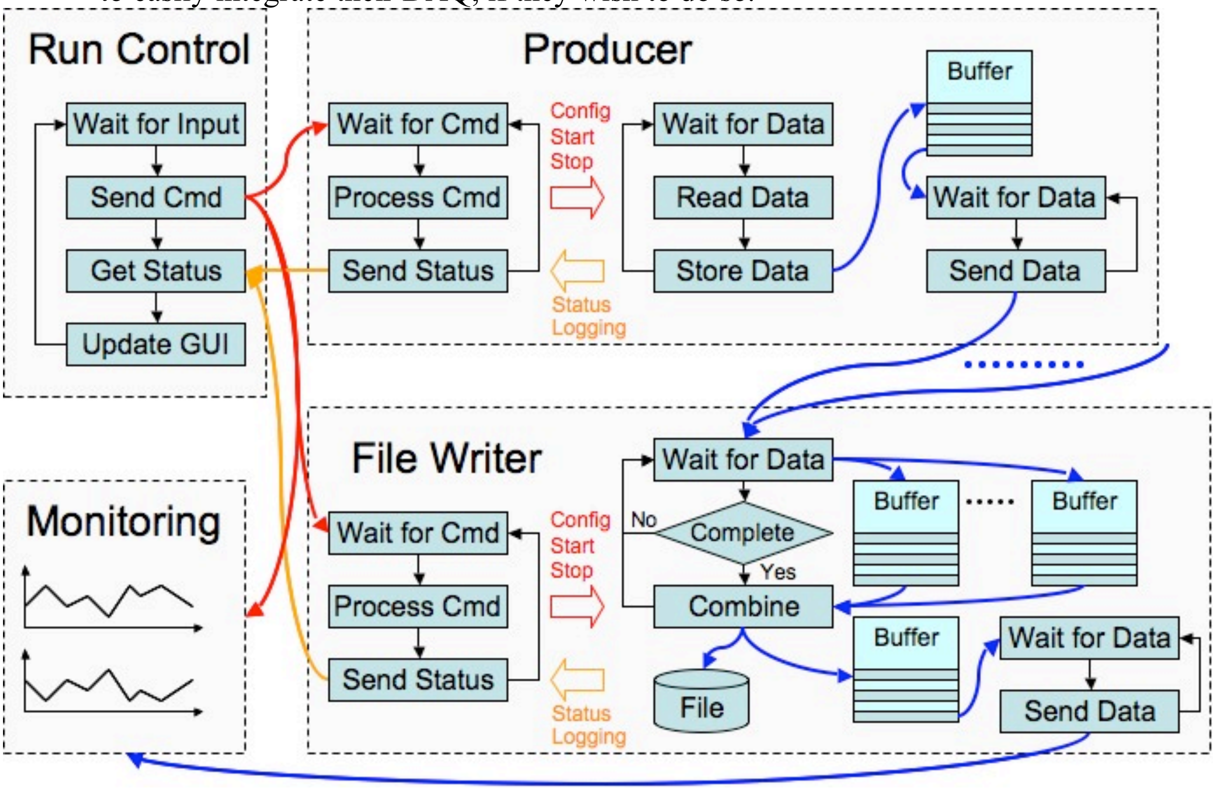

Figure 8: The scheme for the JRA1 data acquisition

A first implementation of the DAQ software version 2 will be ready for the demonstrator pixel telescope in July 2007.

5 Future developments

The demonstrator telescope needs to be ready in July 2007, including a demonstrator DAQ system. To fulfil the tight schedule, pre-integration of software and hardware will start in January 2007 at the Geneva University, including a first EUDRB-card with a VME-readout, a TLU and first software versions. In early March, a front-end sensor will be added to the system and in April a second EUDRB will be added to the system. End of April, it is foreseen to ship the fully integrated DAQ system to DESY for global integration with the telescope hardware.

In parallel, the different DAQ working groups of EUDET are discussing to find common aspects in the design and try to harmonize their efforts. As a first success, we could convince the JRA3 group (responsible for the calorimeter) to use the TLU designed by Bristol. JRA2 proposed some modifications of this unit, so that they could profit from the development as well. On the software side, we plan a workshop in January to see where we could implement common concepts (for example, the use of LCIO [4] as data format, but also a possible standardization of interprocess communication).

Acknowledgement

This work is supported by the Commission of the European Communities under the 6th Framework Programme "Structuring the European Research Area", contract number RII3-026126.

References

[1] Kimmo Jaaskelainen, Imager 12-bit digitizer-controller card with USB2.0 connection, Gilles Claus, Imager board DAQ software overview, CMOS DAQ Application Reference Manual
[2] A. Cotta-Ramusino et al., Status of the EUDRB VME-based DAQ system for pixel sensors, EUDET-Memo to be assigned
[3] Eudet-Memo-2007-02: D. Cussans, Description of the JRA1 Trigger Logic Unit
[4] http://lcio.desy.de