EUDET-Report-2008-06

Status of the data acquisition system for the EUDET calorimeters

V. Bartsch (UCL), M. Wing (UCL), T. Wu (RHUL)

for the CALICE-UK DAQ groups: Cambridge, Manchester, RHUL and UCL

January 28, 2009

Abstract

A data acquisition (DAQ) system is described which will be used for the next generation of prototype calorimeters using particle flow algorithms for the International Linear Collider (ILC). The design is sufficiently generic and scalable such that it should have numerous applications either for ILC detectors or elsewhere within high energy physics in general. The DAQ system will be implemented using FPGAs and built using off-the-shelf components and networking hardware with programmable FPGAs. The software for the DAQ system is based on an existing framework, DOOCS, which is a server/client object-oriented system. The design philosophy, current status of the project (including significant updates since the EUDET annual meeting) and its aims are presented in this report.

1 Motivation of the data acquisition system

The concept of particle flow algorithms (PFA) and the beam structure of the planned International Linear Collider (ILC) accelerator put constraints on the data acquisition (DAQ) system which are described in the following. PFA is a widely accepted approach to improve the energy resolution at the ILC. Particle flow uses the high segmentation of about 1 cm×1 cm (or smaller) in the detection layer of a sampling calorimeter to track particles through the calorimeter. The segmentation results in about 24 million readout pads in the electromagnetic calorimeter (ECAL) of a planned ILC detector [1]. It is estimated that about 250 GB of raw data per bunch train need to be handled depending on the very front end electronics (VFE) and the detector type chosen for the final calorimeter. Because of the large number of readout channels it is necessary to minimise the cost of the electronics. This is achieved by using standard networking chipsets and protocols. Already at an early stage of the calorimeter prototypes the DAQ system has been designed for scalability. In the current detector plans the space for the electronics and cooling inside the calorimeters is very restricted. In the EUDET prototype the DAQ components are not yet miniaturised to allow some flexibility for the test beams. The DAQ system under development for the EUDET programme is aiming at a generic DAQ system, which will be applicable for the ECAL, analogue hadronic calorimeter (AHCAL) and digital hadron calorimeter (DHCAL) prototypes within the EUDET project. At the ILC it is assumed that all data will be collected with no triggers implemented. During a bunch train, data will be stored on the detector and then read out during the inter-train gap of about 200 ms. The funnel-like DAQ then wraps and transmits all the data over network and finally saves all data into central storage for later analysis. The existing system components, based on this generic design, have been built within the EUDET framework. The aim is to complete a DAQ system for the EUDET test beam in 2009.

2 Introduction to the DAQ architecture

The data are taken by the detectors on the detection slabs and digitised by the very front-end on-detector electronics or slabs. Then all output data are delivered to electronics boards at the end of the detector. Data is then sent to a concentrator card which collects data from many detector units before being sent off the detector on Gigabit optical fibres to a detector receiver in the counting room. Figure 1 gives an overview over the generic DAQ scheme, whereas Fig. 2 shows how this scheme fits into the EUDET prototype under development.

The calorimeter is designed such that VFE chips are embedded in the detector slabs. The slabs are served from the ends with a Detector Interface (DIF). The DIFs transmit the data to the Link-Data Aggregator (LDA) which is fitted on-detector and serves many

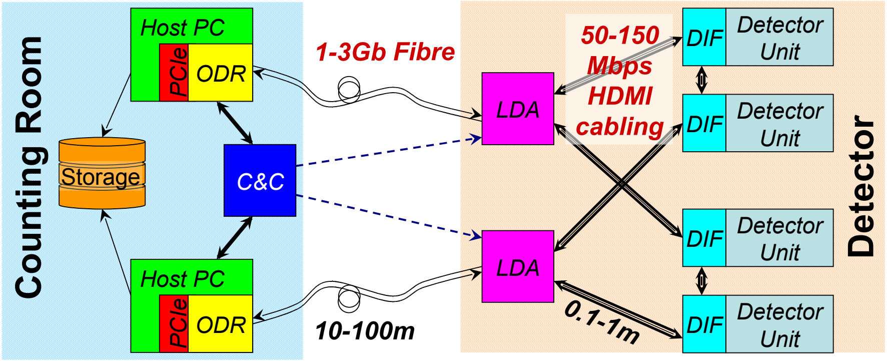


Figure 1: An overview of DAQ architecture.

DIFs via standard HDMI links. The data from the LDAs are fed to the Off-Detector Receiver (ODR) via the Gigabit optical cables. The ODR is a dedicated PCI-Express card which is hosted in a DAQ PC, being capable of saving the data to central storage.

2.1 Detector Interface Board

The detector signals are buffered and digitised by the VFE ASICs situated on the detector slabs. The VFE ASICs depend on the detector used in the prototypes accommodating the different features of each detectors. For the EUDET prototype three different detector types, ECAL, AHCAL and DHCAL require a customised VFE. The interface between the VFE and the DAQ system is provided by the DIF board which is located at the end of the calorimeter slabs. The DIF board consists of a customised part which translates the signals of the ASICs into more generic signals which are independent of the detector used for the calorimeter. The interface has been defined together with the three detector groups and is common to the three different detector types. A generic part of the DIF passes the data on and sends configuration on to the VFE electronics. Currently three ECAL DIF boards exist, the picture of one of them is shown at Fig. 3. The link between the DIFs and the link data aggregator (LDA) has been tested. The DIF is connected via HDMI cabling to the rest of the DAQ system. HDMI [2] is a home entertainment system standard with small cabling and connectors which are commercially available at low cost. They are rated at more than 300 Mb/sec for data. To provide redundancy against link failures, a provision is made on the DIF to connect to its neighbour. By connecting even and odd DIFs to different LDAs, the system is protected for the loss of a DIF-LDA link, or even the failure of an entire LDA.

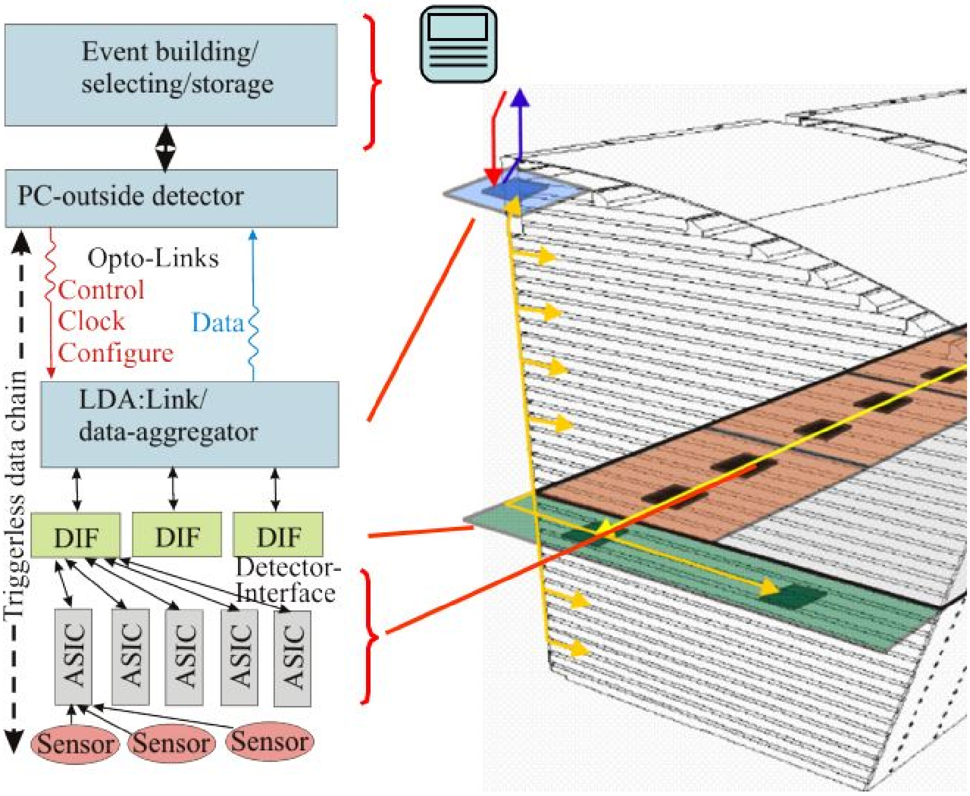


Figure 2: Integration of the DAQ components in the EUDET module.

2.2 Link Data Aggregator

In order to accumulate the data sent from the DIF, an LDA collects the data from several DIFs and sends it further to the ODR. The number of DIFs aggregated in the LDA depends on the detector type chosen in the detection layer. The optimal number of DIFs/LDA should take into account the number of available pins on the FPGA, the bandwidth per DIF and the effective performance-price ratio to maximise bandwidth of the optical link. The current LDA prototype can host up to eight links to the DIFs with a small upgrade up to ten DIF connections. The LDA has banks of HDMI connectors for connections to the DIFs and a small form-factor pluggable (SFP) connector for the optical link off-detector. The prototype version of the LDA is built on a commercial development board with a Xilinx Spartan3-200 FPGA. The two sets of connectors are physically add-on boards: one providing the SFP and serialiser chipset for the optical link (see Fig. 4(a)), the other hosting eight working HDMI connectors with clock fan-out hardware (see Fig. 4(b)). A first prototype of the LDA has been produced by the company Enterpoint [3] and has been thoroughly tested. After several improvements the producer of the LDA board has delivered a working board. The ethernet connections of the board which posed problems in the past have been successfully tested and are working. The interface to the DIF is currently being debugged and firmware for the

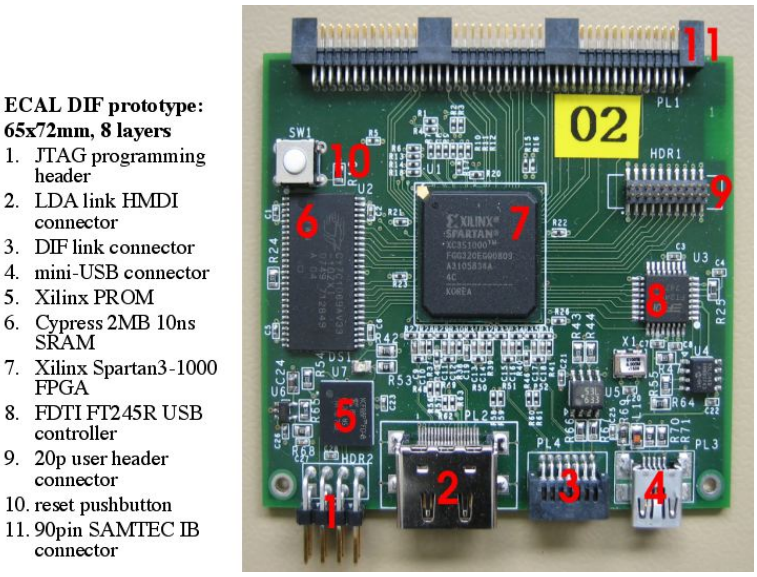

ECAL DIF prototype:
65x72mm, 8 layers
1. JTAG programming header
2. LDA link HMDI connector
3. DIF link connector
4. mini-USB connector
5. Xilinx PROM
6. Cypress 2MB 10ns SRAM
7. Xilinx Spartan3-1000 FPGA
8. FDTI FT245R USB controller
9. 20p user header connector
10. reset pushbutton
11. 90pin SAMTEC IB connector
02
1
2
3
4
5
6
7
8
9
10
11

Figure 3: Picture of a DIF.

processing of the packets received by the ODR will be implemented soon.

2.3 Off Detector Receiver

The LDA is situated at the edge of the detector, the ODR however is located in the counting room connected to the on-detector DAQ system by an optical fibre. In the current prototype the optical link between the LDA and the ODR will use a 1G ethernet connection. As well as receiving data from the detector, the ODR sends control and configuration data to the LDA for distribution to the DIFs and finally sends the data off the detector to the event building. The ODR is shown in Fig. 5. It is realised as a PCI-Express card and can serve up to four LDAs per card, with one PC hosting up to two ODRs. For the proposed DAQ system currently a Xilinx Virtex 4 FPGA is used, thereby using a commercial FPGA board designed by PLDA [4]. At present, the data stream which will come from the detector in the EUDET test beam is simulated by an internal data generator in the firmware or via an on-board ethernet interface; this allows for a thorough debugging and optimisation of the system before integration with the detectors.

The user interface to the ODR card contains two parts: a customised driver and a client program. The former is mainly tasked with mapping card memory to the user space and providing direct memory access (DMA) support. The latter client retrieves data from the ODR card memory and stores it on the local disk. The performance of the prototype ODR has been investigated using an ethernet interface to provide an input data stream which is copied to host memory and is currently higher than 150 MB/s writing to an

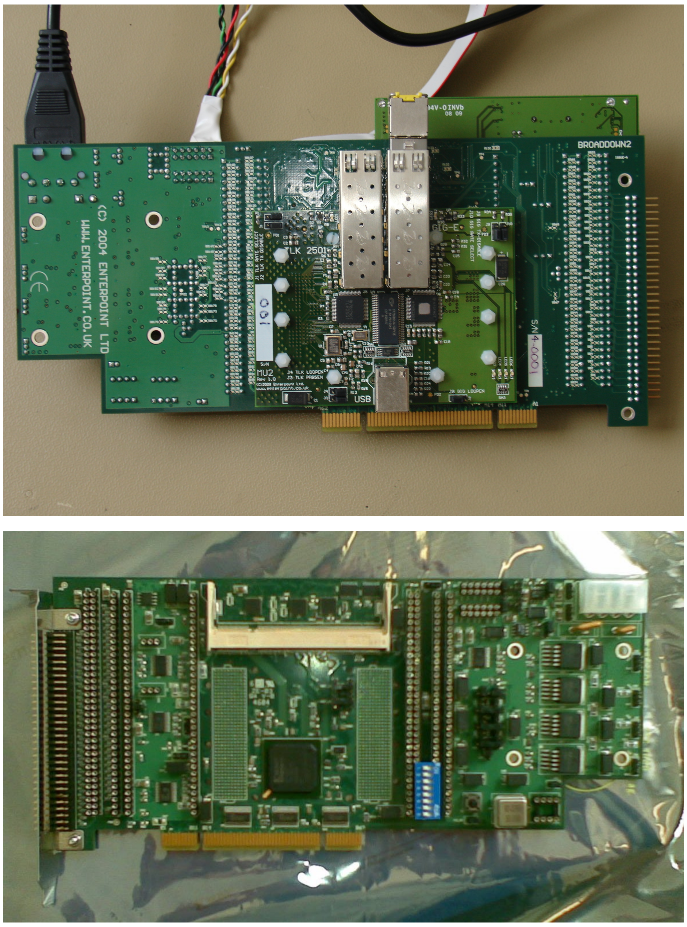


Figure 4: Pictures of both sides of the LDA: The top picture shows FPGA (in the middle of the board) and the add-on board HDMI connectors with the clock fan-out hardware (at the left of the board). The bottom picture shows the other side of the LDA board hosting the add-on board providing the SFP and serialiser chipset for the optical link (in the middle of the board).

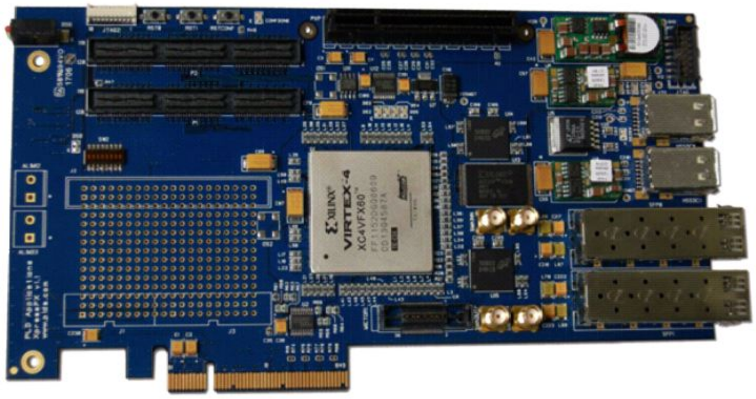


Figure 5: Picture of an ODR.

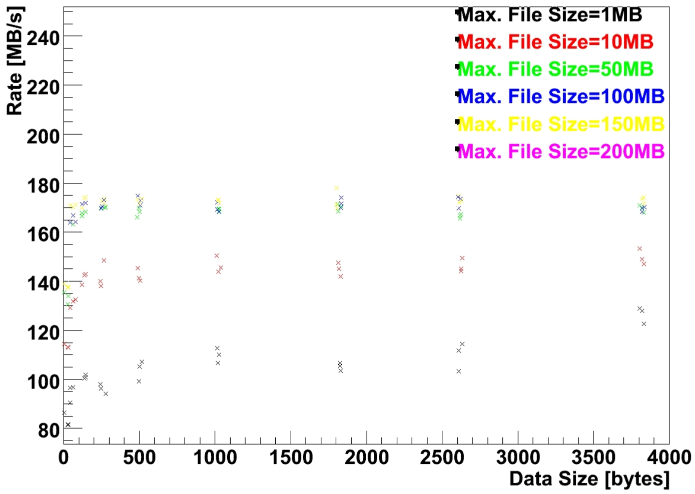


Figure 6: Data transfer rates of the off detector receiver depending on the data size.

array of disks as shown in Fig. 6. A higher data rate of up to 700 MB/s can be achieved when writing into the host memory. Thus the data rate to disk seems sufficient for the prototype which will use a 1G ethernet connection to the LDA. The number of DMAs has been optimised for the performance as illustrated in Fig. 6. The ODR prototype is ready and can be optimised further in the next months.

3 Clock and control handling

For the event building a good clock is essential because the events will be built using time stamps. It is understood that a machine clock will be fed into the ODRs and fanned out

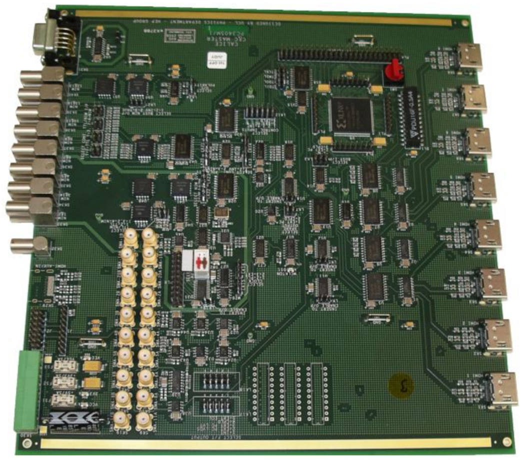


Figure 7: Picture of the clock and control board.

to the LDAs and DIFs. The requirements on the clock are a low jitter and a fixed latency between the machine clock and the clock in the DIFs. Commercial networking hardware is not suited for this task as it is most efficient when it can buffer data and provides no guarantees on delivery times. Similarly networking hardware built into modern FPGAs suffers from varying latency. Therefore a clock and control (C&C) board has been designed with links to eight ODRs (see Fig. 7). The C&C logic is implemented on a board with 6U format, translating into a board size of 100mm × 160 mm.

The clock and control module must interface with the machine and provide stand-alone signal and clock generation. The machine clock is expected to run between 50 and 100 MHz. The bunch clock will be derived as a multiple of the machine clock. It is able to deliver fast asynchronous triggers. It will also receive a busy signal from the VFE. The HDMI cables used for the DIF to LDA link will be reused here for sending control data. The C&C board has been delivered and has been successfully tested. In parallel the firmware for the C&C board is being developed.

4 DAQ software

4.1 DOOCS architecture

The most important requirement on the DAQ software for the calorimeters is its scalability. Already for the EUDET test beam the number of devices which need to be served by the software is in the order of a thousand devices including the VFE. Therefore it became clear at an early stage of the development that a software framework will be used so that the measures for reliability and scalability which already have been tested in a software framework can be reused. This approach is somewhat different from the software approach of the JRA1 and JRA2 activities. JRA1 and JRA2 do not need a software which is as scalable as ours. Therefore within JRA1 framework the EUDAQ software has been developed and is being reused by the JRA2 activity [5, 6]. During the calorimeter software development history, several software candidates have been surveyed, including Experimental Physics and Industrial Control System (EPICS), Adaptive Communication Environment (ACE) and Distributed Object Oriented Control System (DOOCS). Functionality, scalability and flexibility of the software are the main requirements for our DAQ software; DOOCS was found to be suitable for our DAQ needs. An overview of the DOOCS functionality is shown in Fig. 8.

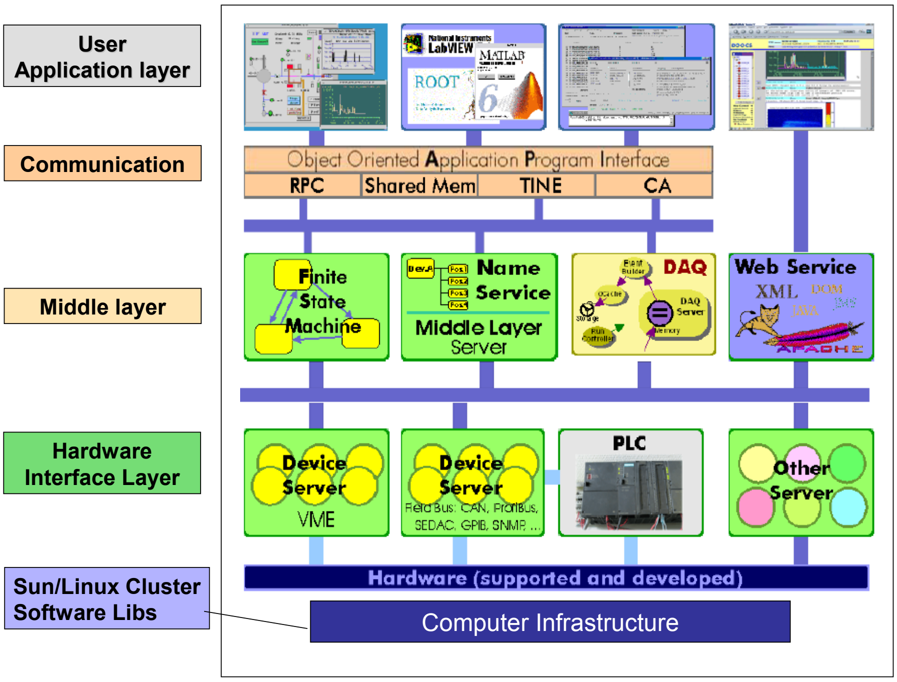


Figure 8: A schematic overview of DOOCS.

DOOCS was designed for HERA and TTF control applications and also developed for XFEL at DESY [7]. It is a server/client and object-oriented system, with communications capable of handling the conversion of data structures between the client and server processes. The whole system is written in C++ and runs on several operating systems including Sun, Unix and Linux workstations.

Fig. 8 shows that, on the computer infrastructure level, class libraries have been built into blocks for device servers, system programs and communication protocols, which support clusters with different platforms. Many types of hardware are supported via a hardware interface layer, including VME, ProfiBus, FieldBus, Sedac and CAN etc.. New hardware interfaces, so called device servers, can be easily added to the existing DOOCS framework. This feature has been used in our software project since the devices used in our

DAQ system require customised interfaces. The middle layer consists of State Machine, Naming Server, DAQ Server and Web Server with Java, XML and Apache/Tomcat, etc.. The upper layer is the communication level. DOOCS provides several types of communication protocols, including Remote Procedure Calls (RPC), Shared-Memory, Channel-Access (CA used by EPICS) and Three-fold Integrated Network Environment (TINE). The default communication is realised by RPC. In case RPC is not suitable for a specific project, the user can choose from the three additional communication protocols. The type selection of the communication protocols is transparent for the user. The top layer provides user APIs, such as the DOOCS Data Display (DDD) tool, LabView, MATLAB, ROOT, high level application and utilities like the RPC GUI Utility, and a web browser supporting E-logbook, alarm and information display components.

4.2 Current EUDET DAQ software

To adapt the DOOCS software framework to our DAQ software the Equipment Name Server (ENS) has been used to integrate all DAQ components into one system. ENS provides services for naming resolution and RPC communication. The ENS RPCs resolve the names of the devices used in the control system with one entry per server. The naming structure is defined to consist of the Facility, in other words the experiment the devices are located in, the Device, i.e. the device type, the Location, i.e. the physical device and the Property. In our software the facility names are defined as CALICE.ECAL, CALICE.AHCAL and CALICE.DHCAL. The devices are defined as ODR, LDA, DIF and ASIC according to the types of current DAQ components. Based on the object-orientation concepts, each type of device is modelled and served by a device server running in the DAQ PC. Every device server can handle many instances with specific attributes called Property in DOOCS. A client program runs independently which sends control messages to device servers and receives feedback from them. Whenever a device server is activated, new device instances are created and corresponding properties are added respectively. All the properties for each device instance have been customised beforehand. Once any change is made for instance to the properties operation, the changes are immediately effective and available over the network. All the queries and communications are resolved by ENS via RPC. DOOCS communication and RPC Plot Utility tools can be used to display such an ENS server with Facility/Device/Location/Property structure layers.

In order to implement our DAQ system within such a model, all the functionalities and properties for each device of the DAQ components have to be classified and defined. As seen in Fig. 2, the ODR is the first hardware layer which can communicate with DOOCS. This is shown schematically in Fig. 9. Furthermore, a prototype of the device is completely ready. Therefore the software for the ODR has been the starting point of the DAQ software development. An LDA emulator is used instead of the real device, which runs in an independent PC to connect to the ODR card via an optical link. The emulator can generate fake data and sends it off to ODR device if requested.

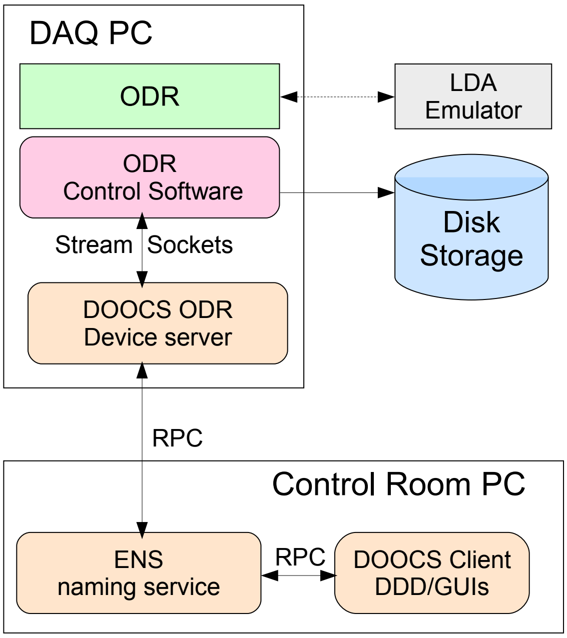
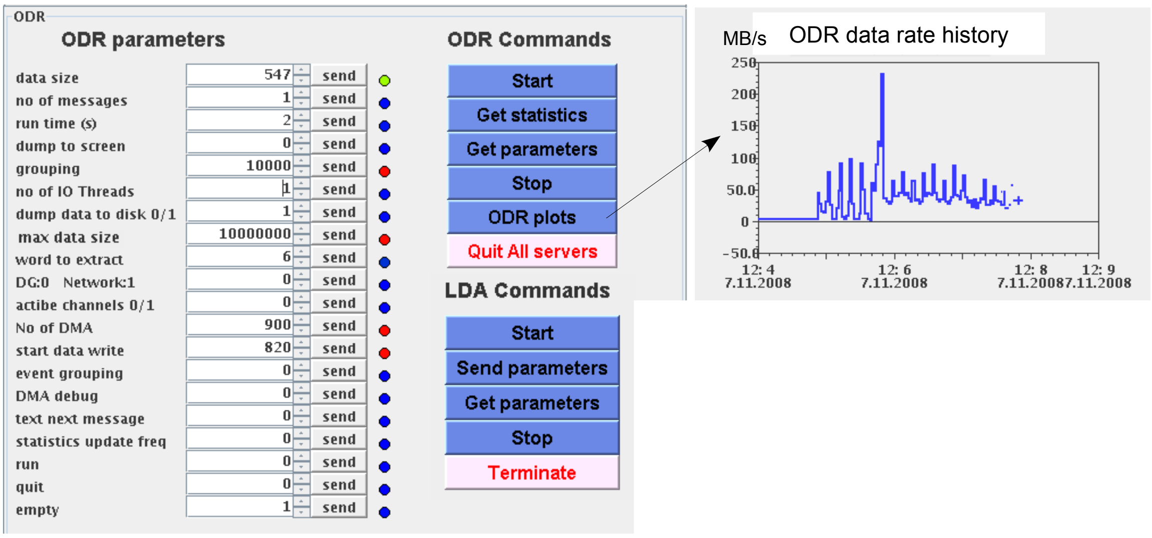


Figure 9: A software flow chart for current DAQ demonstrator.

The ODR card is installed in the dedicated DAQ PC with a custom driver. The ODR control software manages the user-to-hardware interface, transmitting data from the ODR card memory to the DAQ PC host memory. It supports communications to and from the LDA via control messages and the I/O of local data storage. The communication between ODR control software and DOOCS device server is provided via stream sockets. On the client side, the DOOCS clients GUI runs within a PC located in a control room, in which ENS server runs. Users can send control messages to the ODR device server and receive feedback through the ENS server via the RPC channel.

As shown in Fig. 10, the DOOCS client GUI has been designed with a list of commands both for the ODR and the LDA, and configurable parameters have been customised for ODR device. Several types of data need to be handled correctly. Configuration data and files are used when the system starts and runs. The ODR control software reads the configuration parameters during the start up. The ODR device server has static configuration parameters like the host name or the port number used for stream sockets connections. Some volatile parameters for the ODR instance properties can be configured during start up and should be changeable when running, like the control command type. The ENS server reads the configuration file, in which the naming convention for device and server, the protocol type, host name and RPC numbers are customised.



ODR parameters

data size	547	send	🟢
no of messages	1	send	🔵
run time (s)	2	send	🔵
dump to screen	0	send	🔵
grouping	10000	send	🔴
no of IO Threads	1	send	🔵
dump data to disk 0/1	1	send	🔵
max data size	10000000	send	🔴
word to extract	6	send	🔵
DG:0 Network:1	0	send	🔵
actibe channels 0/1	0	send	🔵
No of DMA	900	send	🔴
start data write	820	send	🔴
event grouping	0	send	🔵
DMA debug	0	send	🔵
text next message	0	send	🔵
statistics update freq	0	send	🔵
run	0	send	🔵
quit	0	send	🔵
empty	1	send	🔵

Figure 10: DOOCS client GUI with example histograms of data rates.

4.3 Further developments of the DAQ software

To track the file storage a MySQL database is currently being developed. The database keeps track of the file locations and the configuration of the devices during data taking. Because the final DAQ system will consist of many components a device database will do the book keeping on the connections between the devices. It has been suggested that the event building will not be implemented for our DAQ demonstrator as it will be left for subdetector groups. Instead the DAQ software will save all the data from the host memory of DAQ PC to a local disk in a raw format, as sketched in Fig. 9. A framework will be provided in which the detector groups can plug in their event building and LCIO conversion. The EUDAQ framework for the LCIO conversion can be reused here. Different scenarios for data storage have been looked at, e.g. using RAID arrays to provide a fast data-to-disk rate. No choice has been made yet, though.

In addition to the existing ODR and LDA software interfaces or device servers all other DAQ components (DIF, ASICS and clock) have to be interfaced as soon as the devices are ready to be tested with the DAQ software.

To allow operation in test beam mode the whole system needs to transit from an idle state to a data taking state. This will be done by a state machine which starts, checks and configures all hardware components. The DOOCS software could also incorporate other detectors into its framework in the case of their use in combined test-beam running. This would be the ideal solution if simultaneous test beam data is taken with the components from JRA1 and JRA2. An alternative would be to have the state machine activated by a light, overall DAQ infrastructure which still needs to be defined although this adding of an extra layer is not as neat.

To run a reliable system errors need to be identified automatically, they need to be logged and monitored. Therefore the error and alarm handling provided by the DOOCS framework will be used for the EUDET prototype.

5 Conclusion and outlook

The DAQ system for the EUDET detector will be a hierarchical DAQ system which minimises the space needed on the detector and uses commercial components to minimise the cost of the DAQ system. In order to make the development for the test system independent of the detector type tested a DIF board will make the readout generic. An LDA card will concentrate the data from the DIF and send it off to the ODR which is located off the detector in a control room. The task of the ODR is to store the data. All components have in addition the task to send configuration data and control messages to the detector and all other DAQ components upstream. At the time of writing each hardware component of the DAQ system exists and has been tested; the individual components now need to be integrated into a complete system. Current tests have shown that the DAQ software is able to integrate the ODR into the DAQ system based on the existing software framework DOOCS and manage successful communication and control tests. The goal is to have the whole DAQ system ready for the EUDET test beam in 2009.

References

[1] TESLA Technical Design Report, Part IV, A Detector for TESLA, T. Behnke et al., 2001

[2] HDMI- High Definition Multimedia Interface Consortium homepage: `http://www.hdmi.org`

[3] Enterpoint, FPGA and ASIC Design, homepage: `http://www.enterpoint.co.uk`

[4] PLDA homepage: `http://www.plda.com/index.php`

[5] D. Hass and E. Corrin, EUDET-Memo-2008-21, JRA1 Progress Report: The data acquisition framework

[6] X. Janssen et al., EUDET-Memo-2008-25, Status of the Data Acquisition System and Slow Control for the JRA2 TPC test beam infrastructure

[7] DOOCS, Distributed Object Oriented Control System homepage: `http://tesla.desy.de/doocs/doocs.html` Linear Collider".