Overview of the DAQ system

The TPC readout electronics is based on the system developed and used in the ALICE TPC system. It consists of preamplifiers, the Front End Cards (FEC) with the ALTRO digitization chip. The FECs are read using a Readout Control Unit (RCU). The FECs and RCU are connected with a common backplane. The RCU hosts the Source Interface Unit (SIU) and the Detector Control System (DCS). The Detector Data Link (DDL) provides an optical connection between the SIU and the DAQ, whereas the DCS is used for detector control and has an ethernet interface. Parts of the DCS functionality can instead be handled by the DDL. To the RCU is distributed the trigger and clock signals. The Trigger Timing Control (TTC) system used in ALICE can not be used in our case. Further, the FEC must be modified to use a different preamplifier chip.

The readout in ALICE is implemented in a package called DATE, which contains the DAQ, detector control, monitoring, data and message logging. It may be possible to use parts of the software for our purpose. The hardware consists of the DRORC, a PCI-64bit module. The DRORC has two Destination Interface Units (DIU), i.e. two DDLs can be handled by one DRORC. A schematic diagram is shown in Fig.1.

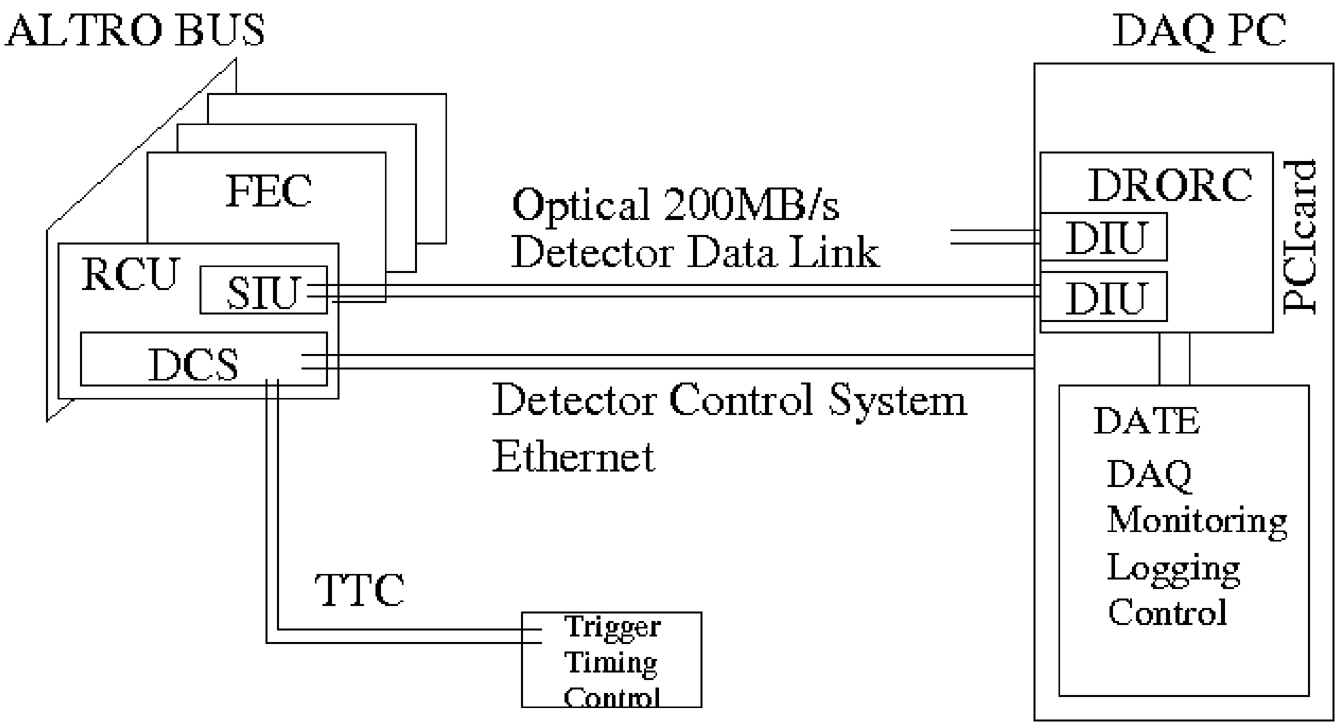

Figure 1: Schematic diagram of the ALICE TPC readout system

Trigger, current (non)understanding

The trigger signal to the readout electronics is distributed from the JRA1 Trigger Logic Unit (TLU), see Fig.2. This unit can as input take four LEMO connectors for photomultiplier input pulses of negative amplitudes. The inputs have individual thresholds, set from the front panel with a trimmer. The input can also handle NIM-signals as inputs.

The pattern of AND/OR of the inputs is programmed via a USB interface. The USB interface is also used to monitor the TLU. The TLU stores the trigger history in a buffer (a time stamp for each trigger and trigger counter), which can be read via the USB interface.

The trigger signals are distributed differentially through RJ45 connectors (LVDS):
Trigger
Busy
Reset
Data-clock

There are six connectors, i.e. up to six subdetectors can be connected.

There is a simple handshake between the TLU and Device Under Test (DUT) including the following steps:

1. TLU receives trigger from PMTs
2. TLS asserts trigger
3. The DUT asserts BUSY
4. TLU de-asserts trigger, the trigger line is then an output of a shift register holding the trigger number.
5. the DUT clocks the data from the shift register
6. When the DUT is ready to receive new triggers it de-asserts busy.
7. System is ready for new trigger

Two out of the six can be connected to LEMO outputs instead, but only the trigger , busy, and reset. The trigger number is not available.

In order to synchronize the DUT with other sub detectors, e.g. the beam telescope, it is necessary to be able to read the trigger number from the TLU. This should be done from the hardware, and inserted into the data stream. One may also choose not to read the trigger number. In our system they all require hardware/firmware modifications of the ALICE TPC system of FEC/RCU to be used.

- Reading the trigger number by modifying a RCU, or making a module that can be connected to a DRORC.
- Making a module that looks like a FEC from the RCU view, with the only task to register the trigger number.
- Use the alternative not to read the trigger number and trust that the triggers do not get out of phase.
- In all cases the trigger must be distributed to all RCUs, and the BUSY sent back to the TLU.

It must be studied how an interface to the TLU can be realized and if we need any modifications to the firmware in the TLU.

Link to a description of the TLU:
http://svn.phy.bris.ac.uk/svn/uob-hep-pc017a/trunk/www/index.html

The DAQ

The question is whether the DAQ system should be based on the ALICE DATE system and in that case, which parts of it can be used? DATE is a rather complex package, and it needs to be modified to interface to the proposed EUDET Common DAQ. The support from ALICE, if any, will be at a very low level. It might be more flexible, and easier to make a very simple system that do what we want. The only part used from the ALICE software would then be the drivers and the Application Programmers Interfaces (API). A schematic diagram of the DAQ system is shown in Fig. 2. The software modules to be implemented are:

- *READOUT*, the part that talks directly with the readout hardware and which must reside on the local DAQ machine. It is controlled from either a local or common Run Control. This communication is done through the network. If it is possible the run control interface protocol should be the same for the common and local DAQ. The DAQ has the options to log the data on a local disk, or to send the events through the network to the common DAQ for logging and monitoring. Other network destinations may be a local monitoring or a data recorder on another machine, in which cases one should be able to choose to send only a fraction of the events. Messages/status is logged to the message logger, also through the network.
- *DCS* is the part that talks with the hardware for setting up configurations, pedestals, etc.. It can be on a different machine, but if the DCS unit on the RCU is not used, then it must reside on the same machine as the DRORC. It gets the instructions from the detector control through the network. Messages/status is logged to the message logger through the network. Some information may also be needed by the local monitor.
- *DATA RECORDER*. This task is responsible for writing data to disk during local running. It should be called either directly via an API call for logging on the same machine, or be accessible through the network.
- *MESSAGE LOGGER*, is writing delivered messages from the different tasks to disk. Can only be accessed from the network.
- *RUN CONTROL*. This is the human interface to start/stop/pause/continue/configure a run. Should if possible use the same network protocol as the common DAQ.
- *DETECTOR CONTROL*. This is the human interface to configure the detector. This is specific to the detector under test.
- *LOCAL MONITOR*. It requests events from the DAQ and makes histograms/event displays to be viewed.
- *DATABASE*. Configurations must be stored to and fetched from somewhere, either as ASCII files or in a proper database (MySql).

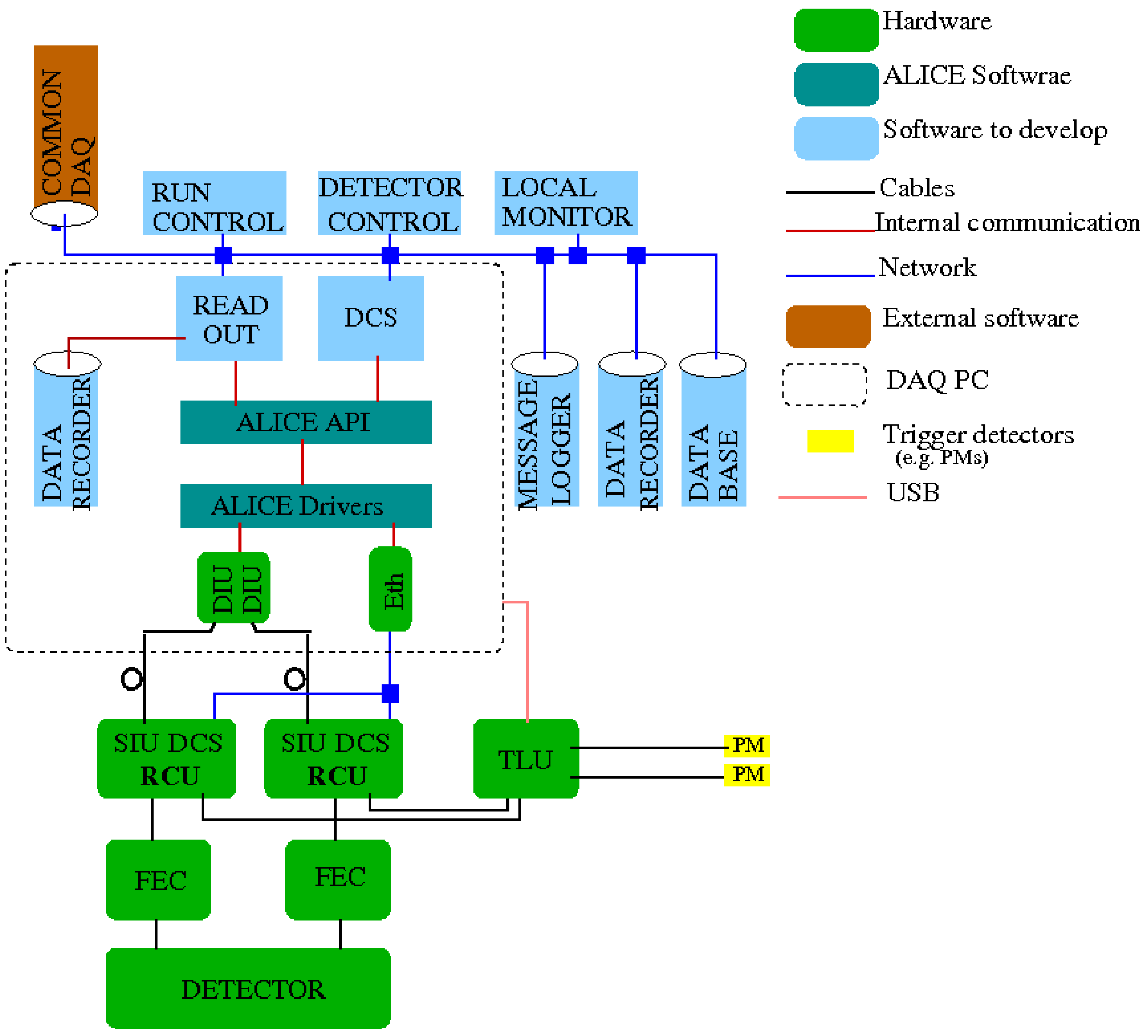

Figure 2: A schematic view of a possible design of the local DAQ

Tasklist

A possible division of tasks in the case of only using the low level ALICE parts.

Item	***Task***	***Who***
1	Installation and test of ALICE drivers/API for DRORC/RCU, and PC configuration. The configuration is needed to define the hardware of the PC, memory banks etc.	Lund/Bonn/?
2	Installation and test of ALICE U2F/DATE. This is to understand if/what parts of DATE that can be used.	Bonn
3	a) DAQ: Write a rudimentary readout of the DRORC/RCU using the ALICE drivers/API, and PC configuration. b) Interface to RUN CONTROL	Lund/Bonn/?
4	a) DATA RECORDER b) MESSAGE LOGGER	Lund/Bonn/?
5	a) RUN CONTROL b) DETECTOR CONTROL	Lund/Bonn/?
6	DSC	Lund/Bonn/?
7	a) LOCAL MONITOR interfaces b) Histogramming/event display etc...	Lund/Bonn/?
8	DATABASE	Lund/Bonn/?
9	RCU modifications for the TLU trigger number	Lund/Bonn/CERN/?
10	FEC: modifications for the new preamplifier chip	Lund/CERN
11	Modification of DATE, if to be used.	Lund/Bonn/?

If we decide to make our own system, then one should keeps things as simple as possible. There is no need to make a complete general system. One should investigate the possible use of common DAQ parts also for local nondetector specific parts (e.g. items 3b,4,5a,7a,8).

Hardware

The table shows what is needed for in total 10000 channels. What has been already ordered is sufficient for 2000 channels. The already ordered items are included in the needed numbers given in the table.

Item	*Cost (SFr)*	*Ordered*	*Total need*	*Total cost (SFr)*	*Delivery time*
DRORC	2400	1	2+1 spare	7200	?
RCU	350	1	4+1 spare	1750	?
SIU	950	1	4+1 spare	4750	?
DCS	350	1	4+1 spare	1750	?
FEC	?	0	80+5 spares	?	?
optical cables	?	1	4+1 spare	?	?
computer	2000	0 (have an old)	1	2000	?