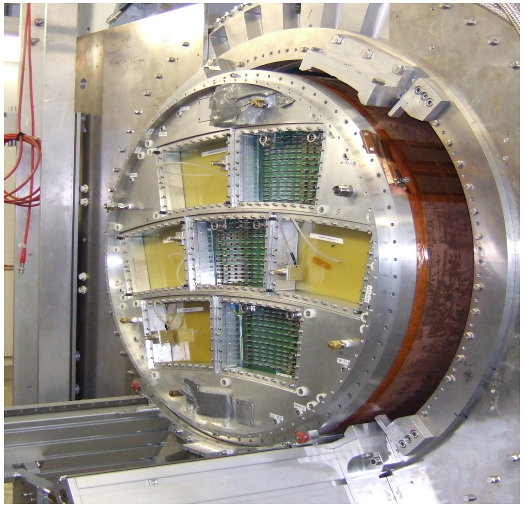
Silicon Reference Tracker into EUDAQ and TPC DAQ

Readout & DAQ

Dimitra Tsionou

EUDAQ Workshop, 16-Nov-2016

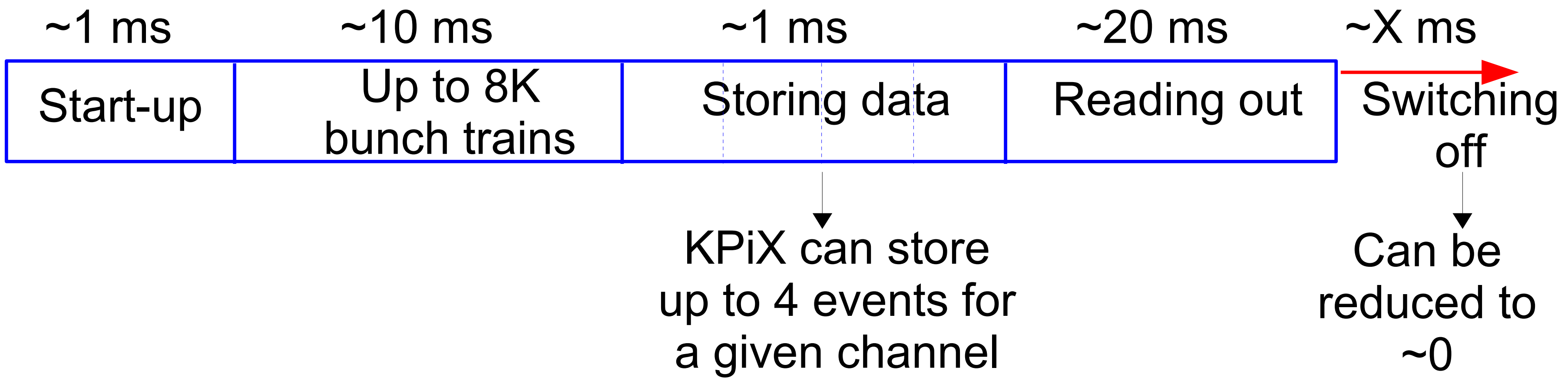
Introduction



- Test beam infrastructure at DESY including a 1T magnet (T24/1)
- We want to build and install in this area a large area Silicon telescope
- Used widely by the TPC collaboration
  - Testing different readout technologies (GEMs, InGrid, Micromegas)
  - Each technology uses its own DAQ system
- Infrastructure also used by other groups: Belle II, ATLAS
- Goal: Combined test beams with the DUT (TPC)
- Requirements:
  - Interface the Silicon telescope to EUDAQ
  - Synchronise with TPC

KPiX cycle

- KPiX chip will be used to read out the Silicon sensors (SiD tracker +calorimeters). It has 1024 channels
- KPiX developed for an ILC environment (power pulsing)



~1 ms
Start-up

~10 ms
Up to 8K bunch trains

~1 ms
Storing data

KPiX can store up to 4 events for a given channel

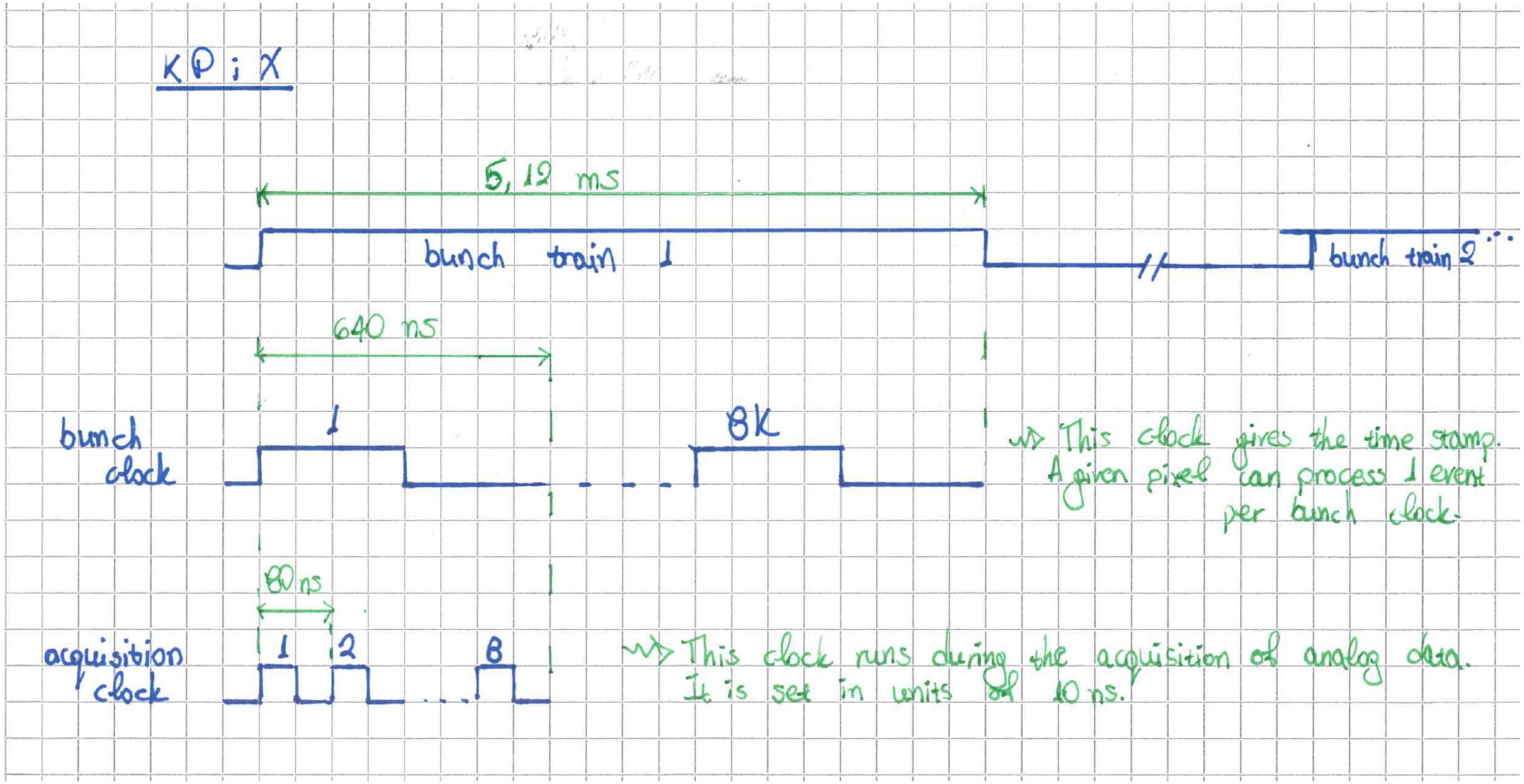
~20 ms
Reading out

~X ms
Switching off

Can be reduced to ~0

- KPiX can keep up with a ~30Hz cycle during data taking
- KPiX can work on self-trigger or forced trigger mode
- How efficiently can we use it in the test beam (with the TPC) with such a cycle?

KPiX clocks



⇝ This clock gives the time stamp. A given pixel can process 1 event per bunch clock.

⇝ This clock runs during the acquisition of analog data. It is set in units of 10 ns.

KPiX DAQ software

- Group developed DAQ (SLAC)
- Willing to integrate to EUDAQ on a top level basis (control start/stop, slcio output,..)

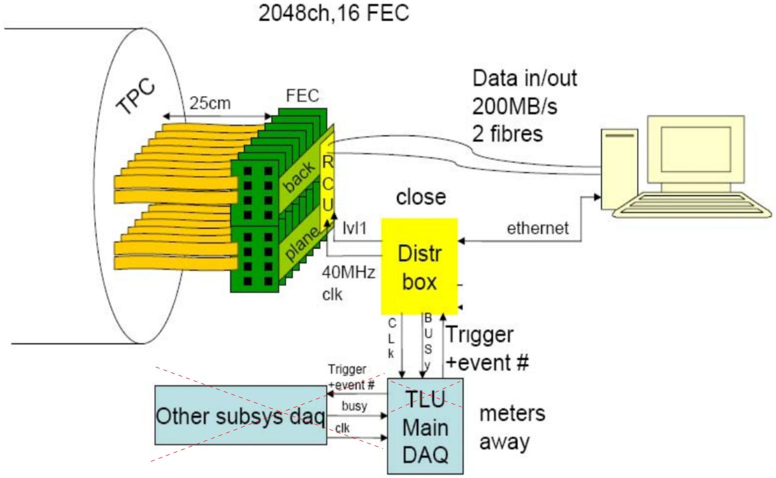
- Ongoing: Attempting communication between the two systems
- Basic producer developed for EUDAQ (based on HCAL)
- KPiX DAQ accepting information?

- Any help on this is more than welcome!

TPC readout system

- The TPC uses a modified ALTRO system from ALICE TPC – University of Lund

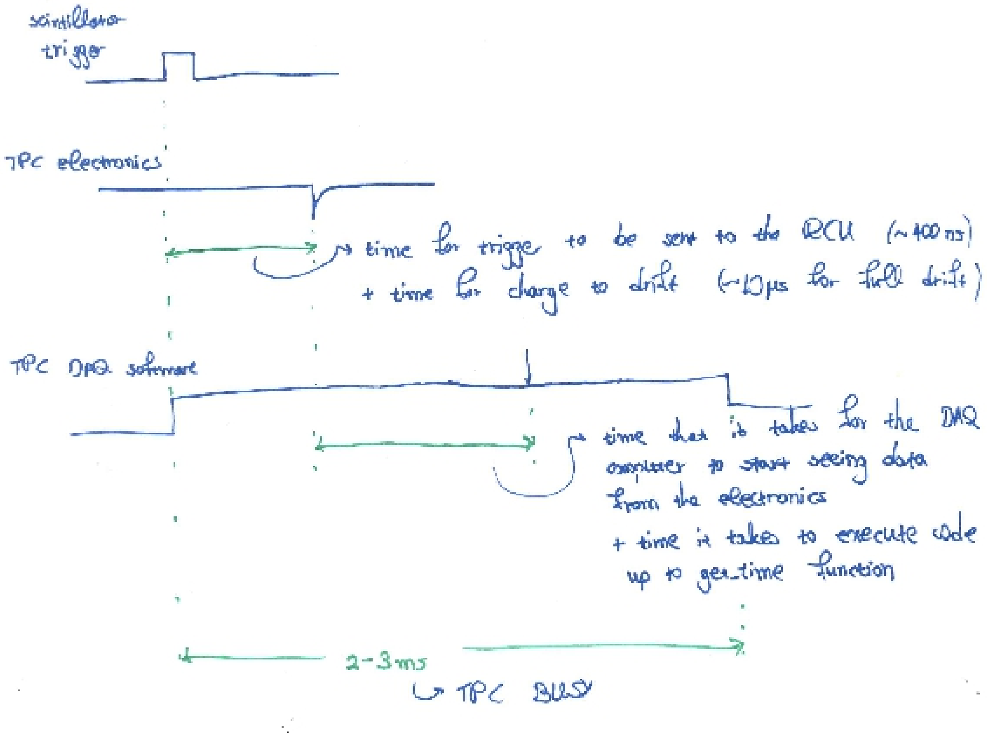
- General point: The electrons from the primary ionisation of the beam particle need to drift through the TPC volume before reaching the equipped endplate



- The signal is amplified, shaped and digitised in the 16-channel ALTRO chip with a 10bit resolution and is then stored in the event buffer (1k 10bit words)
- They are then transferred via an optical link to a PC and saved to disk
- The sampling frequency can be set to 5, 10, 20 MHz
- Run Control (control & Monitor PC): Graphical user interface (JAVA) → It communicates with the ILCDAQ server (DAQ PC) via TCP/IP using a special protocol consisting of text messages

TPC timing

- TPC is externally triggered (scintillator setup) → Trigger is sent to the front-end electronics (RCUs) that start the readout → RCU sends the data to the DAQ PC
  - Note: No hardware trigger directly to the PC
- Once the trigger is sent, the TPC goes to busy for 2-3 ms

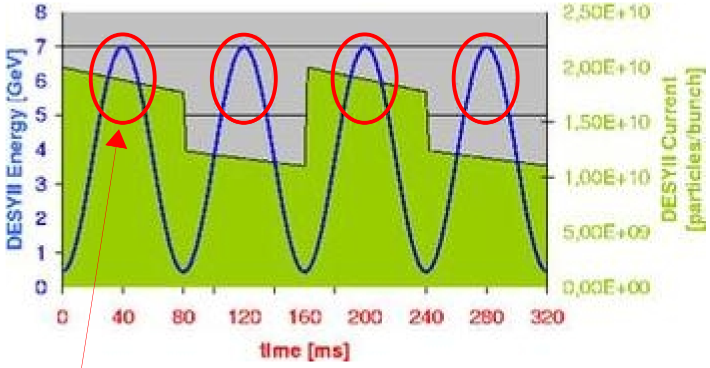


scintillator trigger
TPC electronics
time for trigger to be sent to the RCU (~400 ns)
+ time for charge to drift (~10µs for full drift)
TPC DAQ software
time that it takes for the DAQ computer to start seeing data from the electronics
+ time it takes to execute code up to get_time function
2-3 ms
TPC BUSY

- Space for timing information in the event header
  - DBOXEVTN: DBOX hardware trigger number (read from distributor box)
  - DBOXTIME: DBOX time stamp (read from distributor box)
  - TLUEVTNM: TLU event number (read from distributor box)

DESY test beam characteristics

- Bremsstrahlung photons generated from the DESYII electrons. These photons are then converted to electron-positron pairs. The beam is then spread out with a dipole magnet and certain energies can be selected at the test beam

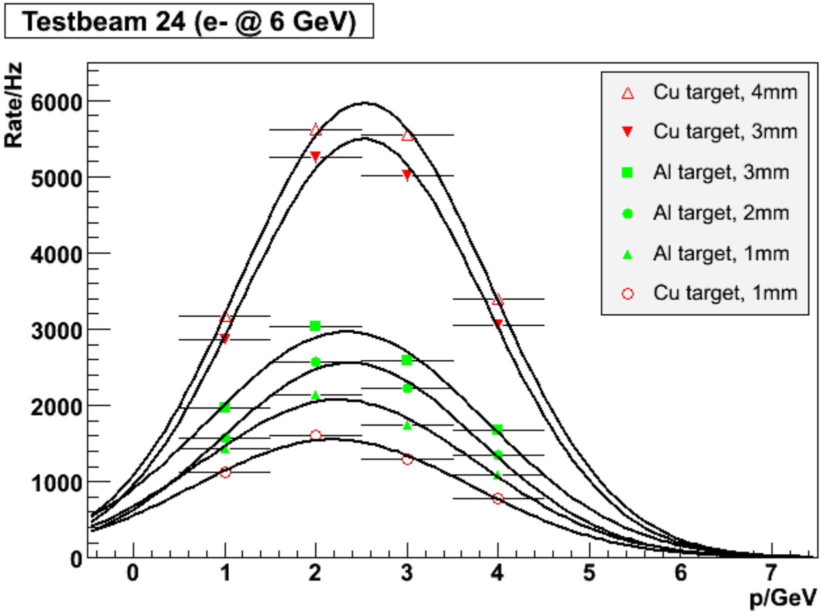


- DESYII cycle: electron energy follows a sine wave function with a minimum of 450 MeV, maximum 6.3 GeV and a period of 80ms

- 5GeV electrons at test beam area possible when DESYII energy is above a threshold. Probability maximises towards DESYII maximum energy ("spill")

- Synchronisation with kpix power pulsing for efficient data taking?
  - Time when DESYII is at minimum energy can be provided in the test beam area.
  - We can calibrate the system to know when to send a "power on" signal to kpix and adjust the kpix data taking duration

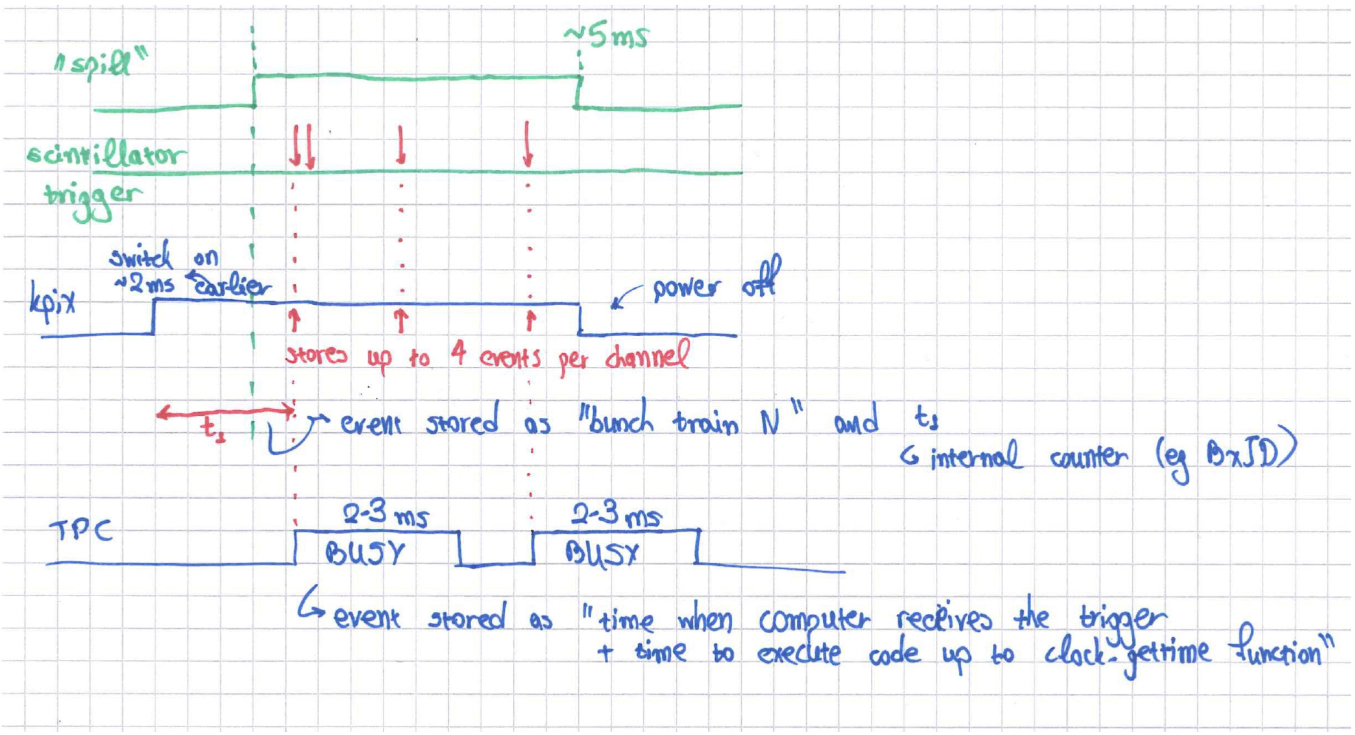
DESY rates and synchronisation



- Example: Let's assume 5GeV electron beam with 3KHz rate and 5ms “spill”
- Beam only present during the “spill” period (~5ms for 5GeV e)
- DESY has a cycle of 80ms ie 12.5 cycles per 1s

- Therefore we would have 240 e per cycle or more accurately 240 e in a 5 ms spill (on average 1 particle every 21 μs during spill)
- KPiX/TPC time precision better than that?
- Synchronisation of the two systems so that they record the same event

Integration of KPiX and TPC in the DESY test beam


"spill"
~5ms
scintillator trigger
switch on ~2ms earlier
kpix
power off
stores up to 4 events per channel
t_s
event stored as "bunch train N" and t_s
↳ internal counter (eg BxID)
TPC
2-3 ms
BUSY
2-3 ms
BUSY
↳ event stored as "time when computer receives the trigger + time to execute code up to clock-gettime function"

→ ~1 event per cycle (~10 Hz)

- Two separate data streams and data files that will be merged offline using timing information to correlate different events
- How? Synchronisation. Both systems accepting some external time reference.
- Can TLU play that role? (TPC can accept TCP/IP signals)

Summary

- Integration of KPiX DAQ to EUDAQ should be possible and we want to pursue this

- Synchronisation of KPiX and TPC is still an open issue
  - Meeting between Lund, SLAC and DESY next week to discuss the timing information needed and how it can be distributed/accepted/stored by each system

- Manpower limited. Help is welcomed!

Back-Up

TPC readout system (2)

- Run Control: Graphical user interface (JAVA) → It communicates with the ILCDAQ server via TCP/IP using a special protocol consisting of text messages

- Monitor: (C++, root) → Requests histograms from the readout and processes the event. It stores histograms that can be viewed in the presenter

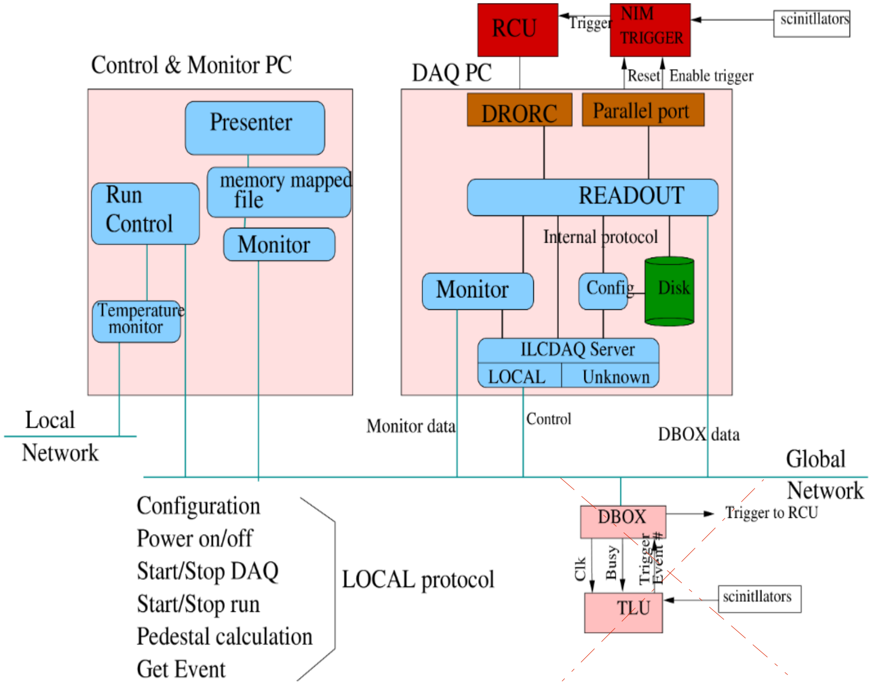


Figure 1: ALTRO TPC DAQ

TPC readout system (3)

- ILCDAQ server (C) – Interface to “outside”
  - Receives commands from run control, translates commands to instructions for hardware,..
- Starting ILCDAQ server → starts up the readout program → starts threads for handling different parts (hardware readout, configuration,..)
- Data read from the front end is stored in a local binary file

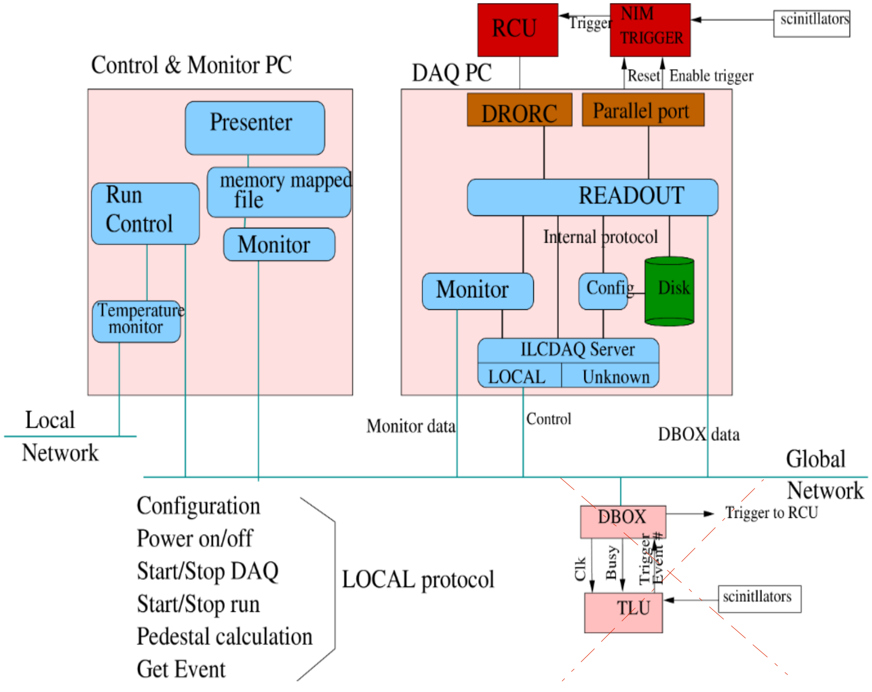


Figure 1: ALTRO TPC DAQ

- The readout part which communicates with the hardware was intended to be a standalone program and not compiled with a common data interface provided by a remote common DAQ
- Instead, foreseen to be able to send the data to any destination via the network with a format/protocol defined by the remote receiving system

Trigger (TPC system)

- A NIM crate is used as a standalone trigger system. The system gets the trigger from the beam scintillators. It works on a common stop

- When it receives a trigger, the system goes to busy (~40μs) and sends the trigger to the RCUs

  → The readout is waiting for data from the RCUs → writes the data to disk if logging is set → sends a signal to reset the busy → System ready to accept a new trigger

- The signals are sent via the parallel port of the readout PC

Past attempts – EUDAQ

- There have been some past attempts towards combined test beams between the TPC and a Silicon tracker (~2008)
- An effort was made to integrate EUDAQ and EUDET TPC DAQ
- The control part could be implemented in the ILCDAQ server
- BUT the problem was the readout → It was not possible to compile the ALICE DATE based parts (C) together with the skeleton provided by EUDAQ (C++)

  This was needed in order to send the data over the network to an event builder

  → Effort discontinued

- Possible solutions
  - Integrate the control part in the ILCDAQ server and write data to local file instead of to a common event builder
  - EUDAQ can provide the protocol to send the data?

Past attempts – TLU

- A trigger Distributor BOX (DBOX) was built as an interface between the TLU and the TPC DAQ (Brussels group)
- The box is supposed to handle the trigger from the TLU, busy from the TPC DAQ, send the hardware trigger to the RCUs, a software trigger together with trigger information to the readout program via the network using the LVDS interface of the TLU

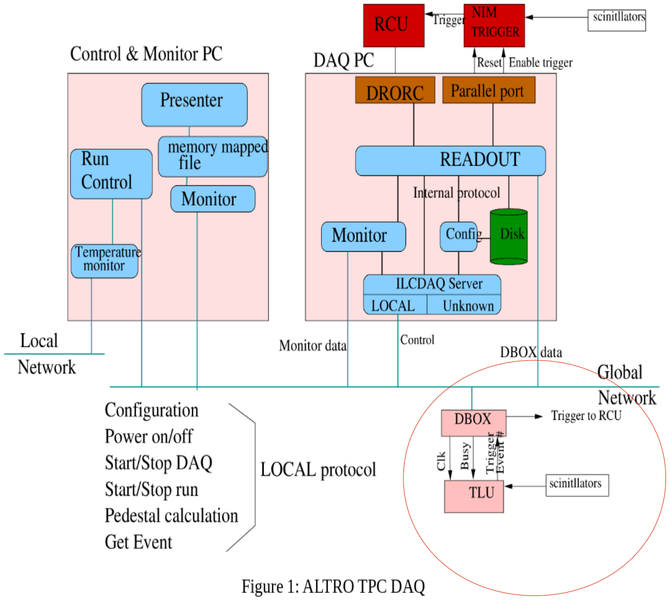


Figure 1: ALTRO TPC DAQ

- Problem: packets were lost on the network. It worked at tests in Brussels but it never worked reliably in the test beam. The reason was not understood → Effort discontinued

- Possible solutions
  - Leave out DBOX. Interface TLU with the NIM crate trigger system for TPC DAQ via the Lemo interface. The Lemo interface provides only signals for trigger and busy therefore the trigger information will not be available for event synchronisation in the TPC data

Possible future configurations – Idea 1

- Two independent DAQ systems with a common NIM based trigger system

- Relatively straightforward
- TPC DAQ is event by event based
- Data files must be merged offline
- Problem if one of the systems loses the trigger synchronisation
- Possibly sending the synchronisation information via the network between the DAQ systems?

Possible future configurations – Idea 2

- Use of a common DAQ/Trigger such as EUDAQ and TLU

- Modify the ILCDAQ server to handle the EUDAQ control and write separate files to be merged offline
  - This will avoid compiling the TPC readout with EUDAQ skeleton
- Interface the TLU with the TPC NIM trigger crate
- The TLU trigger information will not be in the TPC data stream

- Maybe it's possible to operate the TLU without EUDAQ if one doesn't want to modify the ILCDAQ server?

Summary

- TPC DAQ not foreseen and not flexible for combined test beams / common system (EUDAQ/TLU)
- Implementing full EUDAQ/TLU chain will require quite a bit of effort
- ALTRO TPC DAQ is based on an old version of ALICE hardware and software. Modernising this will be quite an effort
- Probably Si system will be more straightforward to interface to EUDAQ
- Effort will be required on the TPC DAQ side. Ongoing effort on sALTRO system
  - Main person from Lund retiring next year
- Manpower needed

Desired use of EUDAQ

- Issues to be addressed
- Combine Kpix with external scintillator trigger (if strip Si sensors are used)
- Combine the TPC DAQ (Altro) with the Si system one

- There were some past attempts in the past to combine the TPC system with a Si tracker (APV25 readout chip) using a TLU but they were not successful. Effort has been discontinued
  - Using TLU for synchronisation and with a special interface to the TPC DAQ software, send the trigger via ethernet