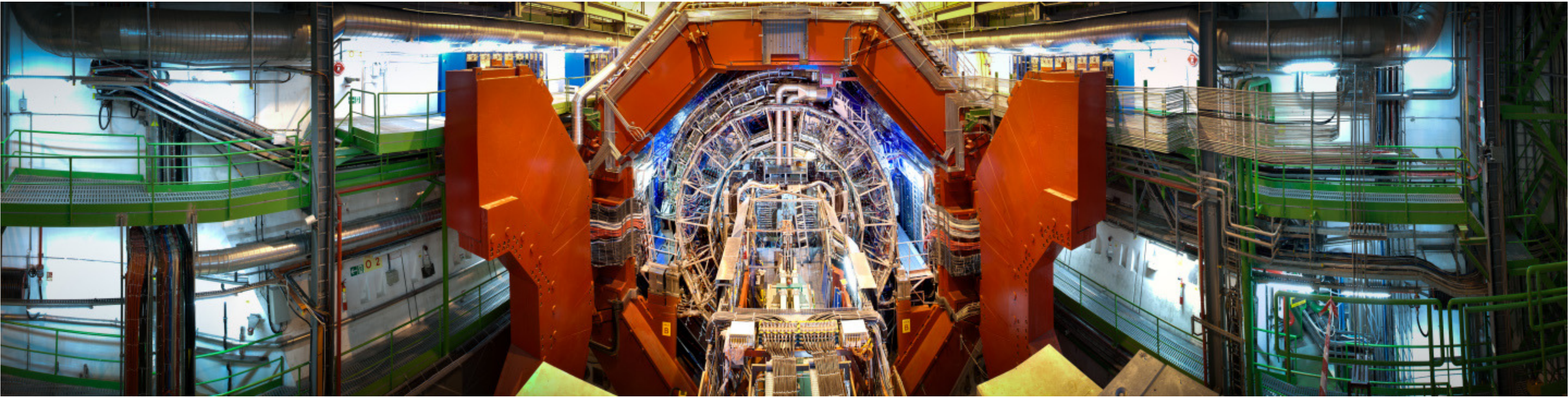
HOW LOW COST DEVICES CAN HELP ON THE WAY TO THE ALICE UPGRADE

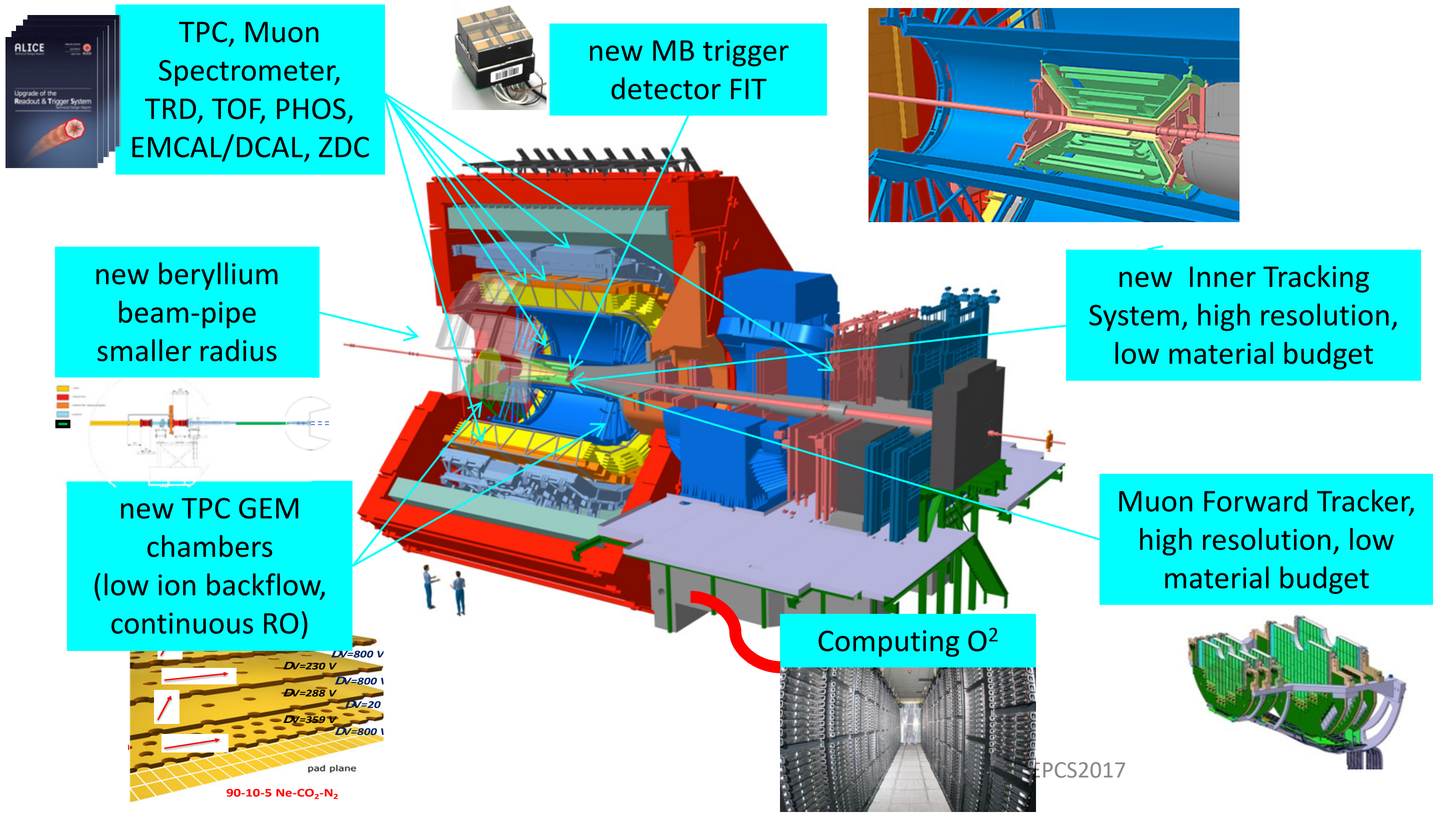


Integrating Raspberry Pi, Intel Edison and Arduino in control systems and upgrade studies

Ombretta Pinazza, CERN and INFN Bologna

on behalf of ALICE DCS group

ALICE detectors upgrade



New detectors, continuous readout, higher interaction rate: during RUN3, ALICE plans to read data at 50 kHz Pb-Pb and 200 kHz pp and p-Pb.

3.4 TB/s of data will have to be processed, selected and stored, and eventually compressed up to 100 GB/s for offline analysis.

A new powerful data processing system integrating online and offline computing (O^2) has been designed.

ALICE O^2 project

Requirements

1. LHC min bias Pb-Pb at 50 kHz ~100 times more data than during Run1
2. Rare physics processes with very small signal over background ratio
3. Triggering techniques very inefficient if not impossible
4. 50 kHz bigger that TPC inherent rate
   Support for continuous read-out

New computing system

- Read-out the data of all interactions
- compress these data intelligently by online reconstruction
- One common online-offline computing system: O^2

Unmodified raw data of all interactions shipped from detectors to online farm in triggerless continuous mode.

HI run 3.4 TByte/s

Baseline correction and zero suppression.
Data volume reduction by zero cluster finder.
No event discarded.
Average compression factor 6.6.

500 GByte/s

Data volume reduction by online tracking.
Only reconstructed data to data storage.
Average compression factor 5.

100 GByte/s

Data storage. One year of compressed data.

120 GB/s

Tier0, Tiers1 and Analysis Facilities

200 GB/s 20 GB/s

Asynchronous (hours) event reconstruction with final calibration

Detector Electronics
9000 GBTs links

270 First Level Processors
HW acc: FPGA

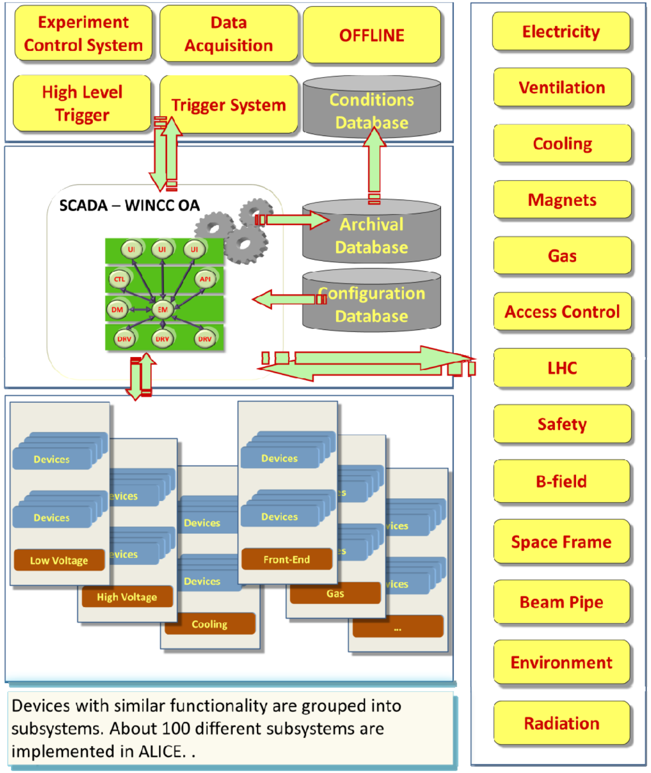
Switching networks

1500 Event Processing Nodes
HW acc: GPU

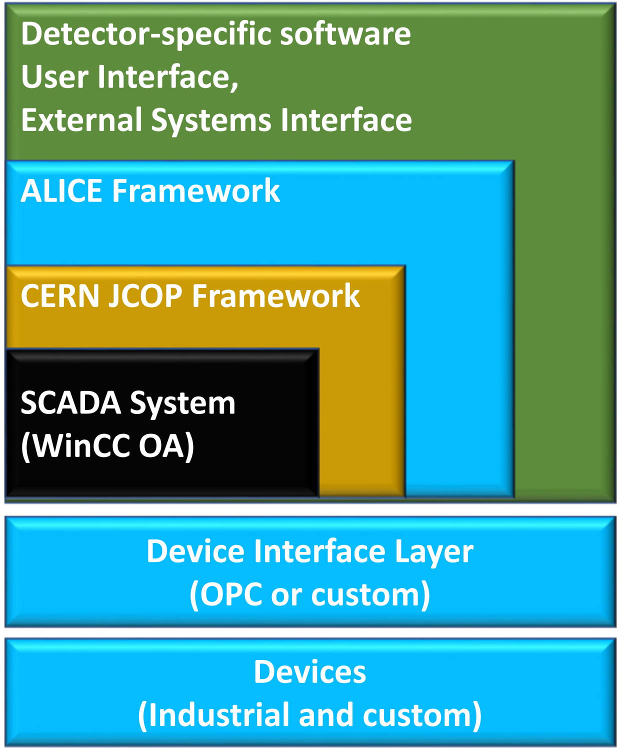
Switching networks

Write 120 GB/s
Read 320 GB/s
Capacity 60 PB

The present ALICE DCS – Context and Architecture



Devices with similar functionality are grouped into subsystems. About 100 different subsystems are implemented in ALICE. .



ALICE DCS size

- 19 autonomous detector systems
- 120 WinCC OA systems
- >100 subsystems
- 180 control computers
- >700 embedded computers
- 1200 network attached devices
- 300 000 OPC and frontend items
- 1 000 000 supervised parameters

DCS in the O² architecture

condition data flow

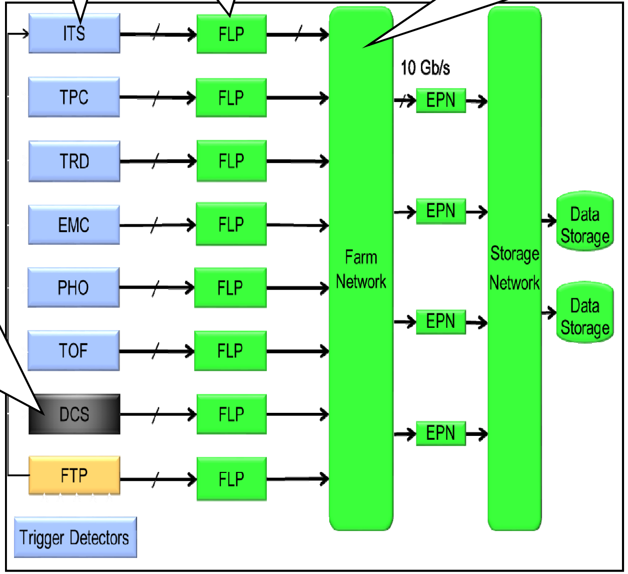
Detectors

270 First Level processors (FLP)

- 1500 Event Processing Nodes (EPN)
- ~100 000 processor cores

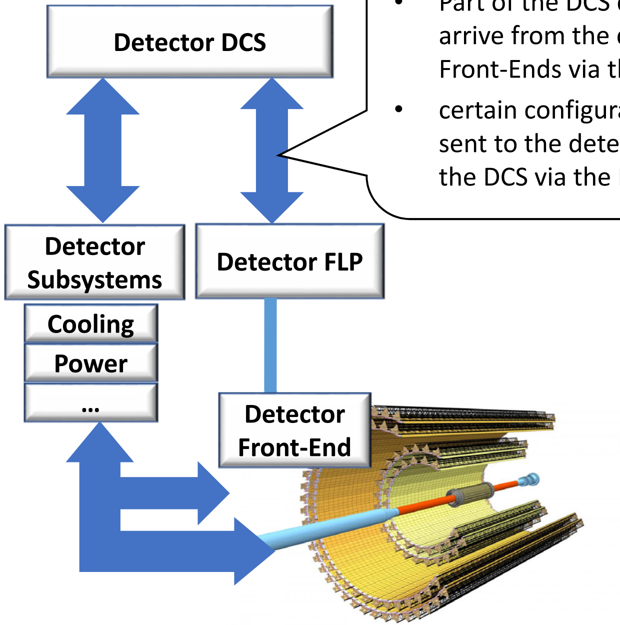
DCS provides input to O²

- ~100 000 conditions parameters are requested for the data processing in O²
- Data has to be injected into each 20 ms data frame



control data flow

- Part of the DCS data will arrive from the detector Front-Ends via the FLP
- certain configuration will be sent to the detectors from the DCS via the FLP

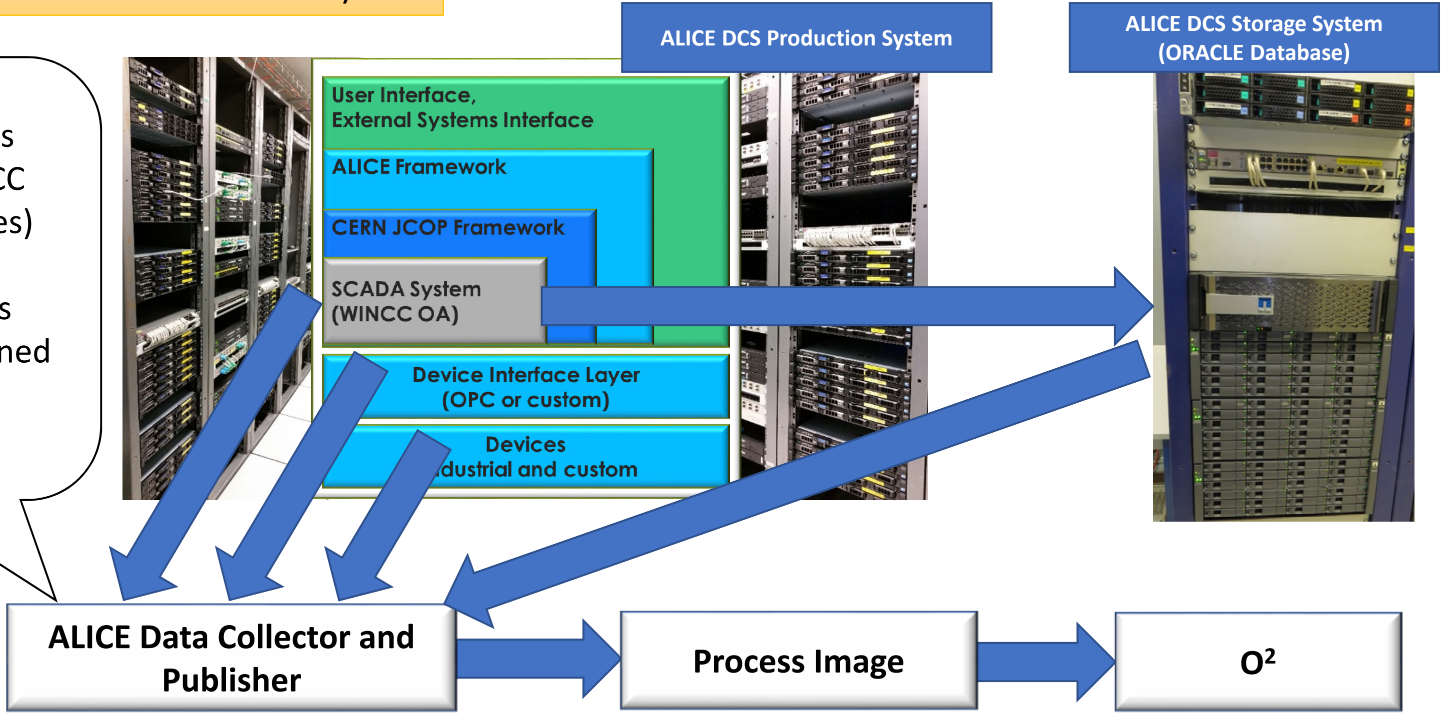


17:09 **TUMPL09** - Peter Chochula (CERN): Challenges of the ALICE Detector Control System for the LHC RUN3

The DCS-O^2 interface for conditions data flow

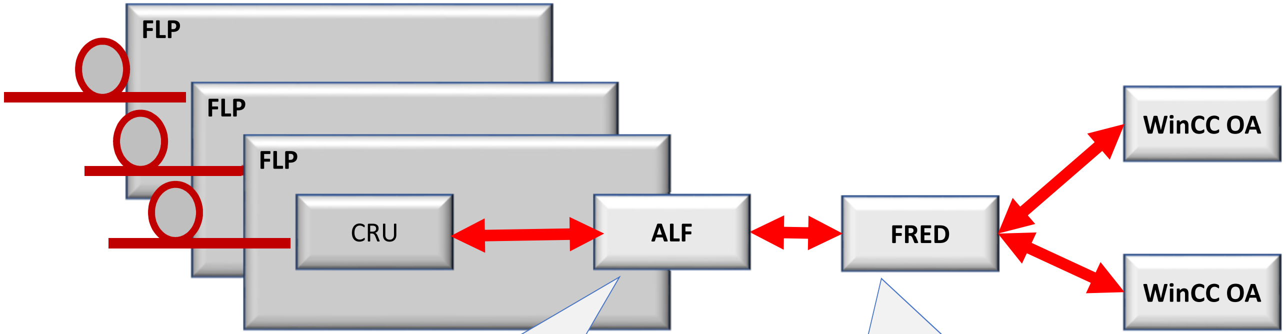
ADAPOS, the Alice DAtaPOint Server)

- the Collector receives data from DCS (WinCC OA systems or devices)
- A process image, containing conditions data is built, maintained and sent to O^2



ALICE upgrade: the DCS place in the front-end chain

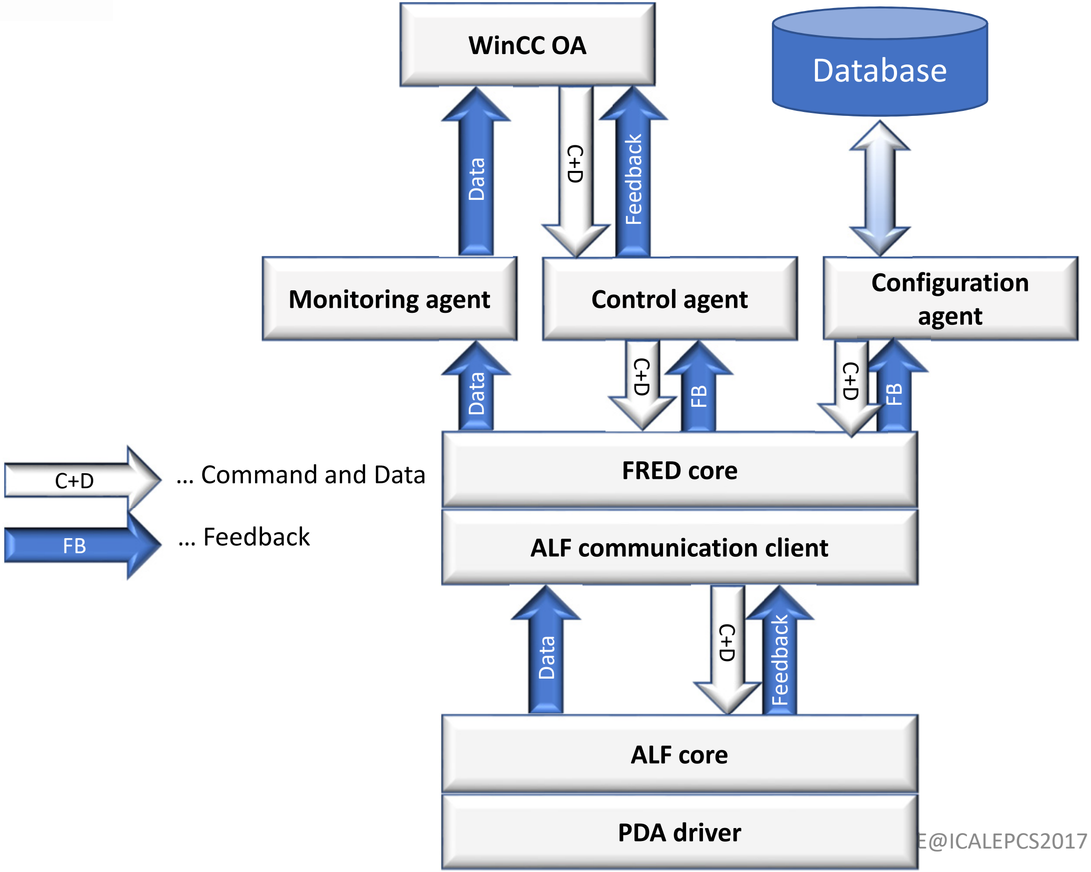
ALF-FRED, Alice Low level Frontend and Front End Device



- DETECTOR NEUTRAL LAYER ALF (Alice Low Level Frontend interface) provides communication interface to CRU (Common Read-out Unit) firmware

- DETECTOR SPECIFIC LAYER FRED (Front End Device) runs on a dedicated server
- Receives commands from WinCC OA and forwards them to ALF
- Receives data from ALF and publish it to WinCC OA

DCS-CRU software components



WinCC

WinCC OA tasks
- Full control functionality
  - Alert handling
  - Configuration
  - Control and Monitoring
  - Archival
  - User interface

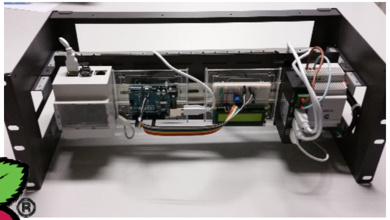
FRED

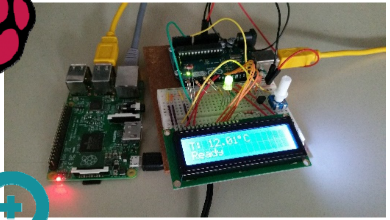
FRED tasks:
- Execution of macro commands
  - Translates complex DCS subscriptions to ALF commands
  - Example: monitor T123 on Sector 7
- Decode and analyze data
- Publish data to WinCC OA (possibility to add smoothing)

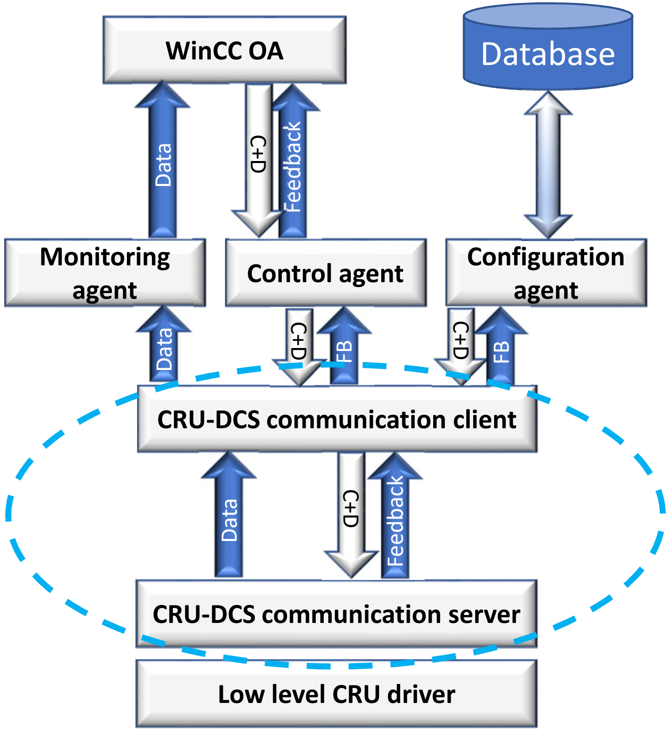
ALF

ALF tasks:
- Basic I/O
- Translation of commands (read I2C) into atomic I/O operations
  - Possibility to execute periodically
- No detailed knowledge on detector structure

ALICE ALF-FRED test benches







- Recent activities suffering from the absence of real hardware
- Full prototype realized using Raspberry Pi based emulators of the detector
- Proof of concept
- until we got a first version of the read-out setup

Lab setup: Arduino+Raspberry Pi detector simulator

WinCC OA

FRED

Raspberry Pi reading from Arduino publishing on eth via DIM

breadboard with sensors and small LCD display

Arduino read-out

PS and USB hub

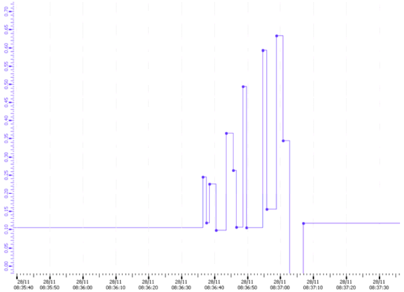
DC DC converters voltage regulator power monitoring

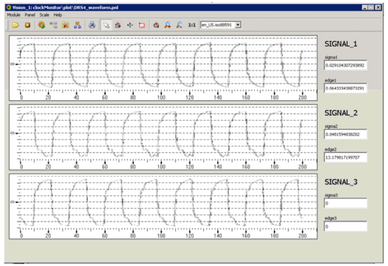
LCD touchscreen and Raspberry Pi user interface for DIM data

Other mini-computer installations in ALICE

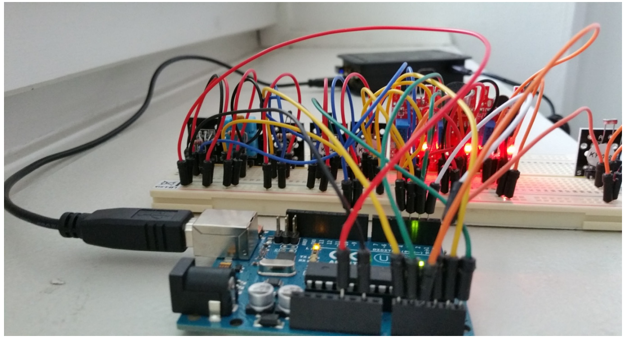
LHC clock jitter monitor

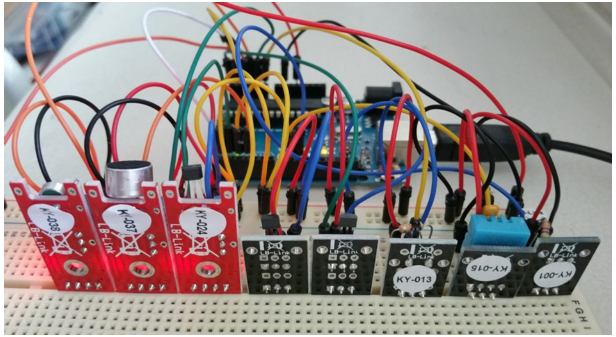
- In September 2014 we installed a Raspberry Pi in CR4, connected to a DRS4 evaluation board (PSI) able to monitor the phase difference between 4 clock signals from the LHC.
- C++ program running on the Raspberry Pi evaluates the signals phase and publishes values in a DIM service.
- able to detect clock misalignment of ~100 ps.

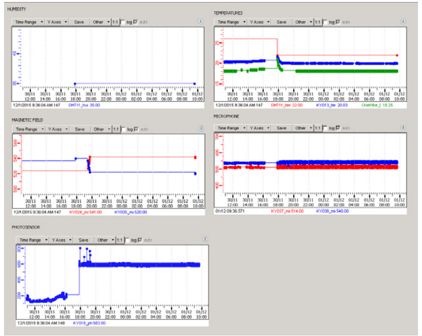


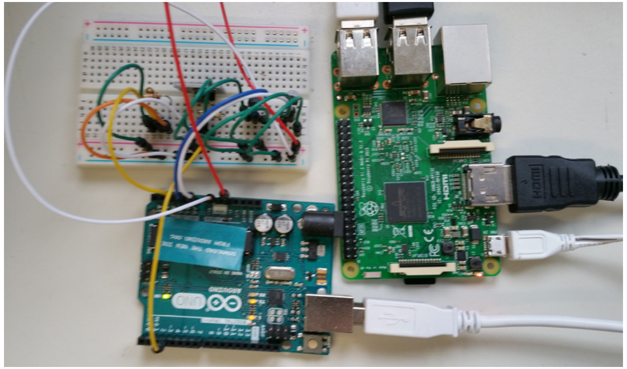


Various Raspberry Pi + Arduino + sensors setups





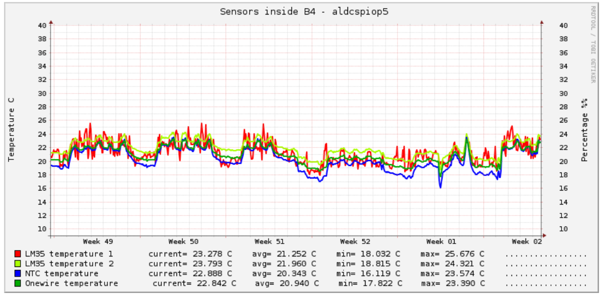




Used in our lab. for several different purposes: sensor evaluation, ad-hoc environment monitoring, demo, long-term tests.

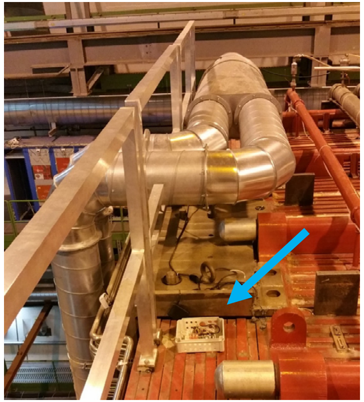
Data published via DIM
and collected on WinCC OA.

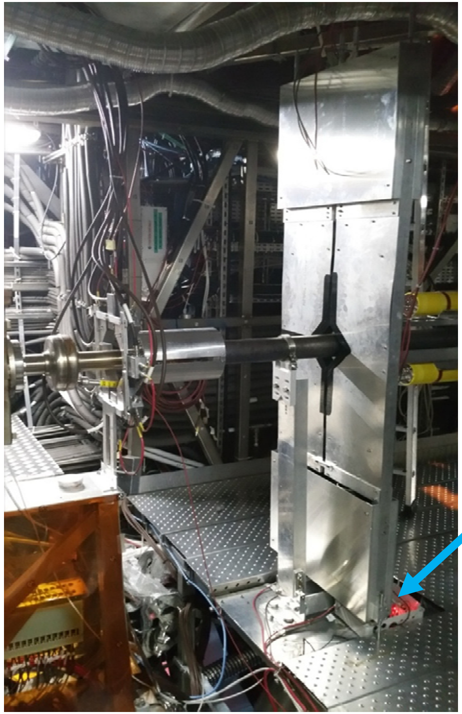
Scripts and RRD local archive
to monitor the Wi-Fi quality.

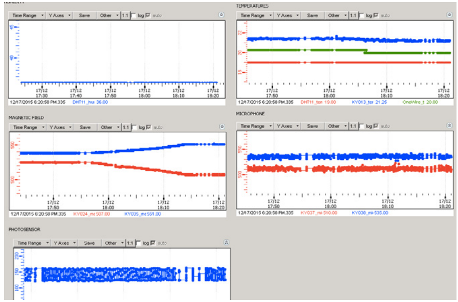


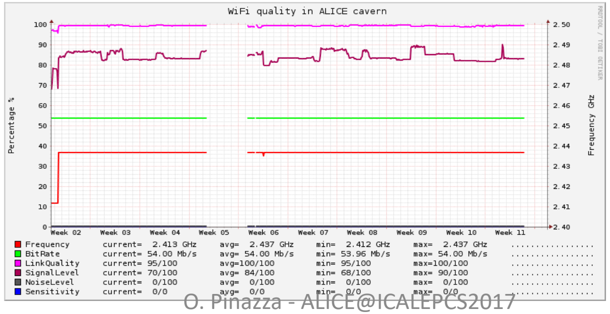
Installation inside the ALICE experiment site

measurements in various positions between October 2015 and December 2016





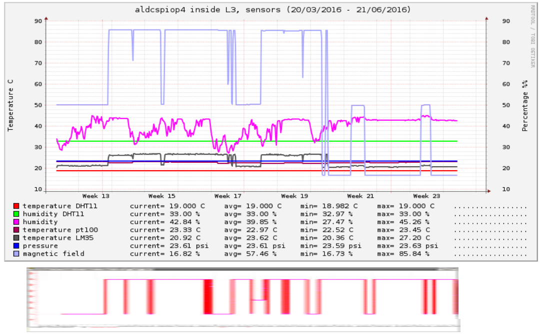




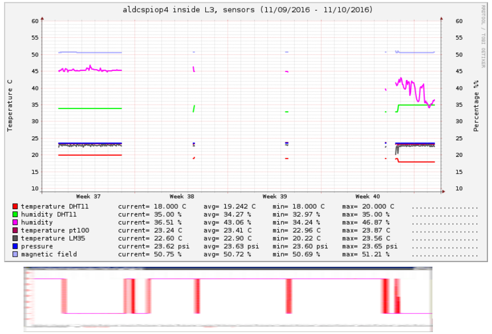
Example of data from sensors and Wi-Fi quality analysis, inside ALICE solenoid

Problems observed:

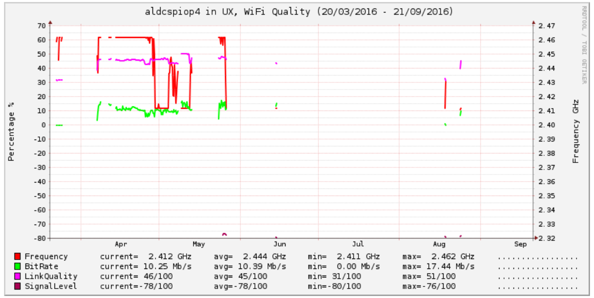
- power supplies broke
- Wi-Fi connections became erratic
- SD/uSD cards damaged

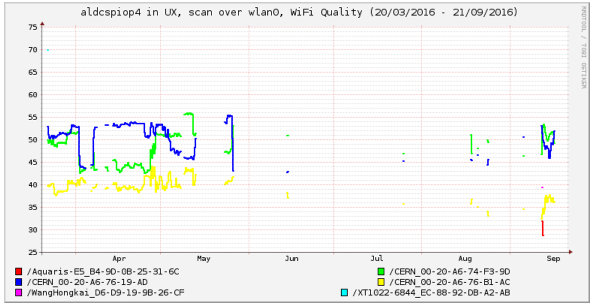


The setup survived for several weeks...



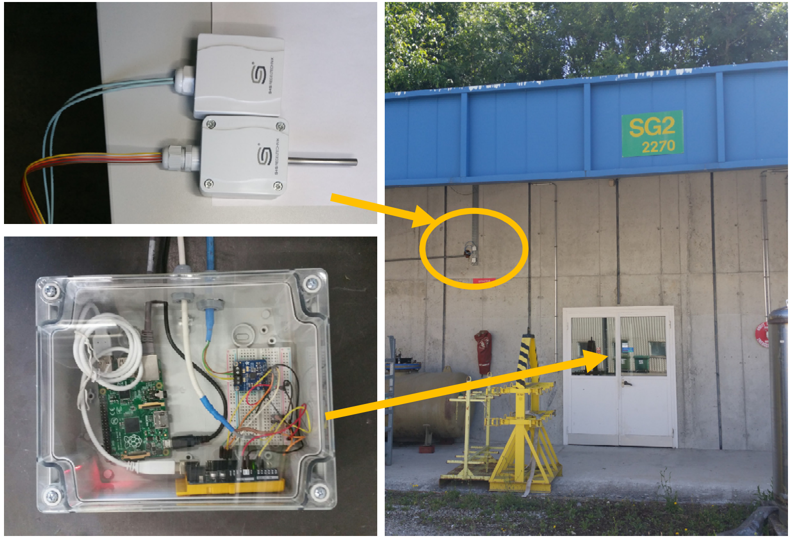
...later, the Raspberry Pi took data only when the magnet was switched off

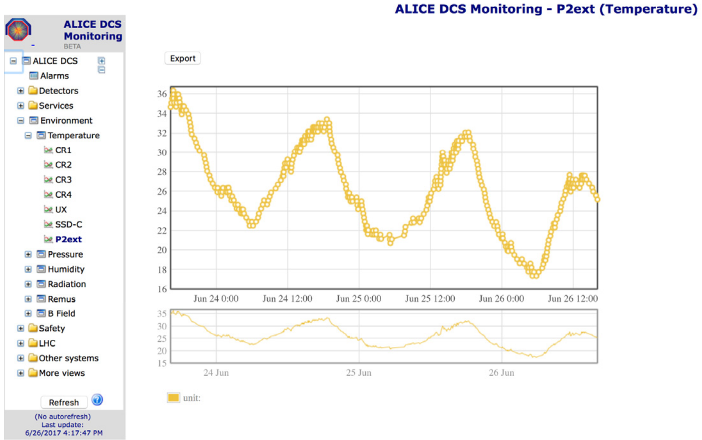


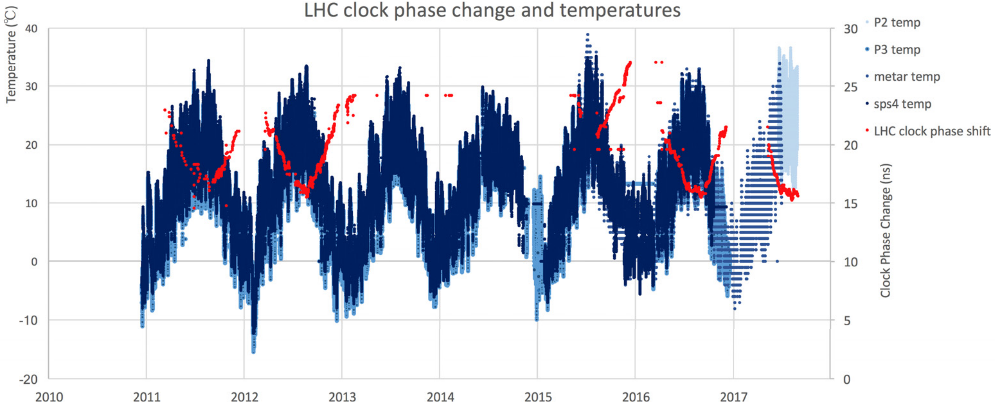


Weather station

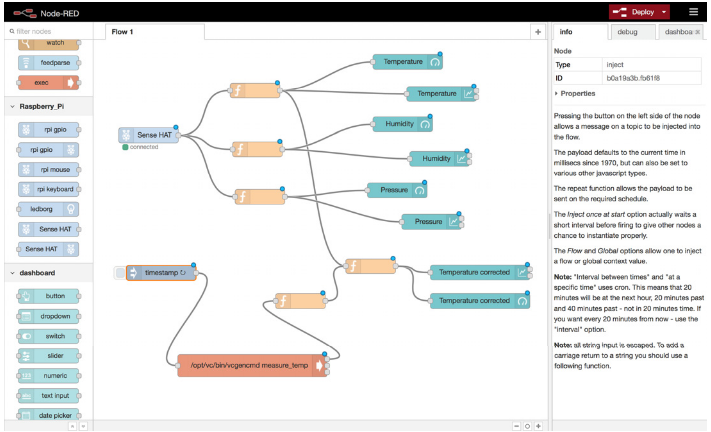
External temperature monitor integrated in the ALICE DCS, allowing online correlation with LHC clock phase shift.

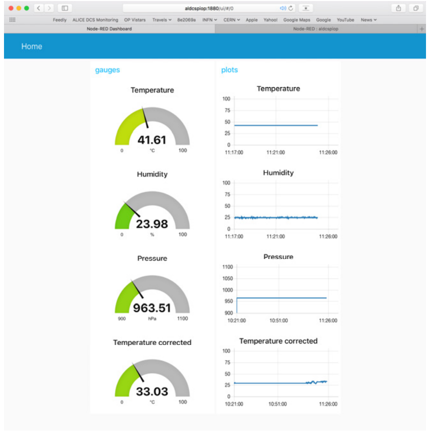






High school students’ project





Conclusions

- Minicomputers and microcontroller boards are performing sufficiently well to be used in production.
- Software developed on the minicomputers (C, C++ and python DIM services, python, perl and RRDtool scripts, GIT, ZeroMQ, Node-Red, etc.) has been partially integrated in the real ALICE DCS environment.
- Raspberry Pi and Arduino setups are successfully used in ALICE for the simulation of front-end devices, where electronics prototypes are not yet available, thus permitting the software and framework development in parallel to hardware production and validation.
- Setups like those presented here are very useful to introduce students and trainees to the realization of simple, but realistic, control systems.

Thanks.