ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA208

COMMUNICATION ARCHITECTURE OF THE DETECTOR CONTROL SYSTEM FOR THE INNER TRACKING SYSTEM

J. Jadlovský, A. Jadlovská, S. Jadlovská, M. Oravec, D. Voščєk, M. Kopčík, J. Čabala, M. Tkáčik,
Technical University of Košice, Košice, Slovakia
P. Chochula, O. Pinazza, CERN, Geneva, Switzerland

Abstract

This paper presents the proposed communication architecture of the Detector Control System (DCS) for the Inner Tracking System (ITS). The purpose of the DCS is to acquire and control the states of the ITS. Since the ITS is not yet fully implemented, an emulator of the communication architecture is being developed. The proposed architecture comprises five levels. At the bottom, the detector is emulated by sensors connected to microcontrollers. Each microcontroller is then connected to a Raspberry Pi which represents the ALICE low-level front- end (ALF) electronics at the second level of communication architecture. The third level is represented by Front-End Device (FRED), a Linux server where more than one ALF device can be connected. FRED is then connected to the fourth level, implemented by the SCADA interface – WinCC OA. Above all these levels is an archiving and configuration database setup. Configuration bypasses the SCADA interface and is managed directly through FRED. The purpose of the emulator is to verify the proposed architecture in terms of data throughput and cooperation of the mentioned modules.

INTRODUCTION

Since 2012, the Technical University of Košice, represented by the Center of Modern Control Techniques and Industrial Informatics at the Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics has been the member of the ALICE project and since 2015, the full member of the ALICE Collaboration in CERN. During this time, its members have worked on several assigned tasks. One of the tasks carried out by our team is the development of a new communication mechanism dedicated to the front-end electronics control and monitoring to be used in the ALICE experiment at CERN. A new architecture called ALFRED (Alice Low-level Front-end Device) is now being tested using various electronics prototypes. The final version of the software will be deployed for the new inner tracker system (the ITS) currently developed by the ALICE collaboration [1], [2].

The new data acquisition architecture for ALICE [3] requires a complete redesign of the front-end access mechanism. A modular, scalable and reliable architecture is required to controls the front-end modules installed in the experiment. This paper focuses on the developments serving as verification of the concept and base for the final implementation. After the proposal of the communication architecture, tests to obtain throughput of this architecture were performed.

DETECTOR CONTROL SYSTEM AND INNER TRACKING SYSTEM

The present inner tracker of the experiment has been designed to cope with the requirements of the LHC operation until 2018. During its operation its performance fulfilled the expectations. However, the architecture of the ITS is not compatible with the new requirements after the planned LHC upgrade. The ALICE experiment operation at high luminosities triggers a need for faster detector readout and improved tracking capabilities. The new ITS composed of pixel detectors replaces the present tracker, combining silicon pixel, strip and drift detector technologies.

Detector Control System

Detector Control System (DCS) continuously ensures secure and stable operation of all sub-detectors as well as the entire ALICE detector. DCS provides remote control and monitoring of all systems. DCS uses various industrial as well as custom devices such as low and high voltage power supplies, PLC computers, custom made data acquisition boards or front-end electronics. They are hierarchically interconnected within a distributed control system. The majority of these devices communicate via Ethernet, controlled either by industrial standards such as OPC or the DIM protocol [4] developed at CERN. Other devices use other industry standard buses such as CANbus, Profibus or RS232.

Supervision of individual devices is provided by SCADA/HMI software WinCC OA from Siemens using the JCOP framework. WinCC interface enables a single operator to operate the whole ALICE experiment. Within the ALICE experiment, each detector has its own independent distributed control system containing from 2 up to 16 WinCC OA systems. All of these systems are interconnected by a central DCS, comprising together one complex system. Control hierarchy of the entire detector and all its subsystems is made up of finite state machines (FSM) organized into multi-layer tree structure.

Detector Control System for Inner Tracking System

The structure of the new ITS control system follows the new architecture of the detector as seen in Fig. 1. The main changes in the traditional design are mainly in the frontend part, which will be accessed by the GBT link [5], shared with the data acquisition system. The physics data, containing the information about the particle trajectories, is sent to the further levels of processing implemented in the O2 facility [3]. The conditions data supervised by the DCS is processed in the SCADA system and a subset of the

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA208

information is merged with the physics data stream, to be used as reference for analysis.

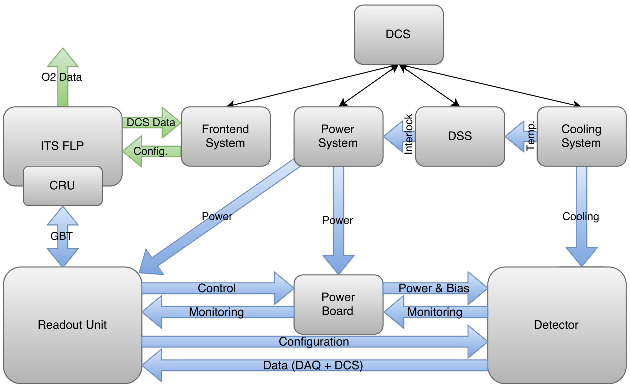

Figure 1: DCS schematics for ITS detector.

Besides the front-end control, the main DCS blocks include the power system, cooling and temperature monitoring and a safety system that protects the detector in case of dangerous conditions. Power supply of pixel detectors is provided by the CAEN power system comprised of the Mainframe that communicates with the WinCC OA over Ethernet through the OPC protocol.

Easy Crates are placed in the near vicinity of the experiment and they are designed to work in a strong magnetic field and at a high level of radiation. They are controlled by the Mainframe on the surface and provide power to Readout Unit (RU) and Power Board (PB) units.

Subsequently, the PB provides power to the pixel detector itself and monitors its temperature and current. RU allows control and monitoring of the PB as well as data configuration and read-out from the detector itself through the Gigabit Optical Line (GBT) to Common Readout Unit (CRU). The CRU sends signals through the optical band Gigabit Optical Line (GBT) to the detector electronics such as commands for setting electronic parameters, ordering data, setting memory detector registers, etc [6].

Detector response comprises physical data and DCS data (temperature, register values, masks, electronics settings, streams ...) which is inserted between physical data as frames. ITS FLP (First Level Processor) is a computer that hosts a Common Readout Unit (CRU) and selects DCS data from the stream which is then sent to the Front-end system [7].

The Front-end system connects the DCS with the detector itself and serves for configuring detector and reading DCS data.

Control and detector module conditions data monitoring shares the GBT link with the data acquisition system. This elegant approach reduces the required infrastructure, however it does not provide sufficient redundancy for the controls data. Merging the two systems with two different roles also creates difficulties for the hardware and software maintenance. In the case of the FLP unavailability, the DCS would not be able to access the chip information and a detector shutdown would be required. For this purpose a new link based on CanBus has been implemented. It will assure the basic DCS functionality during the downtime of the FLP.

Another important task of DCS is the cooling of the detector, which is provided by two systems. The air cooling system ensures ventilation between the individual layers of pixel detectors, thereby maintaining the same air temperature throughout the whole detector. The water cooling system is used to cool individual staves at the same time while monitoring the temperature, pressure and flow rate in the inlet and outlet tubes of the stave cooling system. The cooling device itself communicates with the DCS via the Modbus bus. Measured temperatures are monitored by DSS (Detector Safety System), which consists of two redundant PLCs. In the case when DSS detects critical values on some of the inputs, it generates signal Interlock, which turns off detector power. This process is fully automated and independent on the WinCC OA control system.

ALFRED - ALICE LOW-LEVEL FRONT-END DEVICE

In general, the ALFRED (Alice Low-level FRont-end Device) describes the infrastructure which determines the communication between the low level front-end devices and SCADA system of the new DCS system for the ITS in the ALICE experiment, as seen in Fig. 2.

At present, an R&D project has been launched to explore the possible implementation of the final system. A test setup based on detector and front-end emulators has been prepared. Low-cost devices based on Raspberry Pi and Arduino as well as a general purpose device Board51 [8] provide the information which is close to the final environment. This approach overcomes the limitations due to the lack of realistic hardware that is still being designed and allows for developments of the high-level software modules. The main task is the design of a universal API, which will collect and manage the data produced by custom electronics modules.

The development setup uses devices and program modules, which simulate the DCS infrastructure and its levels. The lowest level groups temperature and/or current sensors and DC/DC power source. The second level of the DCS infrastructure id formed by control and DAQ devices such as Arduino, Board51 or ELMB, which are based on the single-chip microcomputers as well as the PLC automates. This level of the DCS corresponds to the future front-end electronics subsystem of the ITS.

The devices from the second level are interconnected with Raspberry Pi computers, where the ALF(ALICE Low-level Front-end interface) server module runs. Each Raspberry Pi computer communicates via Ethernet using DIM protocol with Linux server, where the FRED (FRont-End Device) server module runs and multiple ALF modules can be connected to each FRED module. The FRED module merges and distributes data from multiple ALF modules and manages the configuration data from the configuration database

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA208

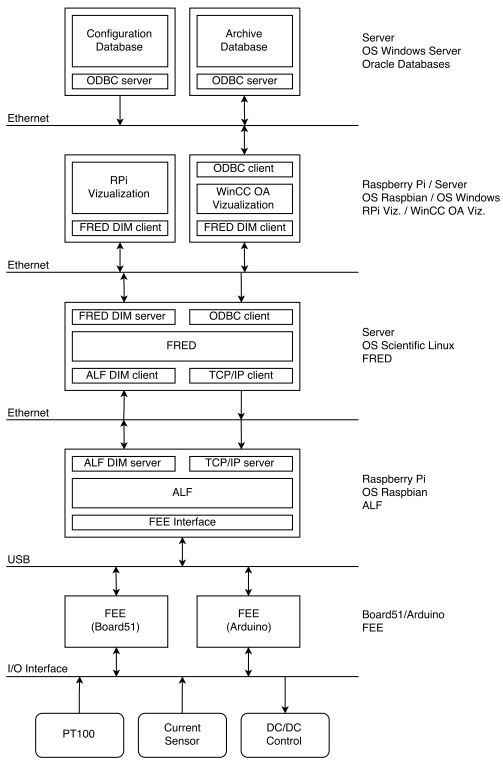

Figure 2: ALFRED communication architecture.

to the each FRED. The ALFRED module will be explained in detail later in this paper.

SCADA/HMI level is represented by visualization in WinCC OA, which ensures the acquisition and visualization of the collected data obtained using FRED module from the lowest level of the DCS of ITS detector. WinCC OA also subscribes to the control commands of the electronic devices through the ALF - FRED modules and the data archiving in database. A similar visualization is implemented on Raspberry Pi with a touch screen that provides DC-DC converter control as well as current values of the temperature and current.

Front-end Electronics

Board51 is a universal device for data acquisition from connected sensors and also could be used for the control of the actuators. Detailed description of individual parts of Board51 is given in [8]. As far as the communication architecture is concerned, the A/D converter of the device measures voltage on the temperature sensor (PT100 sensor), which is mounted on the load of the DC-DC converter. The A/D converter is also used for the measurement of a current, which flows through DC-DC converter. Measured values for both the temperature and current are then calculated from the measured voltage values of the A/D converter. The digital input of the Board51 is used for indication of the DC-DC converter state, which can be switched on or off. If the DC-DC converter is switched on, the temperature or current is checked for their overpassing of maximum allowed values, i.e. threshold. In case of the temperature or current value exceed their maximum threshold, the digital output of the Board51 is set to logical zero and the DC-DC converter is automatically switched off.

Measured data from the Board51 are published to Raspberry Pi and the control commands are subscribed from the Raspberry Pi via DIM protocol. There is a running ALF module in the Raspberry Pi, which provides data and obtains control commands from the higher levels of the DCS of ITS.

ALF Program Module

The ALF module provides communication between front-end electronics and FRED via DIM protocol and it is implemented in C++. It exchanges data and consists of the DIM service *ALF / FEE_DATA*, which publishes the measured values of temperature, current and state of the DC-DC converter obtained from the Board51. The ALF module also includes DIM command *ALF / DCDC_CMD*, which is used for the control of the DC-DC converter connected to Board51. In the separate thread of the program, the TCP server runs, which waits for the configuration data from the FRED module. After receiving of these data, the configuration string is sent via the serial link to the Board51.

FRED Program Module

The FRED program module is implemented in C++ and ensures collection of data from ALF modules and their distribution to the WinCC OA visualization tool. It also downloads the configuration data from the database which are then sent to ALF modules via the TCP protocol. The DIM services publishes data, which are received from the ALF modules. Three DIM services are used, specifically *FRED TEMP* for temperature measurement, *FRED DC_CURRENT* for current measurement and *FRED DC_STATUS* for the DC-DC converter status. FRED module also includes DIM command *FRED ID_COMMAND* that sends identification number of the configuration file in database. According to this identification number configuration parameters are downloaded to the front-end electronics (FEE) using ALF modules.

WINCC OA Visualisation

Data from the FEE devices, which flows through the ALFRED infrastructure, are visualized in SCADA / HMI tool WinCC OA, developed at Siemens. The graphical window of the visualization allows the user to switch the DC-DC converter on/off, display measured data and download selected configuration of the FEE devices. WinCC OA is connected to the FRED program module using the tool based on DIM protocol, which is part of the JCOP framework. The DIM services and commands, which are published by the FRED server, can be found and subscribed using this framework.

The application also enables archiving of measured values, which are received from the FRED, to Oracle databases using the built-in RDB archiving module.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA208

Raspberry Pi Visualization

In parallel to the main data processing architecture, a visualization of the acquired data is available on a touch screen controlled by a Raspberry Pi computer. Visualization is displayed by a GUI implemented in C++ and Qt graphical libraries (Fig. 3). Functionality of this visualization is very similar to the one implemented in WinCC OA, but the GUI is simpler. The program module, created using the DIM protocol, is connected to the FRED server.

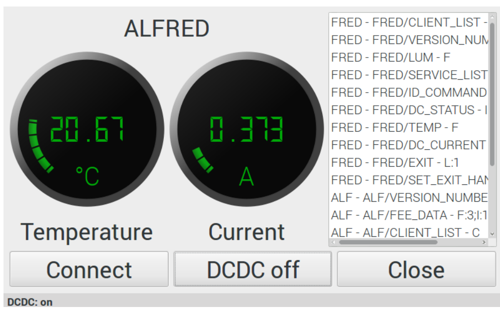

Figure 3: The GUI of Raspberry Pi Visualisation.

RESULTS

The measured time responses of the data block of various sizes can be seen in Fig. 4 for DIM. The time response increases almost linearly due to the increasing data block. The DIM protocol has a response of 260 μs at block sizes of 1B and 890 μs at a block size of 10KB.

Using the DIM protocol, as can be seen in Fig. 5 data collisions do not occur with the increasing size of the transmitted block and the bandwidth of the line stays at almost 12 MB/s at the data size of 10KB. Since the computers used in these tests had Ethernet network cards with a maximum transfer rate of 100 Mbps (12.5 MB/s), it can be noted that the line throughput was not restricted by the DIM protocol but by the hardware implementation of the line itself.

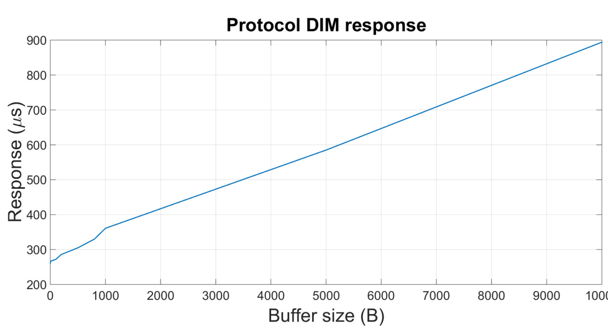

Figure 4: The DIM response according to buffer size.

CONCLUSION

The purpose of this paper was to present the proposed communication architecture of DCS for ITS and to describe its individual modules. The communication architecture is emulated within conditions which are expected to be

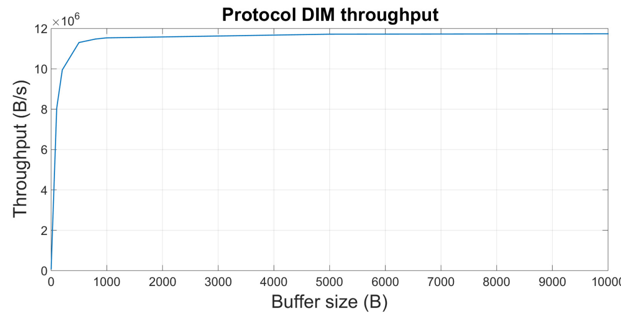

Figure 5: The DIM throughput according to buffer size.

present in the real ITS detector. Therefore complex conditions have to be met and taken into account. The architecture is based on several different implementations starting with Arduino MCU, Raspberry Pi, Linux server up to the WinCC OA SCADA/HMI system and databases. Tests validating throughput via DIM and OPC protocols were also conducted in the favour of DIM protocol.

Future research will be focused on replacing emulated parts with real simulators/hardware modules as well as testing the throughput of such a modified architecture.

REFERENCES

[1] A. Augustinus, P. Chochula, L. Jirdén, M. Lechman, P. Rosinský, O. Pinazza and A. Moreno, *"Computing architecture of the ALICE Detector Control System"*, ICALEPCS 2011, Grenoble, France, 2011, MOPKN018.

[2] S. Kushpil and ALICE collaboration, *"Upgrade of the ALICE Inner Tracking System"* Journal of Physics: Conference Series. Vol. 675. No. 1. IOP Publishing, 2016.

[3] K. Pugdeethosapol, V. C. Barroso, K. Akkarajitsakul and T. Achclakul, *"Dynamic configuration of the computing nodes of the ALICE O^2 system"*, in *Proc. of JCSSE 2016*, (pp. 1-5). IEEE. doi: 10.1109/JCSSE.2016.7748855.

[4] C. Gaspar, M. Dönszelmann and P. Charpentier, *"DIM, a portable, light weight package for information publishing, data transfer and inter-process communication"*, Computer Physics Communications, 2001, 140(1-2), 102-109.

[5] J. Schambach, M. J. Rossewij, K. M. Sielewicz, G. A. Rinella, M. Bonora, J. Ferencei and T. Vanat, *"ALICE inner tracking system readout electronics prototype testing with the CERN "Giga Bit Transceiver""*, Journal of Instrumentation, 2016, 11(12), C12074.

[6] J. Mitra, S. A. Khan, , S. Mukherjee and R. Paul, *"Common Readout Unit (CRU)-A new readout architecture for the ALICE experiment"*, Journal of Instrumentation, 2016, 11(03), C03021.

[7] S. Pumma, *"Data Synchronization Method Between FLPs and EPNs"*, (No. CERN-STUDENTS-Note-2014-016).

[8] J. Jadlovský, A. Jadlovská, S. Jadlovská, M. Oravec, D. Voššček, M. Kopčík, J. Čabala, M. Tkáčik, P. Chochula and O. Pinazza, *"A Novel General Purpose Data Acquisition Board with a DIM Interface"*, ICALEPCS 2017, Barcelona, Spain, 2017, THPHA076.