19th International Conference on Electromagnetic Isotope Separators and Related Topics
– Poster session

Design and Development of Control system for the RAON μSR facility in Korea using EPICS

[1]Jae Young Jeong, [2]Wonjun Lee, [2]Ju Hahn Lee, [1]Seung-Beom Goh, [1]Jae Chang Kim, [3]Taek Jin Jang, and [1]*YongKyun Kim
[1]*Department of Nuclear Engineering, Hanyang University, Seoul 04763, Korea*
[2]*Institute for Basic Science (IBS), Daejeon 34047, Republic of Korea*
[3]*Department of Physics, Chungang University, Seoul 06974, Korea*
[*]*a corresponding author: ykkim4@hanyang.ac.kr*

Introduction

For accelerator facilities in purpose of high-energy physics experiments, EPICS is known as a powerful tool to control devices in remote and manage them in the integrated system. The μSR facility in RAON is also controlled by a control system programmed with EPICS and Arduino modules, Siemens Programmable Logic Controller(PLC) are also involved to control and monitor the devices which does not contain serial communication protocols.

Structure of the system

Three EPICS IOCs are executed in individual PC and share there data through the Channel Access gateway. Schematics of the control system structure is described in Fig. 1.

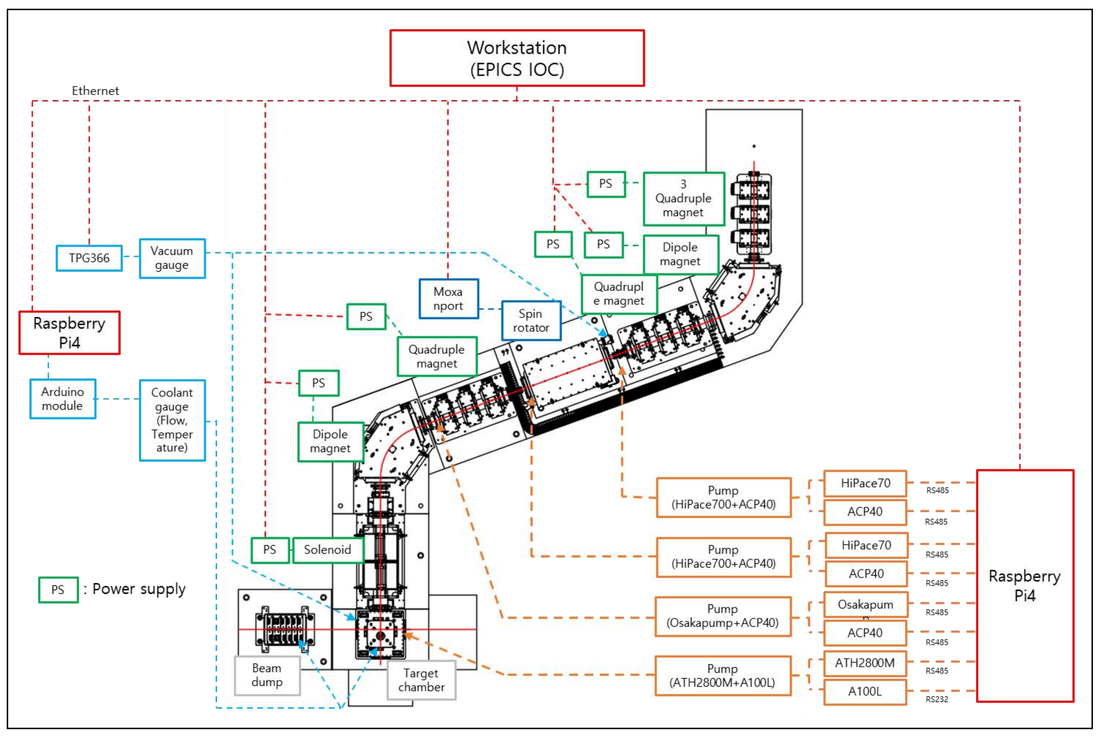

Fig. 1. Schematics of the control system

Red lines mean they are connected by Ethernet network. Orange lines mean that they are parts of the motor control system connected by RS232 and RS485 wires for serial communication. Green wires means they are parts of the magnet control system connected by thick electric wires. Blue lines mean that they are parts of the monitoring system which communicates in other ways such as USB.

The first one is executed on the main workstation. It controls and monitors all devices which are connected at the ethernet network such as MOXA devices and TPG366 pressure monitoring device. The others are executed on Raspberry Pi4 which are directly connected to the devices that have no ethernet port. IOCs on each Raspberry Pi4 only have PVs to control each connected device.

The flowmeters in the facility show the temperature and flow rate of coolant by current level. So, there is necessity of a device which translate the current value into the temperature and flow rate value. We made the Arduino module to do that work. Fig. 2 is the module which placed next to the target chamber.

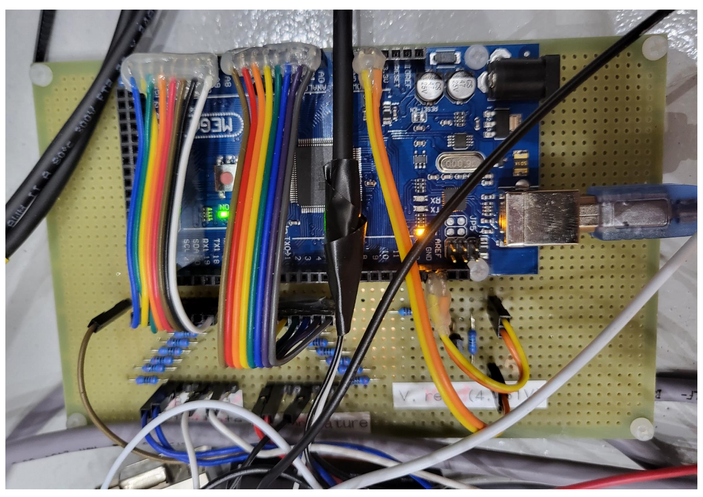

Fig. 2. Arduino module to get temperature and flow rate value from gauges

There are gate valves to prevent a rapid change of the pressure in beamline. The position of gate valves is described in Fig. 3 below.

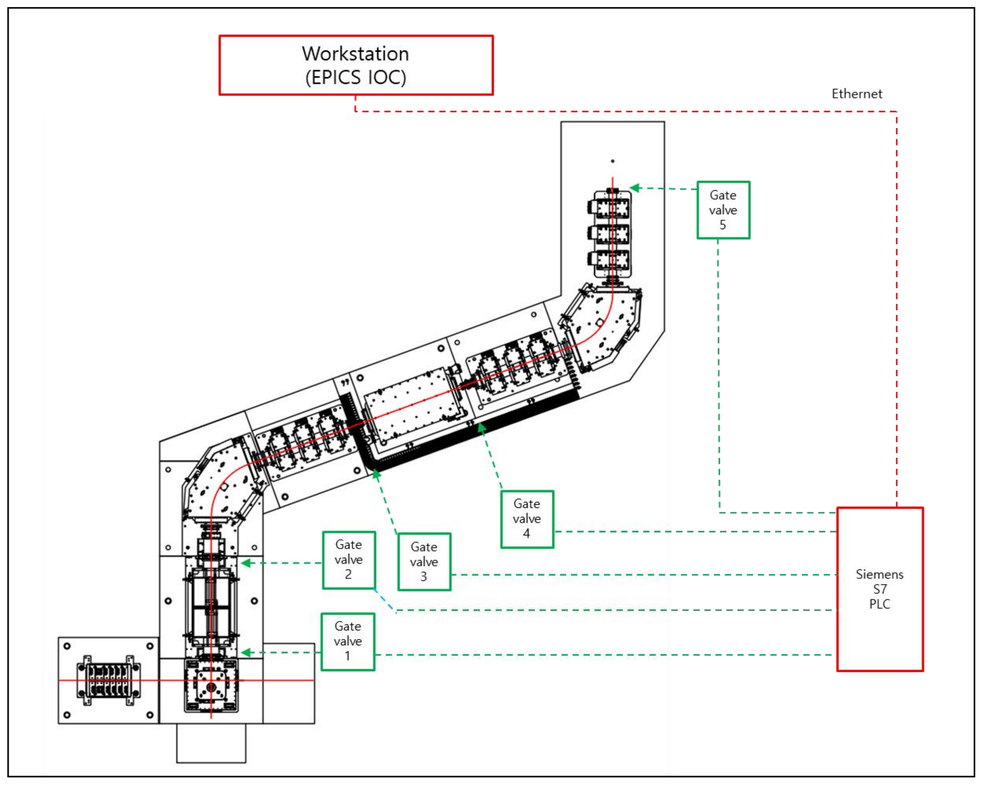

Fig. 3. Gate valve positions in the facility

They are opened or closed when current flows and that current signal is generated by Siemens S7-1500 PLC. We programmed the EPICS IOC which communicates with the PLC in the modbus TCP/IP communication method.

Result & Discussion

In the experiment, the monitoring and controlling coolant system, pump system were done successfully. The graphical user interface was also made and tested as shown in Fig. 4.

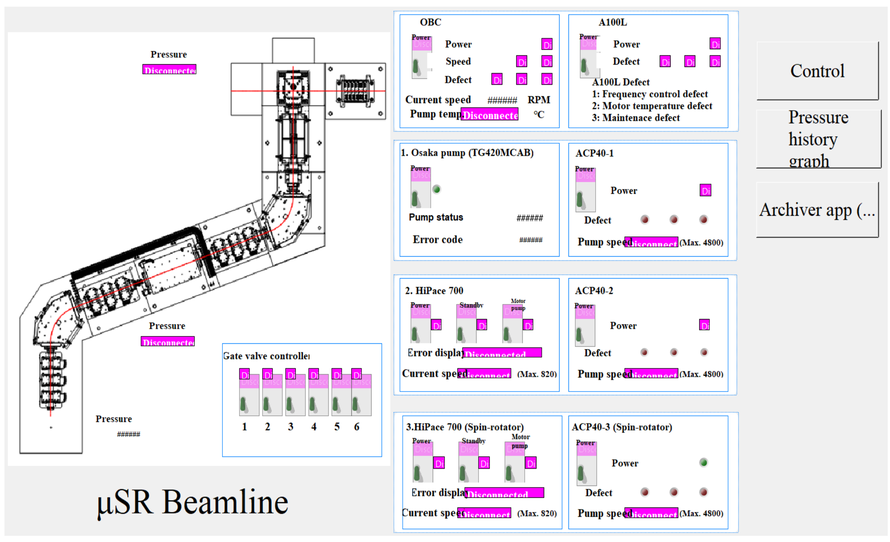

Fig. 4. Graphical user interface made by using CS-Studio

All monitoring data will be recorded in the sub-workstation by Archiver appliance software.

On the other hand, there are some problems to solve yet for the whole system. First, the register in the Arduino module was not collect so the calibration of the value was not perfect. Second, wiring for the remote control of gate valves was not done. Third, the magnet control system was not tested yet due to its late installation. However, we know what should be done to finish them and we expect to finish them within this year.

Reference

[1] PFEIFFER VACUUM, Operating instructions A 100 L Dry Pump (119404EN_03)
[2] PFEIFFER VACUUM, Operating instructions ACP 28-40 (P0568E2) (128852OEN_03
[3] PFEIFFER VACUUM, Operating instructions TC 400 PB, Electronic Drive Unit with Profibus (PT0244BEN_O)
[4] Argonne national laboratory, Experimental Physics and Industrial Control System, https://epics.anl.gov/
[5] Argonne national laboratory, asynDriver: Asynchronous Driver Support, https://epics.anl.gov/modules/soft/asyn/R4-14/asynDriver.pdf
[6] Paul scherrer institut, EPICS StreamDevice, https://paulscherrerinstitute.github.io/StreamDevice/
[7] Argonne national laboratory, synApps: calc, https://epics.anl.gov/bcda/synApps/calc/calc.html
[8] Helmholtz-Zentrum Berlin, EPICS Sequencer, https://www-csr.bessy.de/control/SoftDist/sequencer/
[9] M. Shankar, L. Li, M. Park et al., THE EPICS ARCHIVER APPLIANCE, Proceedings of ICALEPCS2015 (2015)
[10] Paul scherrer institut, S7plc EPICS driver documentation, http://epics.web.psi.ch/style/software/s7plc/s7plc.html