ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-TH013

UPGRADING THE IRRAD CONTROL SYSTEM GUIs USING OPEN-LICENSE AND CROSS-PLATFORM TECHNOLOGIES*

B. Gkotse†[1], A. Abdulhalim, A. S. Mølholm, F. Ravotti
Experimental Physics Department, CERN, Geneva, Switzerland
P. Jouvelot, Mines Paris, PSL University, Paris, France
[1]also at Mines Paris, PSL University, Paris, France

Abstract

The CERN Proton Irradiation Facility (IRRAD) is a reference facility in high-energy physics for the qualification of detectors, materials, and electronic components against radiation. A proton beam with a momentum of 24 GeV/c is delivered from the CERN PS accelerator to IRRAD and impinges on the components being tested, placed on remotely controlled movable stages. This equipment, operated by dedicated control systems, allows for the precise positioning of components in or out of the beam and facilitates the handling of irradiated components, while minimising the radiation received by the IRRAD operators.

Originally, the implementation of the Graphical User Interfaces (GUIs) of the IRRAD control system was based on proprietary software, thus limiting it to specific operating system. To address the issues linked to such dependencies in terms of openness, ease of development and access to state-of-the-art technologies, new GUIs have been designed and developed with open-license cross-platform software. In this paper, the IRRAD control system software architecture is detailed, and the lessons learned while implementing these new feature-rich GUIs are presented.

INTRODUCTION

The Proton Irradiation Facility at CERN (IRRAD) is an infrastructure dedicated to performing radiation hardness testing (irradiation experiments) on detectors, electronics, and materials. The IRRAD facility, located in the East Area of the CERN accelerator complex, receives a proton beam from the Proton Synchrotron (PS) accelerator with a momentum of 24 GeV/c in spills of 400 ms and a Gaussian shape typically 12 × 12 mm^2 FWHM wide [1]. The components that need to be tested (Devices Under Test, or samples) can be placed along the beam trajectory on the top of nine remotely movable stages, called IRRAD tables. The IRRAD tables have three degrees of freedom and can be used to move the samples horizontally (x-axis), vertically (y-axis) or rotate them with an angle (θ) with respect to the beam axis. These tables allow for positioning samples in a volume of up to $20 \times 20 \times 50\,cm^3$ in and out of beam, or performing a scan (e.g. asynchronously move the samples across the beam direction in order to extend the irradiated portion of the samples). Placing or removing samples on the IRRAD tables requires accessing the irradiation area, which can be done only once per week, when the beam is stopped. However, for smaller samples with dimensions up to $5 \times 5 \times 20\,cm^3$, a conveyor system (shuttle) can be used; this can move samples from the outside to the irradiation zone, following a 9m-path without the need of stopping the beam.

During CERN Long Shutdown 1 (2012-2014), a new hardware infrastructure had been put in place at IRRAD and software Graphical User Interfaces (GUIs) had been built, based on CERN-supported proprietary software, for the control of its tables and shuttle. These GUIs were sufficient for operating during CERN Run 2 (2014-2018) [2]. Nevertheless, several restrictions on portability, dependencies on specific platforms and requests for additional functionalities have since deemed necessary the upgrade of the control system GUIs using current open-license and cross-platform technologies. Presenting and discussing this important transition is the subject of this paper.

This article first provides an overview of the hardware components and infrastructure used in the IRRAD control systems. Then, we describe the open-licence and cross-platform technologies used for the development of the new Graphical User Interfaces (GUIs), explaining our software choices and architecture. Details of the new functionalities, software architecture and database schemes are provided for both IRRAD tables and shuttle control systems. Finally, we discuss the lessons learned during this software transition, before concluding and introducing future work.

IRRAD CONTROL SYSTEMS HARDWARE

Since the operation of the IRRAD equipment happens in a radiation environment, the hardware chosen for the tables, shuttle and associated control systems is custom-made, had to ensure radiation tolerance and be easily customizable depending on the experimental user requests. For both systems, some common components have been used but still certain differences remain. Details about the hardware infrastructure are provided in the following paragraphs.

IRRAD Tables

Each IRRAD table uses two stepper motors, one for horizontal movement and one for the rotating axis, while an AC motor is used for vertical movements. Figure 1 shows 3 of the IRRAD tables of the first zone of irradiation in the IRRAD facility. The three motors are controlled using an M300 microprocessor. The communication with the microprocessor is performed through the RS232 serial protocol. Since there are nine of these tables that require this type

* This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 654168.
† Blerina.Gkotse@cern.ch

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-TH013

of communication, Ethernet-to-Serial devices have been installed in the facility, providing multiple RS232 serial ports. This allows computers running the control system software to communicate through multiple virtual COM ports. One control box per IRRAD table containing the microprocessor and control buttons is installed in the IRRAD control room for manually controlling the tables. However, this particular setting requires the presence of operators in the control room and induces some limitations on the type of actions that can be performed.

Figure 1: Irradiation tables (front) and shuttle system (back) in the IRRAD irradiation zone.

Shuttle System

The shuttle system is controlled through two separate motors. The first one is used for the movement of the shuttle over the 9m-long path in or out of the irradiation zone, referred to be the y-axis. The second motor is used for moving the shuttle in or out of the beam trajectory, named x-axis. Figure 2 shows the shuttle outside of the irradiation area and illustrates the two axis it can be moved along. The y-axis uses an AKD Kollmorgen driver [3], and the communication is performed through the telnet protocol. The x-axis has a stepper motor controlled by a M300 microprocessor, and the RS232 serial communication protocol is used, as for the control of IRRAD tables.

SOFTWARE TECHNOLOGY AND ARCHITECTURE

The software technologies used in IRRAD have to be compatible with the hardware located in the facility. Moreover, since IRRAD is a small-scale infrastructure with limited manpower and software expertise, some lightweight and easily maintainable control-software solutions are required. In the first IRRAD run, the control system GUIs that were used were developed using Windows Forms [4] and coded in C++ and C#. Even though the interfaces were considered operational and user-friendly, limitations and dependencies on specific operating system and proprietary software presented some drawbacks (see Section Discussion), which led us to consider upgrading the IRRAD control software.

Figure 2: Shuttle system, part outside of the irradiation area, where the samples are loaded.

Given the fact that the CERN community widely uses LabVIEW [5], SCADA [6] and WinCC-AO [7] for control systems, they were first considered as candidate solutions for our work. However, dependencies on industrial software, not easily maintained by a small team, were a challenge not possible to overcome. Other free and open-source software tools were also discussed such as EPICS [8] and Tango [9]. However, their communities are rather limited, since they are used mainly in high-energy physics, making their learning curve, in the presence of scarce support, too steep. A more lightweight solution was thus considered.

In the following paragraphs, we describe the software technology chosen to upgrade the IRRAD control system architecture and the motivations behind those choices.

PyQt

The PyQt library [10] includes a set of python bindings for the QT application development platform [11]. More specifically, for the IRRAD GUIs use case, PyQt5, released under GNU General Public License (GPL) v3 license, was used. This choice was not only made because PyQt is a free software, but also because it combines the flexibility of Qt for developing fast and interactive user interfaces and the coding simplicity and effectiveness of python.

pySerial

As mentioned in the previous section, the communication with the M300 microprocessor has to be performed serially using the RS232 protocol. For this purpose, pySe-

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-TH013

rial, which is a python module for serial communication management [12], was used.

MySQL

A back-end database was needed for storing several configuration parameters and position history for the GUIs. The well-known open-source MySQL framework was deemed the relational database of choice to host these data [13]. The IRRAD-dedicated database instance is hosted on the Database-on-Demand (DBOD) infrastructure provided by CERN, which allows for an easy back-up and maintenance plan, including regular upgrades [14]. However, the GUIs can be easily configured with a different database, if need be.

Software Architecture

A Model-View-Controller architectural software approach has been adopted for the development of the GUIs, while the application communicates with the SQL database in the back end.

IRRAD TABLE CONTROL SYSTEM GUI

The main requirements for the IRRAD Table GUIs were to allow users to move the samples easily and safely in the required positions. Other important requirements are that users should be able to monitor visually their actions, aware of the position limits and able to intervene any moment by stopping their actions. Moreover, a history of the performed movements and positions selected needed to be kept for logging purposes.

Functionalities

The above mentioned requirements were then translated into the functionalities described in the following items:

- Setting hardware parameters about each motor (e.g., resolution, screw dimensions, offset, etc.);
- Setting communication parameters (e.g., baud rate, parity, COM port, etc.);
- Configuring the sample position, in mm, in the microprocessor memory (since the microprocessor memory is limited, only five specific positions can be configured: park, center, left, right and one that is used for custom positions);
- Moving the IRRAD tables in the configured positions by sending the corresponding commands to the microprocessor;
- Visualising the position;
- Calibrating the motors (since stepper and AC motors are used for the IRRAD tables, a dedicated process for finding the proper correspondence between step numbers and distance in mm, which depends on hardware characteristics such as screw dimensions, is used. For stepper motors, this consists in finding and setting equal to zero the position of a reference switch placed at 0 mm distance and then also placing the motor to the maximum position in mm and with the maximum number of steps).

The user can select to have the full views of the three axes displayed, as shown in Fig. 3, or only one at a time.

Database

Dedicated database tables have been designed and used to save data deemed key for the operation of the GUI and log positions and movements. More specifically, the database tables contain information about the following elements.

Stepper Motor These data contain the stepper motor's hardware settings such as calibration information, resolution, screw dimensions, maximum position, etc.

AC Motor Since AC motors have different characteristics than the stepper motors, a dedicated database table is used for them.

Motor This table contains the names and the motor numbers and is linked to the two previous tables through foreign keys.

Custom Position Custom positions are also stored in the database for each table. In this way, a larger set of custom positions can be saved and used, overcoming the microprocessor memory limitations, by creating visually more buttons with different positions. Nevertheless, in the background, the memory used for storing the custom position is overwritten each time.

Movement The movements are logged in the system for tracing performed actions.

Full documentation and manual for the system can be also found online [15, 16].

IRRAD SHUTTLE CONTROL SYSTEM GUI

Operational safety, precision and usability were the most important requirements for the development of the shuttle GUI, given its operation in a radiation environment.

Functionalities

The main functionalities for the shuttle system are:

- Moving the shuttle in the defined position (Reference, Loading samples, Park and Beam positions);
- Constraining the users on performing certain movements that could affect the precision of the movement (for example, the user should not be allowed to perform certain actions such as going to the Beam position once the shuttle has moved backwards; in that case, the shuttle should move to the Reference position before moving to Beam);
- Monitoring the activity, using two AD6 monitors placed in the Load and Park position in order to comply with safety-related procedures (the data of these monitors are logged and should be controlled before moving the shuttle in the Load position);

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-TH013

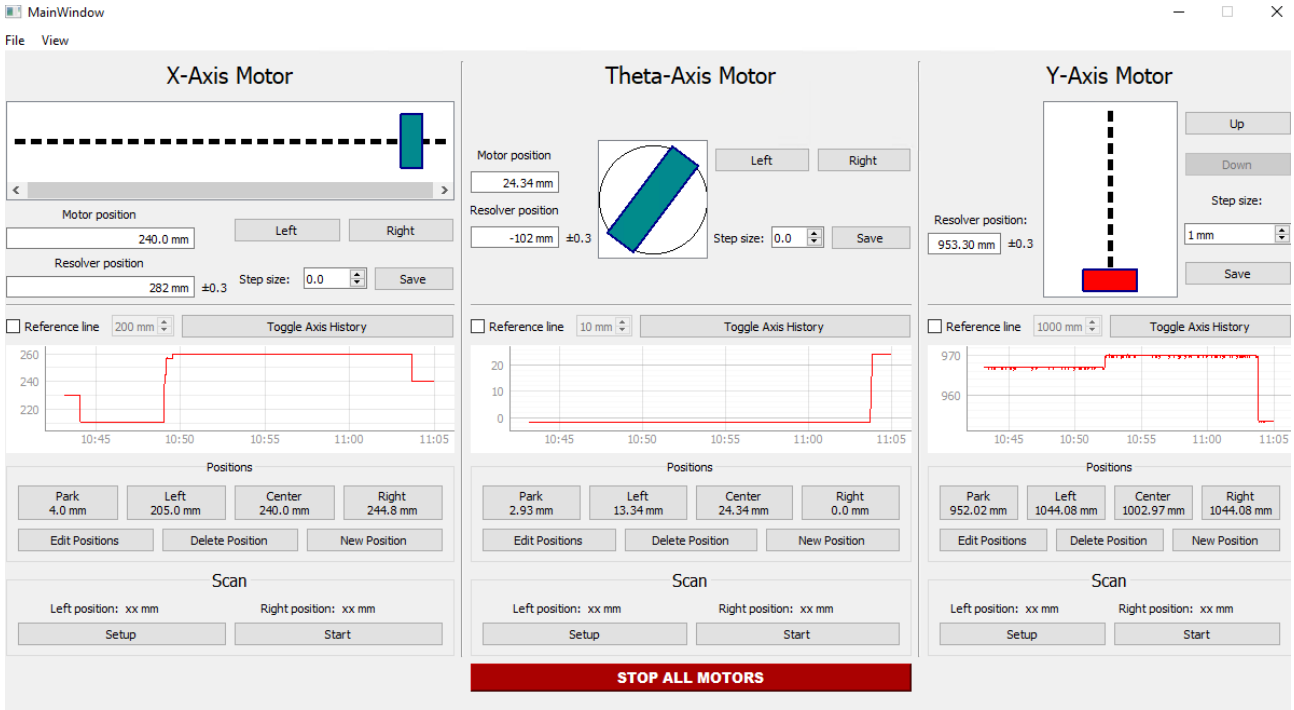

Figure 3: IRRAD Tables Control System GUI.

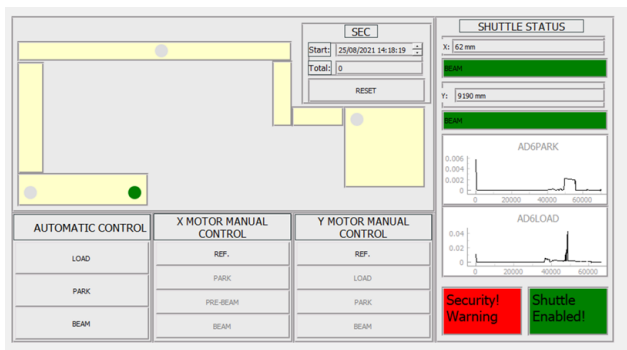

Figure 4: Shuttle GUI.

- Implementing software interlocks, e.g., for not allowing the user to move the table in the Load position if the activity monitored by the AD6 is not lower than a specific limit set by the Radiation Protection authority.
- Visualising on a display the shuttle path (moreover, activity diagrams have been integrated in the interface);
- Monitoring the cumulated proton intensity when the shuttle is in beam position;

A screenshot of the shuttle interface is shown in Fig. 4.

Database

Currently, the data stored in the database for the shuttle system contains the movement history and the cumulated proton intensity for each irradiation run.

Further details and documentation can be also found online [17].

DISCUSSION

Even though IRRAD is a small-scale facility and is thus supported by few software developers, moving its experiment-management software from proprietary to more lightweight, free, open-source and cross-platform technologies turned out to not be very complicated or time consuming. It took a team of two short-term interns under the supervision of one PhD candidate and eight months to come to some workable solutions. The major challenges consisted of designing and implementing advanced functionalities for a robust, safe and user-friendly system that could be easily customisable also by non software-experts. Based on a broad and open-source community and proper documentation [10], finding solutions when problems arose was relatively easy and fast.

Comparing the old C++/C# based and new python-based environments in terms of size, the present choice led to a significant decrease in the number of lines of code. For instance, for the IRRAD Table GUI, we went from about 12,300 lines of code to only 5,500 lines; regarding the Shuttle GUI, the respective numbers are 2,180 and 1,730. This allows for better readability and faster maintenance in case of need. Also, having based the new development on the python language provides the opportunity to extend more easily the code by importing advanced modules and libraries developed by the python community, which is moreover growing larger and larger. Finally, using more recent technologies such as python helps also to find personnel who have the requested software skills and can be trained to work on the project more easily and faster.

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-TH013

In addition, the technology upgrade performed by the IRRAD team is in sync with the CERN IT policy that recommends moving towards open-license software as part of the the license-management MALT project [18], which encouraged the search for alternative free and open-source technology solutions to proprietary ones in CERN projects.

Globally, these technologies have increased the portability and the ease of installation of the GUIs on different operating systems. In comparison to more advanced and industrial control system GUI software frameworks such as WinCC-AO [7] , these interfaces are more lightweight. They seem more suitable for small-scale experiments and infrastructures that can be easily used and maintained by non-IT experts.

Finally, these upgrades were also in line with the needs of the IRRAD facility to cope with a larger demand of user experiments. Traceability and precision were enhanced by the newly introduced logging functionalities such as storing the table movements in a database.

CONCLUSION AND FUTURE WORK

This paper describes the new GUIs developed for the control systems of the IRRAD proton irradiation facility. The hardware components and their uses have been described. For the software development, free, open-source and cross-platform software technologies such as PyQt, pySerial and MySQL have been used and the advantages of this move from proprietary software discussed.

In parallel to these developments, the IRRAD team is working on new beam-monitor detectors and data acquisition systems. These detectors will be installed on the IRRAD tables and used to detect when the tables are exposed to the beam. Therefore, as future work, we envision that the data acquired from these systems could be used to operate a system that would automatically move the tables depending on the beam conditions.

Another future work could include designing and implementing some web interfaces. Since the back end is already developed using python, a python web framework such as Django [19] could be used for this kind of development. In that case, IRRAD system access could be enabled outside of the CERN facility. Therefore, ensuring IT security would be a crucial factor.

REFERENCES

[1] F. Ravotti, B. Gkotse, M. Moll, and M. Glaser, "IRRAD: The New 24 GeV/c Proton Irradiation Facility at CERN", in *Proc. AccApp'15*, Washington, DC, USA, Nov. 2015, pp. 182-187. http://accapp15.org/wp-content/data/index.html

[2] B. Gkotse, M. Glaser, P. Jouvelot, E. Matli, G. Pezzullo, and F. Ravotti, "Towards a Unified Environmental Monitoring, Control and Data Management System for Irradiation Facilities: the CERN IRRAD Use Case", in *Proc. RADECS 2017*, Geneva, Switzerland, Oct. 2017, pp. 1-8. doi:10.1109/RADECS.2017.8696209

[3] AKD Kollmorgen website, https://www.kollmorgen.com/en-us/developer-network/akd-drive/

[4] Windows Form documentation, https://docs.microsoft.com/en-us/dotnet/desktop/winforms/

[5] LabVIEW website, https://www.ni.com/it-it/shop/labview.html

[6] A. Daneels and W. Salter, "What Is SCADA?", in *Proc. International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'99)*, Trieste, Italy, 1999, pp. 339-343. https://jacow.org/ica99/papers/mc1i01.pdf

[7] WinCC-OA Service website, https://readthedocs.web.cern.ch/display/ICKB/WinCC-OA+Service/

[8] EPICS website, https://epics-controls.org/

[9] TANGO website, https://www.tango-controls.org/

[10] PyQt website, https://pypi.org/project/PyQt5/

[11] Qt website, https://www.qt.io/

[12] PySerial documentation website, https://pyserial.readthedocs.io/

[13] MySQL website, https://www.mysql.com

[14] Database-on-Demand website, https://dbod-user-guide.web.cern.ch/

[15] A. S. Mølholm, B. Gkotse, and F. Ravotti, "Python IRRAD Motor Control Application (PIMCA):How it works", CERN, Geneva, AIDA-2020-NOTE-2021-003, 2021, https://cds.cern.ch/record/2750195

[16] A. S. Mølholm, B. Gkotse, and F. Ravotti, "Python IRRAD Motor Control Application (PIMCA):How to use", CERN, Geneva, AIDA-2020-NOTE-2021-002, 2021, https://cds.cern.ch/record/2749936

[17] A. Abdulhalim, "GUI implementation for Controlling and Monitoring of the IRRAD Shuttle System", CERN, Geneva, 2021, https://cds.cern.ch/record/2779934

[18] MALT project website, https://malt.web.cern.ch/

[19] Django website, https://www.djangoproject.com