ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP039

INTEGRATING EPICS CONTROL SYSTEM IN VR ENVIRONMENT: PROOF OF CONCEPT

L. Pranovi†, M. Montis, INFN-LNL, Legnaro, Italy

Abstract

Preliminary activities were performed to verify the feasibility of Virtual Reality (VR) and Augmented Reality (AR) technologies applied to nuclear physics laboratories, using them for different purposes: scientific dissemination events, data collection, training, and machine maintenance. In particular, this last field has been fascinating since it lets developers discover the possibility of redesigning the concept of the Human-Machine Interface. Based on the experience, it has been natural to try to provide to the final user (such as system operators and maintainers) with all the set of information describing the machine and control system parameters. For this reason, we tried to integrate the accelerator's control system environment and VR/AR application. In this contribution, the integration of an EPICS-based control system and VR environment will be described.

PRELIMINARY ACTIVITIES

In our initial investigation [1] for the introduction of Virtual and Augmented Reality (VR and AR) technologies, we used the Microsoft® HoloLens 2 display [2] as an augmented reality (AR) device. This device combines waveguides and light projectors within its enclosure above the brow, utilizing laser light to illuminate the display.

In specialized environments like nuclear plants, the integration of such a device can enhance performance and assist operators in various ways, such as facilitating easy access to information during equipment maintenance or introducing an innovative Human-Machine Interface (HMI) that extends traditional control panels with virtual interfaces (as visible in Fig. 1).

Figure 1: Human-Machine Interface through AR technology for cyclotron supervision.

The primary use case involved the development of a virtual console for operators working with the cyclotron apparatus during its commissioning at LNL. This ongoing renewal process, conducted in parallel with regular operations, aimed to enhance the high-level control interface of the apparatus. In this context, augmented reality (AR) screens are generated alongside the standard control monitors. This setup allowed users to view the critical screens for monitoring and supervising the machine in augmented reality, while the standard computer-based HMI remains available for interaction, including issuing commands and executing procedures.

Conflicting news about the continuity of support provided by Microsoft during 2021-2022 moved our focus on HMI with the employment of VR devices such as Oculus and Valve products.

CONTROLS AT INFN-LNL

The SPES project [3] is currently under construction at INFN-LNL (Laboratori Nazionali di Legnaro) and involves the integration of existing accelerator systems with a new setup comprising the primary beam and the ISOL target. To efficiently control this project, EPICS [4] was selected as the primary Control System framework. Consequently, a transition from the previous control system to the new framework is imperative to adapt the existing system for use in the upcoming facility.

In this moment several systems are under EPICS environment, in particular most of the original accelerator lines though a migration campaign and part of the new upcoming lines which are directly developed in EPICS. However not all the apparatus are EPICS compliant (i.e., cyclotron apparatus – visible in Fig. 2).

As consequence, to reach the goal of extending VR technology to controls supervision using Unreal Engine tool, data exchange between controls and VR environment requires a certain grade of flexibility and the possibility to interface using different protocols.

Figure 2: The 70 MeV cyclotron apparatus used as source for the proton line in the ISOL facility.

† lorenzo.pranovi@lnl.infn.it

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP039

VIRTUAL REALITY AND CONTROL INTERFACE

Integrating an Unreal Engine 5 application with an EPICS control system involves establishing communication between the two systems. EPICS typically uses the Channel Access (CA) and the newer pvAccess (PVA) protocols for data sharing and control. However, this approach is viable only if the system already provides an EPICS interface.

For apparatus not integrated in EPICS, more general protocols can be used for this purpose. One of the candidate solutions we investigated is the usage of the Mobus communication protocol [5], a solution widely used in industrial controls.

In both cases, under Ureal Engine (UE) the idea is to create a dedicated layer to interface the protocol without using a blueprint visual scripting system.

Another assumption adopted for the study is the possibility of only monitoring EPICS Process Variables (PVs) without any write permission.

Channel Access Protocols Interface: Preliminary Studies

Unreal Engine 5 provides a powerful C++ API and interface for programmers to create and extend their projects. The Unreal Engine C++ API and interface allow developers to interact with the engine's core functionality, implement custom logic, and even extend the engine itself. Therefore, a solution through C++ API would be suitable to integrate EPICS communication protocol in the Unreal Engine environment. In addition to that, Unreal Engine 5 Software Development Kit (SDK) runs under the Windows operating system (OS).

According to the EPICS documentation, possible C++ APIs and CA client libraries usable under Windows OS are:

- EPICS Base package
- SCA (Simple Channel Access)

While the usage of the Base package requires the entire EPICS framework installed in the development machine, SCA provides only the minimum set of files needed to implement the communication layer.

Due to the experience in the EPICS framework and the possibility of creating a standalone test bench (I/O controller and custom client application), the base package is suitable.

In this scenario, we only considered the Channel Access protocol (and not pvAccess) because not all the systems in the control architecture are implemented in EPICS version 7 and, therefore, they do not communicate with the newer pvAccess protocol.

Modbus Protocol Interface

Modbus is a widely used communication protocol in the field of industrial automation, particularly in Supervisory Control and Data Acquisition (SCADA) systems, Programmable Logic Controllers (PLCs), and various industrial devices. The Modbus protocol is an open standard that allows for interoperability between different devices and systems. It is also possible to integrate third-party software or hardware into a Modbus-based system.

Considering the idea of realizing an HMI for the cyclotron system which is not integrated in EPICS but provides Modbus communication, a second set of tests and evaluations have been made.

Because of the nature of Unreal Engine 5, the framework does not provide natively an interface to Modbus protocol. For this reason, third-party C/C++ libraries and modules are required. Focusing on open-source solutions, possible options taken into account are:

- Modbus++, a header-only open source C++ class/ library making an encapsulation of Modbus TCP Protocol for client [6];
- Libmodbus is a free software library to send/receive data with a device that respects the Modbus protocol. This library can use a serial port or an Ethernet connection, it is written in C and designed to run on Linux, Mac OS X, FreeBSD, Embox, QNX, and Windows [7].

Due to the purpose of verifying the feasibility of the possible solution based on Modbus data exchange, we preferred to adopt Modbus++ as a candidate for the tests: libmodbus provides a more exhaustive library, but the unavailability of a native C++ layer discouraged the selection of this solution as first choice.

Modbus++ header file provides a dedicated class and a set of pre-defined methods to easily manage communication and data exchange. Considering the idea of realizing only supervision for the apparatus, only read commands have been considered.

Unreal Engine – Application Development

At the Unreal Engine level, the main task is the development of a Virtual Reality-Augmented Reality (VR-AR or XR) application devoted to repurposing the Human-Machine Interface (HMI) of the P70 cyclotron installed at LNL.

The core concept involves providing users with an intuitive set of control panels constituted by specific widgets within Unreal Engine 5, each tailored to represent a distinct data type sourced from EPICS and Modbus communications. The different types of data provided by the two different solutions must be managed to determine the possible widgets required for a proper representation in a virtual UI:

- Graphs and Trends: widgets capable of rendering dynamic graphs to showcase trends in data received from the field. These graphs can be charts or bar graphs, based on the nature of the data received.
- Numeric Displays: widgets to present numerical values retrieved from EPICS or Modbus, allowing users to monitor real-time parameters.
- Status Indicators: incorporate widgets that display the operational status or binary values coming from devices.

Unreal Engine 5 does not have specific built-in packages or modules dedicated to control panels or Human-Machine Interface development. The framework, in general, provides a robust and flexible set of tools for developing

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP039

interactive user interfaces: the entire User Interface (UI) and the single widgets must be built from scratch.

Currently, we are in an exploratory phase, focusing on the definition of a standard UI structure that creates the interface to the Cyclotron apparatus and the proper manipulation of the data coming from the field. The idea of widgets is inherited by the EPICS tool dedicated to control panels, CSS Phoebus.

FURTHER DEVELOPMENTS

The preliminary tests provided interesting results and challenges for a complete interaction between controls and the Unreal Engine framework:

- Under the EPICS-UE5 interface, the integration of the pvAccess protocol is mandatory. In the future all the systems will migrate to EPICS7 and, as a consequence, it will be possible to take advantage of the newer EPICS communication protocol.
- Under the Modbus-UE5 interface, the possibility to use the Modbus protocol can be a versatile solution for all the systems not EPICS compliant and the studies will be not dismissed. Effort will be put into identifying alternatives for the library used with a bigger community.

In general, the security aspect not taken into account in this preliminary study will be investigated, with the aim of extending the interface to actively control systems and apparatus.

CONCLUSION

To extend VR technology to control supervision using Unreal Engine, the interface between the control environment and the high-level graphical framework is a crucial point for the entire application.

The preliminary studies done for the Channel Access and Modbus interfaces in Unreal Engine 5 highlighted the use of a powerful C++ API and third-party C/C++ libraries, and avoiding the coding using the simpler Unreal Engine blueprint method.

The current exploratory phase aims to define a standard UI structure for the Cyclotron apparatus, emphasizing the creation of widgets inspired by EPICS tools. At this moment the entire project is under development and preliminary test, but the upcoming results are very promising.

REFERENCES

[1] L. Pranovi and M. Montis, “Virtual Reality and Control Systems: How a 3D System Looks Like”, in *Proc. ICALEPCS'21*, Shanghai, China, Oct. 2021, pp. 864-866. doi:10.18429/JACoW-ICALEPCS2021-THPV005

[2] HoloLens 2, https://learn.microsoft.com/en-us/hololens/hololens2-hardware

[3] SPES website, https://www.lnl.infn.it/en/spes-2

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

[5] Modbus website, https://modbus.org

[6] Modbus++ GitHub website, https://github.com/fz-lyu/modbusp

[7] libmodbus website, https://libmodbus.org