Prototype development of control and monitoring system for filed instruments using open source technologies with IoT applications

Aniruddh Mali
EC Department
HGCE
Ahmedabad, India
aniruddhkmali@gmail.com

Prof. Mihir Patel
Head of EC Department
HGCE
Ahmedabad, India
mihir.patel@hgce.org

***Abstract*— Open source technologies leverages the freedom of development, beyond the technical limits. In spite of traditional software development platform or environment, open source enables the spark of let's do it. Pyepics is one of the open source tool to develop interactive SCADA system for field instruments. It can also supports open source electronic development boards like raspberry pi, which is a single board computer. Raspberry Pi and pyepics both can be used as an effective outreach set of tools to check/ sense of a slave component, installed at a far distance. In this paper, development of prototype control and monitoring system for field instruments has been explained.**

Keywords— Open source, pyepics, SCADA, Raspberry Pi, prototype

I. INTRODUCTION

Open source technology is defined as the production and development philosophy of allowing end users and developers to not only see the source code of software, but modify it as well. The Linux operating system is a one of the best-known examples of open source software technology. **EPICS** is a set of Open Source software tools, libraries and applications developed collaboratively and used worldwide to create distributed soft real-time control systems for scientific instruments such as a particle accelerators, telescopes and other large scientific experiments. In an effort to help standardize a network communication protocol for large scale scientific experiments, Los Alamos National Laboratory and Argonne National Laboratory collaborated to develop EPICS. Many scientific and industrial organizations worldwide use EPICS and a group of those organizations are charged with maintaining the EPICS standards, documentation and tools. This group of user organizations are also referred to as EPICS.

Large scale scientific applications often require hundreds of devices to communicate over a single network to form large distributed control systems. EPICS provide the standards and tools necessary to make this kind of communication possible. To view more information about EPICS or download the latest documentation and software tools. EPICS applications typically produce large amounts of network traffic, thus the standard calls for a high bandwidth network protocol. EPICS utilizes the TCP/IP based Channel Access (CA) Network Protocol. Channel Access protocol is an application layer built on top of TCP/IP that allows many devices to communicate at high speeds on the same network. Channel Access protocol provides the level of speed, bandwidth and reliability necessary for EPICS applications.

EPICS also implement client/server architecture. Channel Access Servers (CA Servers) can act as real world I/O points through the use of Input/Output Controllers (IOCs). CA Servers publish data to and read data from the network as an EPICS Process Variable (PV). In contrast, Channel Access Clients (CA Clients) monitor the network for updates to process variables. Examples of CA Clients include Human Machine Interfaces (HMI) and data analysis programs.

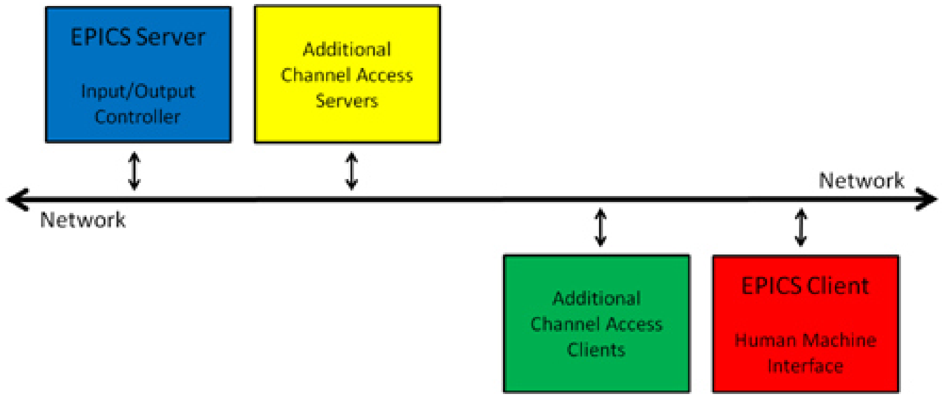

Figure 1 EPICS Network block diagram

Figure 1 highlights the standard EPICS network block diagram. The network is open to having a theoretically infinite number of clients and servers, though this is practically limited by network bandwidth and available IP addresses. All connected devices take turns communicating on the network at high speeds using Channel Access protocol.

II. CONTROL AND MONITORING SYSTEM

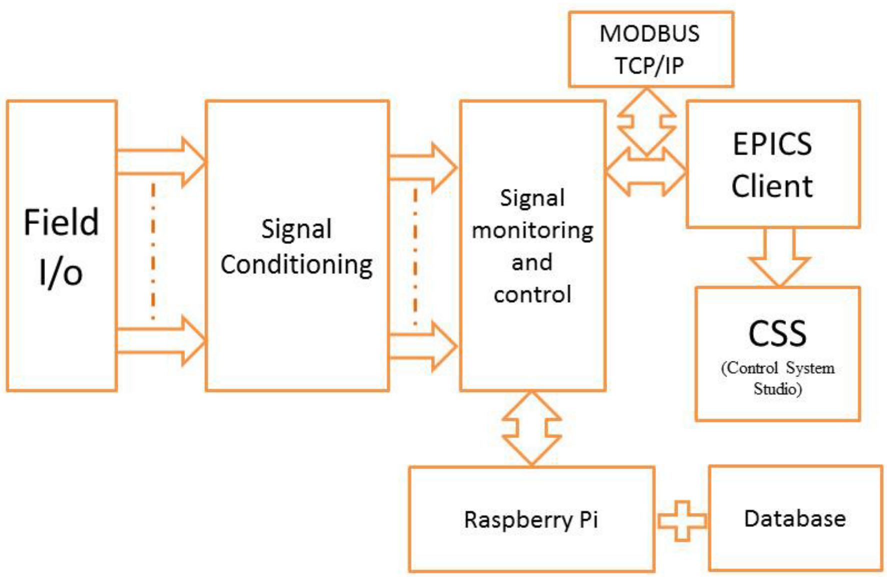

Figure 2 EPICS Network block diagram

Figure 2 shows block diagram of the control and monitoring system for field instruments. Signal receives from filed I/O fed to the signal conditioning block, which keeps the main system protected from any electrical hazardness. Further, signals fed to the monitoring and control section, which checks each signal status and then fed it to the next level. Now, raspberry pi as an embedded controller, will check the health of the received signal and compare it to the already exists database to take further actions, if needed. As it has 1GB of RAM, it can perform very well to provide a high speed data node to the system. Raspberry Pi has EPICS based and pyepics algorithm installed in it, which provide an intelligent master control all over the system. Each field instrument is connected to the existing LAN media, which enables minimal data packet loss into the system.

MODBUS TCP/IP protocol has been used as an universal communication media. A master GUI/ SCADA has been developed using pyepics, which is further deployed on a cloud platform to enable it to provide IoT service around the globe.

III. MASTER SLAVE ARCHITECTURE

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

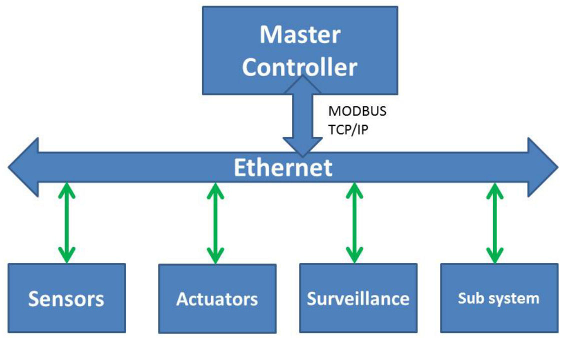

Figure 3 Master Slave Architecture

As shown in figure 3, master controller i.e raspberry pi sends a request to all field instruments (slave). These instruments would response to master controller through MODBUS TCP/IP protocol onto the Ethernet media. As per existing database, controller continuously checks the status/health of each field instrument. It also sends the generated reports to respective nodes periodically. In this project, each slave device is monitored on every 10000 msec. There are different types of slave devices, already installed as part of smart infrastructure for the project site, but it does not have any master control or monitoring system, which can be provided by using the proposed solution for the same. Raspberry Pi will communicate through its Ethernet interface to the slave devices, by following an algorithm identify and check the slave devices. EPICS (Experimental Physics and Industrial Control System) which has been used as control system, uses python for logic implementation. Python is the better choice as it provides both pymodbus and pyepics libraries for single platform interface. Pyepics will be used for EPICS channel access interaction, pymodbus for PLC communication. Control System Studio (CSS) is used for final deployment of channel values which is a user interface framework for EPICS control system based on Eclipse RCP. For monitoring and control of channel values, Control System Studio (CSS) acts as user interface for EPICS process variable interaction using python.

IV. MASTER SLAVE ARCHITECTURE

Figure 4 shows the test set up of the system. It comprises-

A) - Raspberry Pi board is connected to field IO

C) - A power USB hub is connected with it, to manage multiple Data IO coming from filed.

B) - A regulated DC power supply is connected to Master controller, as shown in the figure.

D) – Field IO connection board, developed in house at iCreate.

F) – Field IO connections

We have tested proposed software prototype for more than 24 hours. We have replicated this test setup for different locations at project site. It is working efficiently.

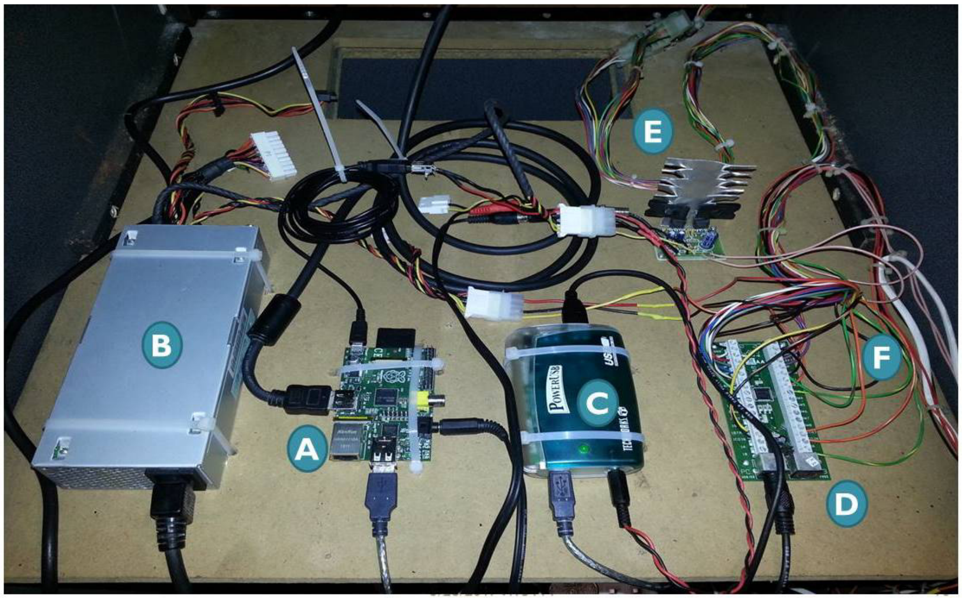

Figure 4 Test Setup

Conclusion

After successful development of prototype, it has been deployed on large area with Master Control integration to all slave devices. It has been implemented in few building block of the campus. Along with it, alarming/ alerts facility has also been implemented for post data analysis using matlab tools. Figure 7.3 to 7.9 shows the implemented smart solutions as part of this project. Alll soutions are working efficiently without fail. However, it needs a stabilized power supply, as it comprises sophisticated electronics inside it.

References

[1] Aniruddh Mali et al. “A new paradigm to access MODBUS supported device using Raspberry Pi and EPICS”, IJICTRD – INTERNATIONAL JOURNAL OF ICT RESEARCH AND DEVELOPMENT | VOL-3 ISSUE-2 | ISSN: 2395 4841

[2] Krupa Mehta at al. “Integration of MODBUS/TCP master monitoring and control system using python for High power RF system”, International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO) – 2015

[3] www-bd.fnal.gov

[4] www.ijcna.org

[5] www.chamber.org.il

