

Design and Implementation of a Web-based Monitoring System by using EPICS Channel Access Protocol

Eun-Mi An*, Yong-Gi Song
PEFP, KAERI, 150, Deokjin-dong, Yousung-gu, DaeJeon, 350-353, Korea
*Corresponding author: aem@kaeri.re.kr

1. Introduction

Proton Engineering Frontier Project (PEFP) has developed a 20MeV proton accelerator, and established a distributed control system based on EPICS [1] for sub-system components such as vacuum unit, beam diagnostics, and power supply system. The control system includes a real-time monitoring and alarm functions. From the aspect of an efficient maintenance of a control system and an additional extension of sub-systems, EPICS software framework was adopted. In addition, a control system should be capable of providing an easy access for users and a real-time monitoring on a user screen. Therefore, we have implemented a new web-based monitoring server with several libraries. By adding DB module, the new IOC web monitoring system makes it possible to monitor the system through the web. By integrating EPICS Channel Access (CA) and Database libraries into a Database module, the web-based monitoring system makes it possible to monitor the sub-system status through user's internet browser. In this study, we developed a web-based monitoring system by using EPICS IOC (Input Output Controller) with IBM server.

2. Structure of the PEFP Control System

Figure 1 shows the architecture of PEFP EPICS IOC web monitoring system. EPICS IOC consists of 5 components, and newly added IOC Web Server is composed of one unit of IBM/X3650 System. Channel Access (CA) is the communication network which allows the IOCs and OPIs (Operator Interface) to communicate. EPICS provides a software component, Channel Access, which provides network transparent communication between a CA client and an arbitrary number of CA servers. EPICS IOC Web server store data every three seconds and periodically stores data of OPI Process Variable (PV). Epics IOC Web server is displayed when the user requires a graphics capability.

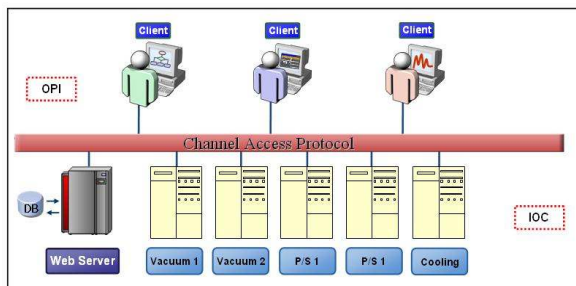


Figure 1. Architecture of web-based monitoring system

2.1 Hardware

- CPU : Processor : Intel Xeon Processor 5160
- Processor Speed : 3.0 GHz
- Main Memory : 2G PC2-5300 CL5 ECC Memory
- Hard Disk : 1TB Hot-swap SATA II HDD
- OS : Linux Redhat 9

2.2 Development Tools

To build a Web-based monitoring server, several software development tools should be installed in advance. The tools installed are EPICS 3.14.8 for a channel accessing, apache server for a web service, Mysql Database for a data storing, PHP for creating dynamic and interactive websites, Perl for a CGI programming, and Gnuplot [2] for plotting both 2D and 3D data.

3. Data Converter Module (DCM)

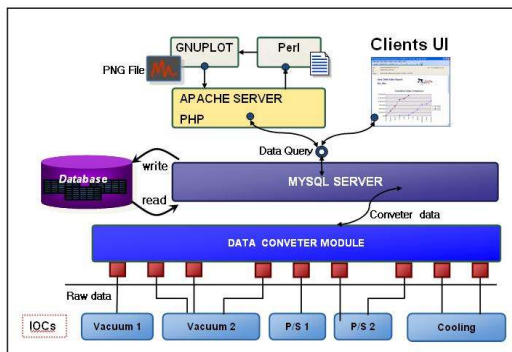
As a first step, the web-based monitoring system is able to present many parameters about the sub-systems. Figure 2 shows the transmission procedure of the original data that are processed in the web server. As shown in the diagram, the PVs data transmitted from all the IOCs are collected by the Data Converter Module (DCM). The DCM is able to access the field data of each record in IOCs and convert them in the MySQL data format. The Converted data from DCM is stored in MySQL databases. The DCM is developed by using EPICS CA and database library. The CA library is as follows:

- caSearch: Locate the IOCs containing selected process variables and establish communication with each one.
- caGet: Get value plus additional optional information for a selected set of process variables.
- caPut: Change the values of selected process variables.

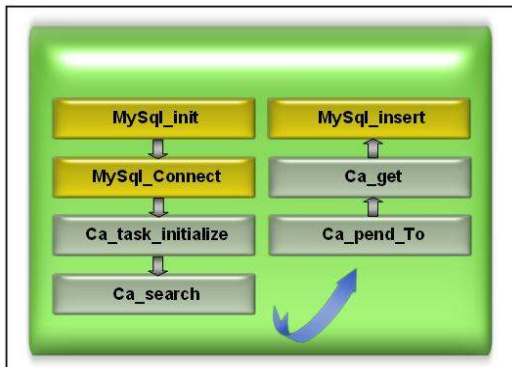
The CA is a protocol used to establish connections to PVs. Once a connection is established, it is possible to read or change PVs values, or monitor their values. DCM based on C language was implemented by using MySQL libraries and CA libraries. The DCM is able to record the access times of CA. Access can be made in every 3 seconds through the DCM, and the PVs

converted in the DCM are periodically forwarded in MySQL database. All clients on Ethernet network can search PV data stored in MySQL databases. For developing web application, we adopted PHP, HTML-embedded scripting language, which allow web developers to write dynamically generated pages quickly. It is implemented by MySQL query instructions through PHP programming. The PHP receives information on user requests, passes them to CA, and generates a HTML page. A graphical plotting viewer of the data searched by web clients is displayed by using the Gnuplot tool. In order to run the Gnuplot binary, another process from the Perl program is spawned.

internet browser via HTML. Figure 3 shows a web browser connection to the web-based monitoring system. Figure 4 shows the input form to search the data range and the graphic image for the data plotting that was searched.



a. Block diagram of Software Structure.



b. Block diagram of DataConverterModule

Figure 2. Block diagram of Software Structure.

4. Implementation of a Web Monitoring System

Gnuplot supports many types of plots in both 2D and 3D. That is, Gnuplot only parse static data and generate the graph. To make attractive looking graphs implemented by using Gnuplot, PHP script including MySQL query sentences access the PV data stored in MySQL and Gnuplot is graphically able to display the searched and responded data in a user's internet browser. To generate a visual graph image from data in a user's internet browser, a web-based monitoring module integrated with PHP, Gnuplot and Perl is implemented. The process interface between the PHP and Gnuplot converts the data file created by APM to a PNG image file and displays a graphical data flow chart in the user's



Figure 3. IOC Web Server Homepage



Figure 4. Search, Plotting Image and Download

5. Conclusion and Future Work

A web-based monitoring system was successfully completed and the functions of search engine and graphical image creation are confirmed. In order to improve the performance of the web-based monitoring system and provide a convenient interface for users, we will continuously upgrade the web monitoring system and study several EPICS extensions that diversify the EPICS based operator interface.

5. ACKNOWLEDGEMENTS

This work is supported by the Ministry of Education, Science and Technology of the Korean Government.

REFERENCES

- [1] EPICS, <http://www.aps.anl.gov/epics/>
- [2] Gnuplot, <http://www.gnuplot.info/>