

EPICS and MDSplus integration for ICRH data acquisition system

Ramesh Joshi

Master of Engineering in Information Technology
 Shantilal Shah Engineering College
 Bhavnagar, India
 rjoshi@ipr.res.in

Manoj Singh

Engineer-SE, ICRH-RF Division
 Institute for Plasma Research
 Gandhinagar, India
 parihar@ipr.res.in

Kiran Trivedi

Associate Professor, Electronics and Communication Engineering
 Shantilal Shah Engineering College
 Bhavnagar, India
 krtrivedi@gmail.com

Abstract— Channel Archiver is a generic periodic sampling toolset for EPICS. Using the EPICS Channel Access (CA) network protocol, it can collect real-time data from any CA server on the network. Ion Cyclotron Resonance Heating (ICRH) Data Acquisition and Control (DAC) system uses the Experimental Physics and Industrial Control System (EPICS) and MDSplus software packages. In, two fundamentally different modes of archiving were discussed: Periodic sampling for historical data logging as opposed to taking a snapshot of data across many channels at a specific point in time for saving and restoring operational set points. The Channel Archiver provides direct data interface between EPICS process variables and MDSplus nodes. Separately, an XML-RPC client was written to access EPICS ‘trended’ data, sampled usually once per millisecond. The client extracts and writes an experiments data worth of trended data to required time with (Modal data system) MDSplus tree. The Channel Archiver acts as a Channel Access Client and stores recorded data, acquired via periodic scan or monitored, into indexed binary files. This paper will discuss the conceptual integration of EPICS and MDSPlus database integration for ICRH data acquisition system.

Keywords-ICRH, EPICS, DAC, CA, MDSPlus

I. ICRH SYSTEM

Ion Cyclotron Resonance Heating (ICRH) is a promising heating method for a fusion device due to its localized power deposition profile, a direct ion heating at high density, and established technology for high power handling at low cost. For the same reason 1.5-Megawatt ICRH system is developed for steady state superconducting Tokamak (SST-1). SST-1 is a large aspect ratio tokamak with wide range of elongation (1.7-1.9) & triangularity (0.4- 0.7). The maximum plasma current is expected to be 220kA. The maximum toroidal magnetic field at the center is 3.0T. The major & minor radii are 1.1 meter & 0.2 meter. 1.5 MW of RF power is to be delivered to the plasma for pulse lengths of up to 1000 second [1]. The block diagram of the ICRH system has shown in the figure-1.

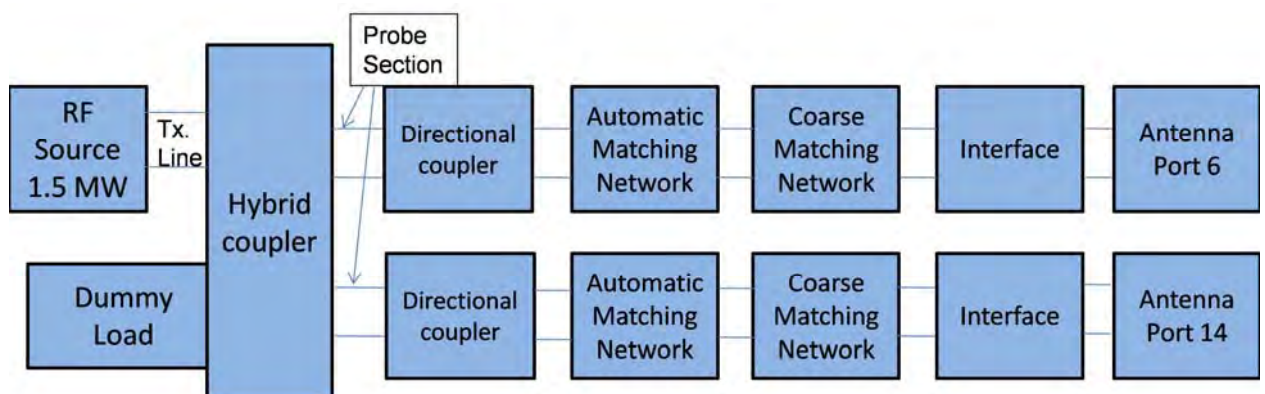


Figure 1. ICRH Block diagram

As we have two different DAC systems whose integrated block diagram has been shown in figure-2. The acquirer of data is also same important with monitor and control. The acquisition of the data during experiment would be done at very fast scale like 1 KHz. At the time of shot the data would be available at the digitizer cards. After shot the data would be demanded by the Linux user interface the socket communication is been established and the packet structure wise data have been available on Ethernet. The processor board will take the data from the digitizer board using messages queue implemented by program. We will take the data from card buffer to processor will be executed and those data would be thrown to network. The data will be acquired (stored) on user interface computer by the GNU socket library program. The acquire data at the slave DAC will be done with the shot number at slave DAC embedded master DAC shot number for synchronization. Shot detail form master DAC would be sent in separate child process because it should not affect the main running process of monitoring and control and master DAC. Same way another thread is waiting at slave DAC would take care for the coming information from master DAC and take action accordingly.

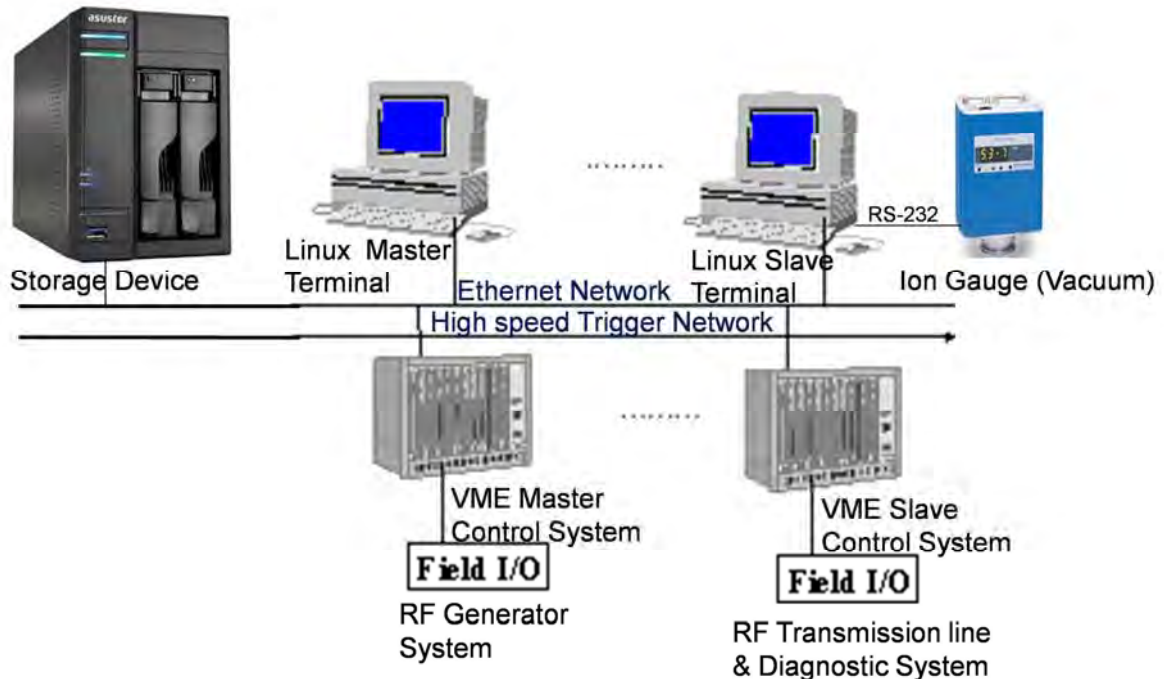


Figure 2. Integrated architecture of Both DAC systems

II. EPICS

The EPICS software was originally designed to be tool based approach to process control and this continues to be its primary application. However, the collaboration recognizes the benefits that might arise from more vigorous development of modular advanced physics modeling and control toolkits that are closely integrated with EPICS. An infrastructure that encourages proper design of distributed software systems is also important. For example, in multi-threaded distributed systems, toolkits need communication software interfaces designed to avoid application programmer introduced mutual exclusion deadlocks. Interfaces must also be properly structured to encourage robust response to loss of communication or other hardware resources. Portability between workstations and embedded systems is an important requirement for certain advanced applications. We expect that they will also be beneficial to advanced modeling and control toolkits. We could able to use the several extensions provided with EPICS like the alarm handler for interlocking and alarming mechanism implementation. User interface design with MEDM which provides motif based user interface. Channel archiver for the storage of data during shot and analysis with Strip tool or data browser. We could also able to use the CSS Best Opi Yet (BOY) for the user interface development which provides the xml based markup language integration with bundled widget options and integration based on Eclipse platform. EPICS provide the mechanism for archiving the data with desired timestamp. The Channel Archiver[2] is a generic periodic sampling toolset for EPICS. Using the EPICS Channel Access (CA) network protocol [3], it can collect real-time data from any CA server on the network. The Channel Archiver acts as a Channel Access Client and stores recorded data, acquired via periodic scan or monitored, into indexed binary files [4].

III. MDSPLUS INTEGRATION

MDSplus is a data management system used in several Nuclear Fusion experiments to handle experimental and configuration data [5,6]. So many application programming interface (API) for local and remote data access are available with different languages, namely C, C++, Fortran, Java, Python, MATLAB and IDL, and a set of visualization and analysis tools are available for data browsing and display. In this way, it is possible to take advantage of the availability of the local and remote data access layers of MDSplus, widely used in the fusion community to handle large sets of data. The MDSplus system has been successfully adopted in many fusion experiments for data acquisition, storage and access [7]. MDSplus provide the hints to achieve continuous data acquisition during the experiments required to use acquired data in the active control of the experiment. We could use Channel Archiver tool for binary data storage of process variable to MDSPlus database. The Channel Archiver is a generic periodic sampling toolset for EPICS. Using the EPICS Channel Access (CA) network protocol [9], it can collect real-time data from any CA server on the network.

There are about 32 analog input channels, 12 analog output channels and 64 digital I/O channels are there at master DAC. We have prepared separate two database files for analog as well as digital channels. The signal naming has been specified at the ICRH:<signal_name>. using softIOC we have broadcasted the PVs. Now make the user interface using state notation language CSS IDE and assign required field widgets with respective process variables. XYgraph has been chosen for monitoring the voltage and current signals. Some python script has been used for the periodic assignment of the channels process variables using caput command. Separate python script is running periodically using execute command function provided on action button click event. In this script we have used pyepics package and import epics as python module and will able to process caput and caput command as per requirements.

We have also used cython package for the load shared object module as dynamic library which was made using C program for socket communication with fast controller. Python package ctypes[10] allows direct linkage to shared libraries without writing wrapper code in C. Using this module we can run python script on action button event for socket communication and reading and writing data as per need. In separate thread we can run this module which will not affect the main monitoring and control process. We have used C program application program interface for data archival in MDSPlus tree nodes. MDSplus provides the ability of having multiple concurrent writers and readers to the MDSplus data files. This enables event driven applications to access the data immediately after each measurement as been acquired and stored in the datafile even as other data is being stored in the same files. We are making the user home directory with tree directory name and would be specified in user profile file by which program get the path of the required tree path name.

A. Reading and Writing MDSplus data

Several data access libraries are available in MDSplus. In particular, one library, called mdslib, is intended to provide easy data access for MDSPlus data. As usual, data access corresponds to the evaluation of a TDI expression, which will specify in most cases the name of a node of a MDSplus tree. When programming in C (and Fortran) things are a bit more complicated than in IDL and Matlab. In the last two languages, in fact, a variable can contain data of any supported type, while in C a variable can only be declared of a given type. The tool named traverse provided with MDSPlus has been shown in figure 3 with ICRH DAC nodes.

B. JScope Tool:

MDSPlus package is provided with the **jScope** tool, a Java application for graphical waveform display. In general, jScope allows the definition of a number of panels, and associated with each panel with one or more MDSPlus expressions (usually signal references), returning an array of numbers, which are then plotted in the panel. Figure-4 shows the jScope screen different signals for 1.5 MW signal acquired with MDSPlus in dummy load test.

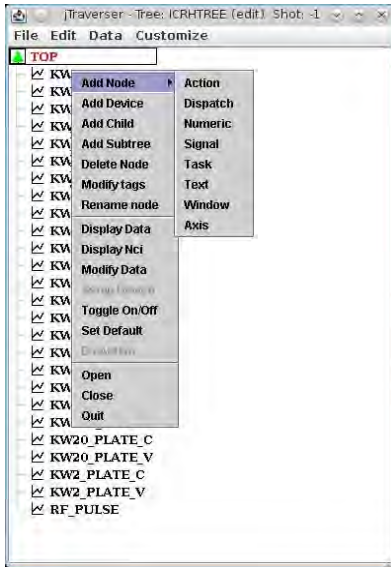


Figure 3. Add new node with different options

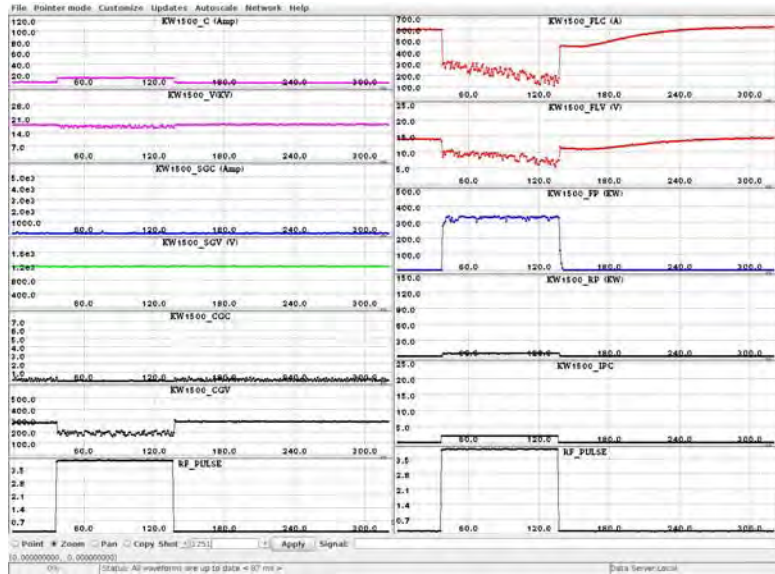


Figure 4. 1.5 MW signal in dummy load test

IV. CONCLUSION

The Channel Archiver has proven to be a useful tool set for collecting and retrieving historical data. It is constantly being extended to meet new user needs, especially in the area of data retrieval. Scripting languages allow arbitrary correlation and other data mining operations, but since these generic languages are not specifically designed for numerical operations and data visualization, Matlab support was added. We are also planning to access MDSPlus directly from Python or Java script which program interface has been provided by the MDSPlus package.

REFERENCES

- [1] J.O. Hill, "Channel Access: A Software Bus for the LAACS", ICALEPCS '89, Vancouver, 1989.
- [2] Manduchi G et al "New EPICS channel archiver based on MDSplus data system" IEEE Transactions on Nuclear Science, December 2011
- [3] Engineering design report (EDR) of ICRH-SST1 SST-2.03-050199-RF Group
- [4] L.R. Dalesio et al, "Data Archiving in Experimental Physics", ICALEPCS'97, Beijing, Nov. 1997.
- [5] J.Stillerman, T.Fredian, The MDSplus data acquisition system, current status and future directions, Fusion Engineering and Design 43 (1999) 301–308.
- [6] T. Fredian, J. Stillerman, G. Manduchi, MDSplus extensions for long pulse experiments, Fusion Engineering and Design 83 (2008) 317–320.
- [7] T. Fredian, J. Stillerman "MDSplus Current developments and future directions, Fusion Engineering and Design", vol 60, 2002, pp 229-233
- [8] G.Manduchi, A. Luchetta, C. Talierno, T. Fredian, J. Stillerman "Real Time Data Access Layer for MDSplus", to appear in Fusion Engineering and Design.
- [9] Ctypes Python Package, <http://python.net/crwe/theller/ctypes/>.
- [10] MDSPlus online reference <http://www.mdsplus.org>
- [11] EPICS Home Page. [Online] Available <http://www.aps.anl.gov/epics>
- [12] XMLRPC Home Page [Online]. Available <http://www.xmlrpc.com>