THE INTEGRATED CONTROL SYSTEM FOR KSTAR

K. H. Kim, M. K. Kim, M. K. Park, J. W. Choi, I. S. Choi, M. C. Kyum, M. Kwon
and the KSTAR Control Team

Korea Basic Science Institute, 52 Yeoeun-Dong, Yusung-Gu, Daejeon 305-806, Republic of Korea

ABSTRACT

The Korea Superconducting Tokamak Advanced Research (KSTAR) control system will be developed as a network-based distributed real-time control system composed of several sub-systems. The central control system includes discharge control, machine control, diagnostic control, timing, and interlocks. There are also local control systems for various sub-systems. We are planning to integrate the entire system with several networks including a reflective memory based real-time network, an optical timing network, a Gigabit Ethernet network for generic machine control, and a storage network. The Experimental Physics and Industrial Control System (EPICS) has been chosen as the middleware of KSTAR control system because the EPICS framework provides network-based real-time distributed control, operating system independent programming tools, operator interface (OPI) tools, archiving tools, and interface tools with other commercial and non-commercial software. However, while EPICS can provide for and solve a variety of problems, there still remain some unresolved issues for the integration of both hardware and software. From the hardware point of view, we have to consider supporting Peripheral Component Interconnect (PCI) Extension for Instruments (PXI) and Compact PCI (cPCI) systems which are not popular in the EPICS community but are required by machine control and physics diagnostics in KSTAR. From the software point of view, we have to make an interface between EPICS and some vested legacy codes in the Nuclear Fusion community, i.e. Model Driven System Plus (MDSplus) from MIT and the Plasma Control System from General Atomic (GA). We will present the details of the integration issues and also will give a brief summary of the entire KSTAR control system from an integration point of view.

INTRODUCTION

The integrated control system for KSTAR [1] should be categorized as a network-based distributed control system, since the entire control system for KSTAR will be composed of several subsystems classified by their functions in the entire system. The subsystems will be connected through networks and will make highly systemized interaction with one another. The integrated control system will be realized by the connections and by the interactions among subsystems as mentioned above. To develop an extensible and reliable control system, the use of verified development tools and middleware is required. These can provide us with a framework for making a distributed control system and can provide software facilities such as a standard communication protocol, processing databases, bundled hardware drivers, etc. We have chosen EPICS [2] as the standard development tool and middleware for the KSTAR control system. Thus, we have studied how the EPICS system can be used for the KSTAR control system and which features in EPICS can be used for specific requirements in the KSTAR control system.

STRUCTURE OF THE KSTAR CONTROL SYSTEM

The KSTAR integrated control system is composed of the central control system (CCS), which provides us with the capability of performing supervisory control for the entire system and many local systems. The central control system contains the plasma control system (PCS), machine control system (MCS), diagnostic control system (DCS), time synchronization system (TSS), and interlock system (ILS).

These systems are connected by several network systems: the machine control network based on Ethernet, the timing network based on optical fibers with KSTAR's own timing network protocol, the reflective-memory-based real-time network (RTNet), and an interlock network based on optical ControlNet™[3]. Figure 1 shows the structure of the KSTAR control system.

The functions of CCS are the following.

Supervising for entire system: The system provides operator interfaces (OPIs) for man-machine interface (MMI) and for performing the continuous monitoring and control for entire systems. EPICS OPI tools can be used to develop MMIs. It also provides the supervisory functions for machine control and plasma control. The supervisory functions provide the information exchange between entire underlying systems and the operating procedure for autonomous plant operation. The channel access (CA) protocol will be the standard protocol for the information exchange and the state notation language (SNL) will be the solution to implement operation procedures for the autonomous sequential operation. The core of the supervisory control is the SNL code and it is located in one of the Input/Output Controllers (IOC) in the CCS.

Plasma discharge control: There are three systems in CCS involving the plasma discharge control. The DCS provides the diagnostic information about the plasma for the PCS. The PCS performs real-time plasma control algorithms and provides the results of the control algorithms to the MCS. The MCS activates actuators such as magnet current supplies, heating systems, and puffing systems. These are required, dedicated, and very low latency communications. We are going to use RTNet for communication in the discharge control.

Data acquisition: There are two kinds of archiving systems required. One is an archiver for machine operation. It would be used for low data/events rate and a large number of channels. Actually, a CA archiver seems to be the best solution for this archiving system. The other archiver is for diagnostic and experimental data. It would be used for large amounts of data and a smaller number of channels as compared to the machine operation. We are considering MDSplus [4] for this archiver, since MDSplus has been used in the fusion community and it also supports many legacy codes. Actually, the MDSplus has an archiving feature as well as other features (for example, a dispatcher which can be used for controlling and arming digitizers in a diagnostic system). But we do not use these features in MDSplus; only the archiving features will be used. EPICS will control the diagnostic digitizers directly to simplify the structure of the system. We have a plan to develop the native MDS protocol, MDSip supports on EPICS, which perform data transfer from data acquisition devices to the MDS server through Ethernet to archive diagnostic data and work as EPICS driver supports.

Networking and interface: The machine control network is based on Ethernet. It delivers CA protocols which are standard protocols in the KSTAR control system to transfer machine control data. The OPIs and machine control data archiver also use the CA protocol. The CA gateway will be involved in the machine control network. The CA gateway is working as a proxy server, so it reduces the number of CA connections for individual IOCs. The CA gateway also provides an access control feature. We are planning to use the host base access control to give security for the control system. The diagnostic systems are connected to the MDS server though Ethernet, which is separate from the machine control network, to archive experimental data into the MDS server. The optical timing network delivers 100 MHz resolution timing clock and event information for the timing system. The RTNet is based on an optical ring network and VMIC5565 reflective memory modules. It has very low latency (< 0.4 μs/node) and high data thru-put (>174 Mbytes/s). So, we plan to use this network for the PCS system to deliver plasma diagnostic data and actuator control data. An interlock system based on a Programmable Logic Controller (PLC) system uses a separate interlock network, the optical ControlNet™. This network can be accessed by EPICS through EtherIP™ communication modules, which make conversion between ControlNet and User Datagram Protocol (UDP) based EtherIP. EPICS has driver supports for the EtherIP protocol. So we can easily combine the interlock system into the integrated control system.

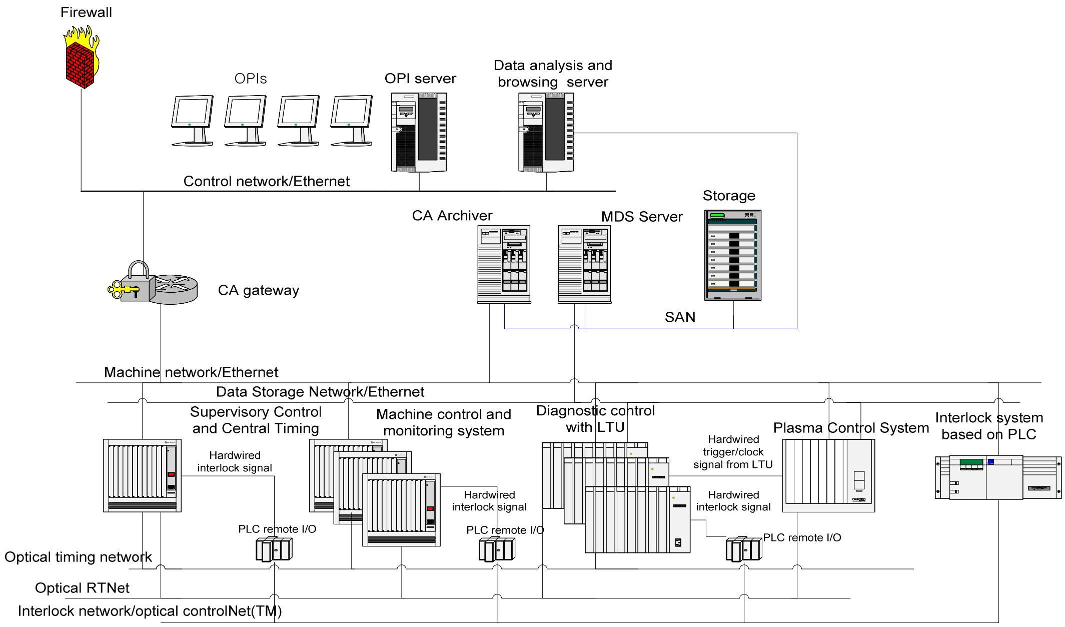
Firewall
OPIs
OPI server
Data analysis and browsing server
Control network/Ethernet
Storage
CA Archiver
MDS Server
CA gateway
SAN
Machine network/Ethernet
Data Storage Network/Ethernet
Supervisory Control and Central Timing
Machine control and monitoring system
Diagnostic control with LTU
Hardwired trigger/clock signal from LTU
Plasma Control System
Interlock system based on PLC
Hardwired interlock signal
Hardwired interlock signal
Hardwired interlock signal
PLC remote I/O
PLC remote I/O
PLC remote I/O
Optical timing network
Optical RTNet
Interlock network/optical controlNet(TM)

Figure 1: Structure of the KSTAR control system

MACHINE CONTROL SYSTEM

The MCS performs monitoring and controlling for various sub-systems closely related in the tokamak operation such as the magnet power supplies, magnet monitoring system, structure monitoring system, cryogenic control system, vacuum pumping system, gas control system, and auxiliary heating systems. The MCS is distributed on many of local IOCs and has several different layouts depending on system requirements.

Some of them are developed on vxWorks/VMEbus systems: The Magnet Power Supply (MPS) system is one example because this system is directly connected to PCS through RTNet and is involved in real-time feedback control. The PCS will generate network interrupts on RTNet when it finishes the calculation of new control values, and then the MPS has to handle the interrupts and data from the PCS within a few tens of microseconds to keep PCS control time slots. Thus it requires a real-time operating system such as vxWorks. It also needs Local Timing Units (LTUs) to keep time synchronization and for catching timing related events from supervisory control. Now, the KSTAR timing system which contains the Master Timing Unit (MTU) and LTUs supports VMEbus system only. Those two are the reason for choosing vxWorks/VMEbus for the MPS. Now, we are developing an EPICS driver for RTNet on the vxWorks environment to adopt it in the system.

Some sub-systems of the MCS are developed on softIOC in the Linux/PCI platform without real-time features. These are using legacy I/O (RS-232, 422, 485), General Purpose Interface Bus (GPIB), etc. One example is the vacuum pumping system. Almost all vacuum gauges and pump controllers support RS-232, 422 serial communications and the system requires some digital I/O (DIO) channels to control and to monitor valve positions. It also requires some analog I/O (AIO) channels to monitor the utility status such as dry air pressure, chilled water temperature, water pressure, etc. A terminal server which is a Serial2Lan converter can be used to handle RS-232, 422 serial communications and to convert them to transmission control protocol (TCP) communications. CompactFieldPoint (cFP) which is operated by our own RTLabView program to make TCP communication with EPICS can be used to handle DIOs and AIOs. The previous two can provide communication with asyn driver supports in EPICS. We made Instrument dev/drv supports on the asyn driver layer for each individual instrument. So, EPICS can completely control many of devices in the vacuum system though TCP communication on softIOC.

Some sub-systems of MCS are developed on the Linux/PXI platform. Even those are not required real-time features but have to handle fast data rates and a large number of channels with a variety of I/O modules. The PXI system can provide variety I/O as well as various commercial signal conditioning modules such as Signal Conditioning eXtension for Instrumentation (SCXI) and also provides sufficient CPU power. We are planning to adopt this system for the magnet monitoring system and structure monitor system in MCS. Usually, the EPICS bundled drivers do not support PXI hardware. Thus we have to develop PXI hardware drivers by ourselves. We have developed EPICS supports for PXI hardware named genericPXI which can support almost any National Instrument PXI hardware.

PLASMA CONTROL SYSTEM

The KSTAR plasma control system is developed on the Digital Signal Processor (DSP) based on the cPCIbus system. This system has a Pentium based Single Board Computer (SBC) which is located on cPCI Slot0 to work as the bus controller and it is operated by Linux. The system has a DSP farm installed on cPCI carrier board located on the next cPCI slot from the SBC. The SBC performs communication with other systems, providing an execution environment for DSPs on the DSP farm, task management for the DSPs, any other house keeping processes for the PCS such as file I/O, etc. Those functions mentioned above will be developed on EPICS as dev/drv supports, namely EPICS PCS device supports, and then EPICS can control the DSP farm and everything in the PCS. The SBC has other EPICS supports, namely MDSip supports, to make communication with the MDS server which will store physics experimental data such plasma position and plasma shape related parameters calculated by the PCS. The DSP farm has four DSP chips – Tiger Shark 250MHz – and performs the plasma control algorithms in parallel on each DSP. It can be easily expanded by adding more DSP farms, if we need it. This system also has a D-TACQ high performance digitizer which has 96 channels of high resolution (16 bits) fast ADCs (250 kSamples/channel) to get magnetic probe data directly. The PCS has a VMIC5565 RTNet interface on the cPCI carrier board which will deliver control data from the DSP farm to the MPS. Actually, when the DSP farm finishes the calculation from the magnet probe data, it then writes all of the control data into the RTNet to provide it to the MPS and makes the network interrupt on RTNet to notify the MPS that it has new data. Figure 2 shows the structure of the KSTAR PCS.

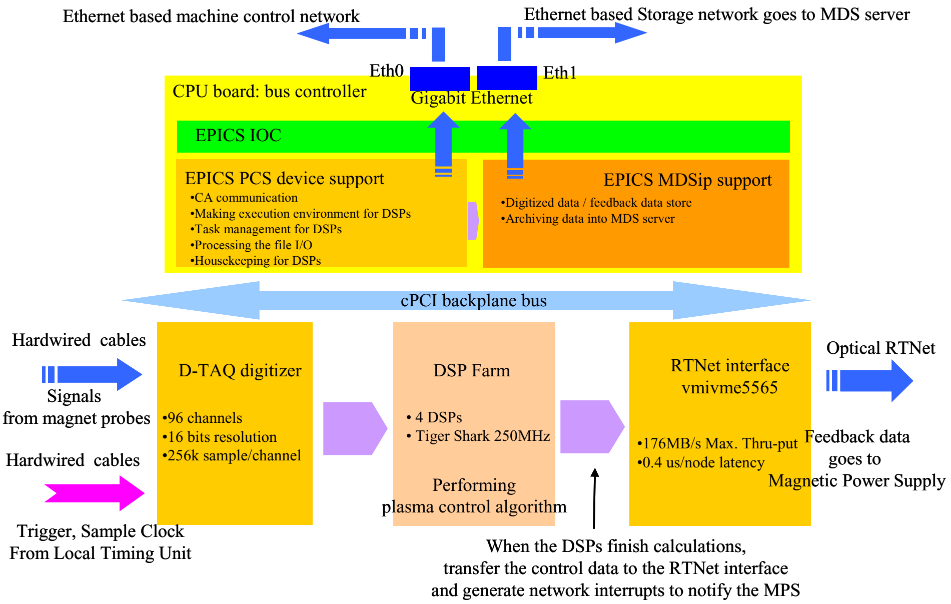

Figure 2: Structure of KSTAR PCS

TIME SYNCHRONIZATION SYSTEM

The timing synchronization system consists of two kinds of VME modules: MTU and LTU, optical based timing networks, and a Global Positioning System (GPS) receiver. The GPS receiver provides 10 MHz timing clock and local time for the MTU though the Inter-Range Instrumentation Group-B (IRIG-B) protocol. The MTU is located on a VMEbus chassis in the MCS and generates 100 MHz master timing clock pulses which are synchronized with the GPS time clock pulses by a phase locked loop (PLL) and distributes the master timing clock pulses to individual LTUs in sub-systems through the optical timing network. The MTU also makes timing related events and distributes them to the LTUs. The timing network protocol delivers timing related events and master timing clock pulses to the LTUs. This protocol is being developed by ourselves and implemented on a Field Programmable Gate Array (FPGA) in a timing module. The LTU provides sample clock pulses and triggers for individual equipment in sub-systems through hard wired connections or through the VMEbus interrupts. We have developed an EPICS driver for the MTU which performs the following two functions: (1) It provides hardware initialization and configuration functions for the MTU and also provides a software interface between the EPICS IOC core and the MTU. (2) It serves Network Time Protocol (NTP) for TCP based time synchronization. So it works as an NTP server and the master clock time is read from the MTU. We have only developed the timing modules for the VMEbus system, however the LTUs have to service other bus systems such as PXIbus and cPCIbus. In that case, the LTU which is located near the instrument devices, but is not in the same chassis with the devices, can distribute timing related signals – sampling clock pulses, triggers – directly to the instruments through hardwired cables, and the VME chassis which contains the LTU takes charge of the timing service for its neighbours. We are going to develop EPICS driver supports for the LTUs. They support the EPICS timestamp, making soft events from LTU VMEbus interrupts, etc. We have a plan to make EPICS supports which are related to the timing system for PXIbus. As we mentioned above, the PXIbus system does not have its own timing system but is serviced by a neighbour VMEbus. However, the PXI instruments could require back plane trigger, clock, and multi chassis synchronization. In that case, we can use PXI slot 2 trigger controllers, which provide multi chassis synchronization, external clock pulses, external triggers, routing signals, etc. This module also requires EPICS driver supports and we have a plan to develop them.

INTERLOCK SYSTEM

The interlock system is based on a PLC system and it monitors the entire system in KSTAR, even the supervisory control system in the CCS. Usually, almost all interlock signals from the sub-systems are collected through hard wired cables. If a sub-system is physically isolated from the interlock system, we will install remote I/O modules near the sub-system to collect interrupt signals and to connect it to the interlock system CPU module through the optical based ControlNet. If a sub-system does not have a hard wired interlock signal because it is a softIOC, the interlock system could monitor a heart beat Processing Variable (PV) in the system. If the heart beat did not flip and flop, the interlock system would assume that the system was down. The connection between the interlock system and EPICS uses EtherIP protocol. It is a UDP to communicate between PLC systems through Ethernet. The EPICS has a bundled driver to understand the EtherIP protocol. We can put the EtherIP module in a PLC chassis. EPICS monitors variables in the PLC CPU and also monitors ControlNet communications, because the EtherIP module can convert the ControlNet protocol to EtherIP protocol. EPICS can assign a PV for a tag, which is a variable in the PLC system and can control it in the same manner as a native PV without additional effort.

INTEGRATION STRATEGY

There are some strategies for choosing software and hardware platforms to make easier integration with EPICS.

For the read-time features that are required, vxWorks/VMEbus system is a faultlessness choice. We already chose this platform for the MPS and the TSS. This solution would allow us to use various bundled hardware drivers in EPICS. But, the VMEbus has a restricted bus speed (in the worst case ~13 MB/s at a single word transfer), so, it is not suitable for handling heavy data rates. The cPCIbus and PXIbus would be an adaptable choice for that case.

For a simple fast monitoring without real-time control the Linux based PXI system is a possible choice corresponding to the requirements. If archiving for MDS is required, we can use MDSip driver supports with the genericPXI.

There is encouragement for using softIOC. When, it is possible to use remote I/O solutions, such as cFP and the Ethernet converter for legacy I/Os, softIOC is the cheapest solution. We now can use soft IOC with SNL and with remote I/O, instead of PLC. The SNL is a native feature of EPICS; it has a powerful state programming language, deep coupling with EPICS, simplified development, easier maintenance, and it relieves the need for a PLC engineer. Moreover, the cFP remote I/O costs less than PLC modules. The above strategies are presented in Table 1.

Application Category	Data rate	Bus type	Operating System	Description	Applied system
Real-time	High	VME	vxWorks	Faultlessness choice for a real-time system, various bundled hardware support in EPICS	Timing System Magnetic Power Supply Some of diagnostic systems
	Extremely High	cPCI	none	DSP system	Plasma control system
Interlock system	-	-	PLC	Interlock network: controlNet™ (optical) EtherIP™ (Ethernet)	Interlock system
non Real-time	Low	-	PLC	Turnkey system, EtherIP™ (Ethernet)	Cryogenic System
	Low	PCI	Linux	softIOC and remote I/O, cost effective solution	Vacuum system
	High	PXI	Linux	Simple monitoring and heavy data rate, genericPXI required (almost completed now)	Some of diagnostic system, Magnetic monitoring system, Structure monitor system
	High	VME	Linux	vmeUniverse rquired (almost completed mow)	Candidate for monitoring systems We have provided vmeUniverse supports for J-PARC project in Japan.

Table 1: Strategies for choosing software and hardware platforms

There are some technical issues for developing a control system with EPICS.

- ✓ Implement the elementary control algorithm on EPICS database: data conversion, proportional, integral, derivative (PID) control, simple calculations, and control.
- ✓ Utilize SNL program for more complex control: sequential control, complex control algorithms, and C embedded programming.
- ✓ Utilize event driven features: EPICS provides useful features for event driven programming in database and in SNL. We can use it to make more efficient codes.

SUMMARY AND FUTURE PLAN

After choosing the EPICS as the middleware for KSTAR control, we have developed low level hardware support to adapt it for system requirements and integration strategy. We already made various hardware supports but still need to develop cPCI supports for PCS and MDSip supports for experimental archiving. We also need to obtain a clearer understanding of tokamak operation to implement the integration control system for KSTAR.

ACKNOWLEDGEMENTS

This work was supported by the Korea Ministry of Science and Technology. We also acknowledge substantive technical discussions with A. C. England, KBSI.

REFERENCES

[1] G. S. Lee, et al., The design of KSTAR tokamak, Fusion Eng. Des. 46 (1999) 405-411

[2] http://www.aps.gov/epics

[3] M. Kwon, et al., The control system of KSTAR, Fusion Eng. Des. 71 (2004) 17-21

[4] T. W. Fredian, J. A. Stilleman, MDSPlus Current developments and Future directions, Fusion Eng. Des. 60 (2002) 229-233