EXPERIENCES AND LESSONS LEARNED IN TRANSITIONING BEAMLINE FRONT-ENDS FROM VMEBUS TO MODULAR DISTRIBUTED I/O

I. J. Gillingham, T. Friedrich, S. C. Lay, R. Mercado,
Diamond Light Source, Oxfordshire, UK

Abstract

Historically Diamond's photon front-ends have adopted control systems based on the VMEbus platform.

With increasing pressure towards improved system versatility, space constraints and the issues of long term support for the VME platform, a programme of migration to distributed remote I/O control systems was undertaken.

This paper reports on the design strategies, benefits and issues addressed since the new design has been operational.

INTRODUCTION

Diamond Light Source is a 3 GeV third-generation light source with a 561m storage ring (SR), a full-energy booster (BR) and a 100 MeV pre-injector Linac [1]. The photon output is optimised for high brightness from undulators and high flux from multi-pole wigglers. The current operational state includes 28 photon beamlines, with a further two beamlines in an advanced stages of design and construction.

In 2010, Diamond adopted a new control system design for future beamlines, adopting fieldbus distributed control as a viable alternative to the long standing VMEbus architecture [2]. In planning for the present phase of photon beamlines, in 2015, the control system architecture and choices of hardware used was revisited.

PREVIOUS CONTROL SYSTEM ARCHITECTURES

The original accelerator and beamline control systems at Diamond are based on VME systems. To support the interface requirements of the equipment, a range of I/O modules based on Industrial Pack (IP) modules (ADC, DAC, Serial, DIO) and VME modules (IP carrier, motion, scalar and timing) were used. The field signals are interfaced via either rear transition modules or front-panel connections. A VME microprocessor (MVME5500) runs VxWorks and EPICS to serve up the control information to client applications.

REASON FOR CHANGE

The original control system architecture served well for the existing accelerators and beamlines; however it was defined a number of years ago, so in the context of the recent and future phase of beamlines the opportunity to reconsider the standards was taken. In doing so, it is clear that not all the hardware capability of VME was required for beamline control; neither was the use of a hard real-time operating system such as VxWorks. It was also apparent that most I/O functionality required for control of beamline and accelerator equipment can now be realised through Ethernet-attached I/O. There is also now good infrastructure for developing and managing Linux based EPICS IOCs on a PC architecture.

OUTLINE REQUIREMENTS FOR PHOTON BEAMLINES

In considering the requirements for photon beamlines, the following technical systems were identified:

- Motion control
- Vacuum instrumentation and other serial devices
- Video cameras
- Analogue and digital signals
- Programmable logic controllers
- Timing signals

The disturbance to beamline operation through restarting an IOC, should be minimised.

The I/O associated with the control system should be located close to the equipment being interfaced; i.e. for signals located in experimental and optics hutches, the I/O modules should be co-located in these areas. However, this is constrained by the possibility of radiation-induced damage to I/O in the optics hutches of high energy (~100keV) beamlines or Storage Ring (SR) vault and by the space available in the some beamline hutches.

NEW CONTROL SYSTEM ARCHITECTURE

Each EPICS IOC runs on a 1U Linux PC located within the relevant Controls Instrumentation Area (CIA). The Linux servers have several physically separate network connections to support the different systems.

Linux Server

The IOCs are standard Dell 1U machines (currently R320), without a hard-disk installed, that boot from an image over the network into a RAM based disk.

The current operating system is 64 bit RedHat Enterprise Linux 6, which has real-time extensions compiled in to facilitate determinacy for the EtherCAT driver, when used at high bandwidth.

On boot, the server picks up its host-name and IP address via DHCP, determined by the MAC address of eth0. The DHCP server also points to the TFTP server which provides the kernel and ramdisk image used to boot the machine.

A startup-script in the host's soft IOC directory is used to configure the server. For instance, network interface

ISBN 978-3-95450-148-9

eth1 could be assigned an IP address, the kernel-module for the event-receiver card could be installed and the EtherCAT service started. This gives users the ability to configure the machine using a version controlled script.

Remote I/O Interfaces

In order to manage various classes of signal associated with control and acquisition, two distinct types of interface have been adopted, PLC Remote I/O (PLC RIO) and EtherCAT. The following table lists and summarises the attributes of each type of interface:

Table 1: RIO Interface Types

Interface	Attribute	Description
PLC RIO	Signal processing	Ladder logic in PLC generates machine protection interlock permits, in rapid and predictable response to input signals
	Interface with IOC	Ethernet or serial communications with slow bandwidth in the order of 1Hz. Sharing of field signals between the PLC and IOC is facilitated by mapping in PLC memory and reading it periodically from a client (e.g. IOC) using the FINS protocol.
EtherCAT	Signal processing	Various signal categories are supported for devices not necessarily associated with machine protection, i.e. for monitoring only. Analogue signals may be sampled at rates exceeding 10kHz.
	Interface with IOC	The EtherCAT network is scanned by a dedicated service on the IOC's host Linux server and readings made available to one or more IOCs via network sockets.. This can take advantage of preemtive real-time kernel extensions when high bandwidth, determinate signal processing is required.

NETWORKS

The servers are supplied with four network interfaces. This is to facilitate multiple dedicated and isolated networks, typically three are used as:

- Primary control network
- Instrumentation network (private to the server and network connected peripherals)
- EtherCAT network

The context of key network components is shown in Fig 1.

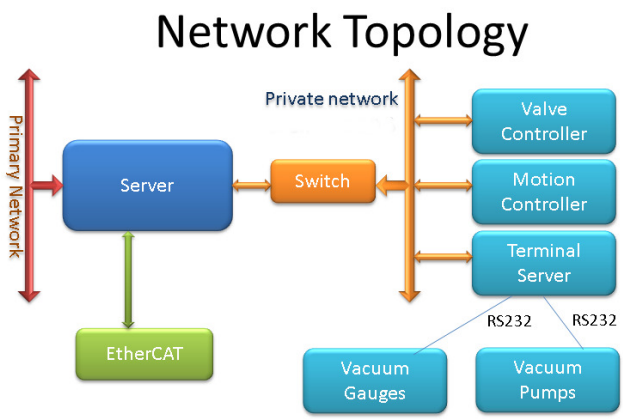

Figure 1: Network topology.

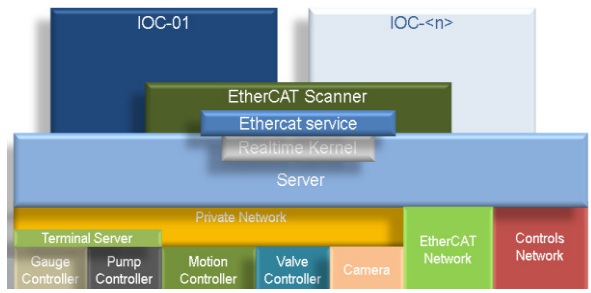

Figure 2: Logical Model.

Instrumentation Network

A number of instrument controllers are now equipped with network interfaces to facilitate control and data acquisition. It is desirable in most instances to secure these on a dedicated network, which is private to the Linux server. This avoids unnecessary traffic on the Primary Controls network and shields the instrumentation from connection outside the scope of the control system dedicated to that area.

EtherCAT Network

Where remote I/O (RIO) is required and not available on a RIO PLC, EtherCAT modules have been employed. The EtherCAT master module Linux may be used seamlessly with either Linux kernels with PREEMPT-RT enabled, or for standard development and "soft" test systems without the real-time extensions. For applications which require greater determinacy, such as high frequency analogue signal sampling, the pre-emptive real-time kernel extensions must be used.

ISBN 978-3-95450-148-9

Tests were performed to measure the effect of jitter on the master process [3]. The kernel patch pre-emptive real-time (PREEMPT_RT) reduces latency and adds the ability to pre-empt most kernel critical sections. However the user-space APIs are unchanged as the necessary calls are already present as part of the POSIX ADVANCED REALTIME standard [4], so that applications can be developed and tested on a standard system. For this application Diamond is using kernel 3.6.11.2-rt33 from MRG.

The following API features are used in the EtherCAT scanner:

- high-precision timers using clock_nanosleep
- SCHED_FIFO pthread scheduling
- PTHREAD_PRIO_INHERIT mutexes

The use of mlockall is mentioned in a previous publication [3] but has since been removed from the code. No adverse effect has been noticed.

Improvements have been realised by:

- Adopting a scheme of using a slave serial number as identifier to allow reconfiguration of the devices.
- Patching the Etherlab library to stop syslog messages overwhelming the system when a section of the bus is disconnected.
- Adding support to read and write Service Data Objects (SDOs) from the bus scanner application and from the higher level control system. The support for SDO came out of the need to support an LED controller, but has wider applications because SDOs are used to configure various EtherCAT Modules.

On a note on the hardware revisions of slaves, recent experience revealed that some PDO names vary between slaves' revisions in the EtherCAT Slave Information (ESI) files from the manufacturer. The scanner configuration files are generated from a description slave models and revision numbers. This information is then used to collect data from ESI files. The EtherCAT driver software propagates the slave information upstream as Asyn parameters. These name changes have prompted the need to revise the management of the slave information files from the manufacturer to screen for name changes before adopting the vendors' ESI files. The impact is limited to a subset of EtherCAT modules and solutions are being discussed to avoid revision specific version of templates. This simplifies hardware swaps in case of failure on an slave module.

The use of EtherCAT for acquisition and control of digital and analogue signals has been in use in beamlines and front ends built in the last four years. The devices have replaced VME signals and solutions have been developed for signal acquisition up to 10 kHz.

The separation of signal acquisition and the scanner publishing the bus signals over a socket, within a Linux server, allows for reusing a signal with different names in different EPICS IOCs. This feature, included in the original design, has proven useful when adding waveform acquisition from an ADC without the need to bring down the running system.

REMOTE I/O PLC

A new generation of Omron PLCs has been selected to enable the adoption of Remote I/O (RIO) and an Ethernet interface between the server and the PLC.

PLC RIO facilitates the reduction of cabling complexity and improving flexibility on deploying I/O into the field. PLC RIO uses standard Ethernet cabling and Profinet, and RIO modules are built up into the standard configurations, one for thermal monitoring and another for general purpose I/O. The Remote I/O does not perform any logic on any of the signals; it simply acts as a receiver and passes everything back to the PLC crate. This necessitates configuration of RIO output modules such that in the event of failure of RIO communication I/O fails to the safe state. The PLC logic is protected by a watchdog timer to protect against failures in communication with RIO.

The new PLC CPU (CJ2M) has a built-in Ethernet connection which allows both EPICS and the programming software to share the connection. EPICS communication is handled over UDP using FINS protocol whilst programming is realised over TCP/IP.

The remote I/O modules were originally prototyped into two standard configurations to enable spares to be maintained and allow faster recovery in the event of a problem. However the reality is that hot swap is not practical without interruption of the RIO communications bus and hence the PLC integrity, as a number of the modules have some intelligence which means that commands have to be sent to the card to configure them.

Problems that Needed Resolving

A significant issue, related to analogue modules, is that the values for temperatures freeze at the last known value when RIO communications is lost. Through the use of a watch-dog timer and detection of communications errors, the analogue is set to an error value.

Benefits of RIO

The biggest benefit of RIO is flexibility of design and the reduction in cabling. This has significantly reduced the number of connections to just those in interface boxes directly mounted out in the field. This has also significantly sanitised the wiring on beamlines. Often the beamline equipment is delivered with different sensors to those expected and we are able to adapt the system for this. The ability to add extra modules into the system as equipment arrives is very useful and suits the piece-meal commissioning of beamline equipment as more complex equipment is delivered.

SUMMARY OF PROGRESS TO DATE

The Diamond Control Systems now implement some 470 Linux based IOCs; a proportion of which utilise distributed I/O, such as EtherCAT. This design pattern has realised an integrated, versatile and maintainable control system, upon which enhancements and upgrades

ISBN 978-3-95450-148-9

can be confidently planned on platforms which provide inherent long term stability and support.

REFERENCES

[1] R. P. Walker, “Commissioning and Status of The Diamond Storage Ring”, APAC 2007, Indore, India.
[2] I. J. Gillingham, “Diamond’s transition from VME to modular distributed I/O”, PCaPAC 2010, Saskatoon, Saskatchewan, Canada.
[3] R. Mercado, I. J. Gillingham, J. H. Rowland and K. Wilkinson "Integrating EtherCAT based IO into EPICS at Diamond." ICALEPCS 2011, Grenoble 2011”
[4] IEEE and The Open Group “The Open Group Base Specification Issue 7” http://pubs.opengroup.org/onlinepubs/9699919799/

ISBN 978-3-95450-148-9