DEVELOPMENTS IN THE INTEGRATION OF VIDEO INTO EPICS AT DIAMOND LIGHT SOURCE

U.K. Pedersen*, M.G. Abbott, M.T. Heron, N.P. Rees,
Diamond Light Source Ltd, Oxfordshire, United Kingdom
M. Rivers, ANL, Argonne, Illinois

Abstract

Firewire (IEEE1394) video cameras, compliant with the IIDC/DCAM specification[1] are used on both the accelerators and photon beamlines on Diamond Light Source (DLS). Initially the integration was through a commercial Firewire/IIDC stack running on VME and VxWorks based EPICS IOCs. Recent developments have migrated the Firewire camera interface to x86 Linux based IOCs using the open source libraries dc1394[2] and EPICS areaDetector[3]. The motivation for this and the software structure is described.

DIAMOND LIGHT SOURCE LTD

Diamond, a third generation 3GeV synchrotron light source[4], commenced operation in January 2007. The storage ring (SR) is based on a 24-cell double bend achromatic lattice of 561m circumference. It uses a full-energy booster synchrotron and a Linac for injection. The current operational state includes twelve photon beamlines and experimental stations, with a further nine beamlines now under design or construction.

DIAGNOSTIC CAMERAS AT DLS

Firewire cameras are in use at DLS for beam diagnostics on both the electron accelerator, and the beamlines and experimental stations. The most commonly used cameras are the Point Grey Flea and Flea2 monochrome models; however a variety of other cameras (mainly AVT and Bassler models) including colour models are in use to meet specific requirements on various beamlines.

As the cameras are installed and used across the entire synchrotron their applications can vary greatly. The main requirements for diagnostics include real-time live calculations of beam size and position. Other applications for image processing have been made by several users around site.

Application on the Accelerator

Monochrome cameras are in use around the accelerators for diagnosing the beam in terms of position, shape and stability. These provide the operators with visual feedback of the beam at various points in the sytem.

* ulrik.pedersen@diamond.ac.uk

Application on Beamlines

The use of diagnostic cameras on beamlines varies across the facility. The most common use is for diagnosing beam position, shape, stability and intensity at key points after major components along the beamline. This is typically used as a diagnostic aid only during experiment setup as it requires fluorescent screens to be moved into the photon beam in order to produce visible light that can be detected by the cameras.

Requests have been made to use some cameras as beam position monitors (BPM) with the intention to automate parts of the beamline alignment procedure. Another use on some beamlines is for sample positioning and monitoring during experiments.

Control System Integration

DLS has chosen the Experimental Physics and Industrial Control System (EPICS)[5] as the base for the control system on both the accelerator and the beamlines. It is a basic requirement that the diagnostic camera integrates well with the rest of the control system. This allows the beam image calculations to feed back to the control system for various control system applications using the EPICS network protocol Channel Access (CA). Integration with the existing control system also saves effort and cost as many libraries and tools can be reused for these tasks.

Client Side Tools

A number of client side tools have been implemented to work with the diagnostic cameras at DLS. They range from simple viewers to complex live calculations on the image data.

The standard control system GUI tool in use across DLS is the EPICS Display Manager (EDM). A video widget can display the raw image data, transferred over the CA network protocol. A number of EDM screens have been designed with the video widget to display the image data and standard EDM widgets to give access to the various control parameters on the cameras (see Fig. 1). These EDM panels offer only control and monitoring features but do not perform any calculations, feedback or automation.

For the cameras in use on the synchrotron machine a number of Matlab routines are used to produce diagnostic information from the live image data.

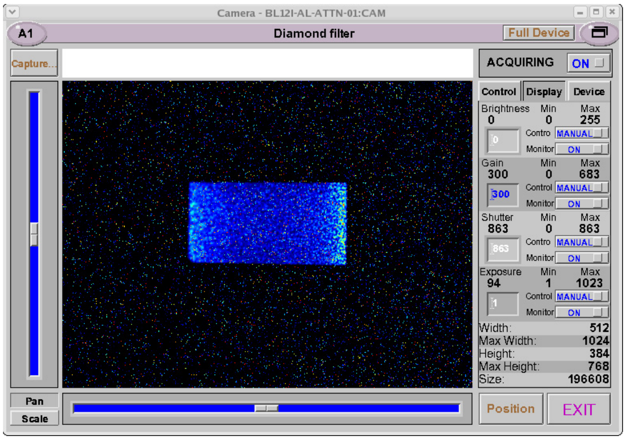

Figure 1: EDM camera control screen on beamline I12.

HARDWARE PLATFORM

Original VME VxWorks Solution

The initial camera solution was chosen to fit in with the main control system platform in use at the time. This used the MVME5500 PPC processor on a VME backplane running the real time OS VxWorks. For firewire support a PMC expansion card was selected which provides one 1394A firewire bus. A proprietary library from the manufacturer of the board was used to integrate control and data readout with the EPICS control system.

This VME based 1394A based solution had a number of limitations, some due to the choice of platform and some due to the specific choice of software implementation:

- Limited to a maximum of 2 cameras running synchronously per controller.
- Many camera parameters fixed by hard coding including a 7.5fps frame rate.
- No practical way to implement colour support.
- Driver library only supplied in binary form, making debugging (without access to source) difficult.
- Limited processor power to implement image processing algorithms.
- Raw image data transferred over the CA network protocol for viewing purposes.
- Each additional CA client multiplies both the image data transferred over network and the controller's CPU load for the network task.

Due to these limitations and issues an alternative solution was investigated which would meet the same requirements and allow re-use of the same cameras and installed equipment.

New Linux x86 Solution

As EPICS now supports the Linux x86 platform this was considered as an alternative to the VME VxWorks based solution for several reasons:

- Open source driver library libdc1394 with a significant maturity and a large and active user and developer forum.
- PCIe 1394B firewire cards available as a mass market product. These provide twice the bandwidth of the 1394A standard.
- Multiple firewire cards can be installed in one server, essentially multiplying the number of cameras that can be used from one server.
- Better integrated development tools as all development takes place on Linux desktops.
- Linux servers were already in heavy use around DLS and in-house expertise readily available.

Further, the Linux solution provides a cash saving as a Linux server and PCIe firewire card are standard products for the mass market. The price for one server with a firewire bus is only around 15% of the price of the existing VME solution with PMC firewire card, and VxWorks firewire library license fee (not including any VxWorks OS license fee).

Switching from Firewire bus 1394A to 1394B and utilising the higher bandwidth, brought out a number of issues with the hardware in cameras and repeaters. These have mostly been addressed now by the manufacturers but have highlighted how volatile the Firewire bus is in practice. The design of the bus topology, cable lengths and the power supplies is very important to ensure the firewire bus operates reliably with multiple cameras.

EPICS SUPPORT

The original VME based solution was implemented as a traditional EPICS Device and Driver support module, interfacing to the proprietary library used for the VME PMC firewire card. Because of the VxWorks specific implementation this could not be re-used for the Linux platform. Furthermore new methods for developing hardware support had become available from the EPICS community.

As new camera software support was required a number of additional features were requested. However, backwards compatibility with the existing client side tools had to be maintained.

Asynchronous Driver Support

One of the libraries to gain increasing support within the EPICS community is known as ASYN[6]. The ASYN module provides a framework to write asynchronous drivers.

In traditional, basic EPICS device support read or write requests have to be synchronous (blocking) and thus any hardware I/O operation must return quickly (in less than a few micro seconds). This approach lends itself very well to drivers for fast, register-based devices but not so well for slower bus devices. Asynchronous support for slower (typically bus-interfaced) devices can be implemented in traditional EPICS device and driver support but requires a multithreaded implementation of the driver with a message queue to pass read/write requests to hardware. The ASYN module implements such a queueing facility in a multithreaded library and provides the user with a simple framework in which to implement the hardware IO operations. Using the ASYN module for writing asynchronous driver support significantly decreases the amount of code to to be written.

EPICS ASYN support for the Linux firewire camera library libdc1394 has been implemented at DLS. This implementation has addressed several of the issues from the original VME VxWorks solution—in particular the issue that only 2 can acquire data simultaneously. With the new implementation the number of simulataneously capturing cameras is only limited by the number of DMA channels available on the firewire interface hardware (typically 4 or 8 channels per firewire interface chip). A new image processing implementation now does a better job at finding the beam spot in the image and calculates the beam size and position.

Area Detector Support

Recent development from the EPICS community has produced the areaDetector[3] library which provides general purpose EPICS support for 2D detectors. This library is based on the ASYN support module and implements a plug-in structure that separates the hardware specific driver implementation from the processing and control parameters that 2D detectors generally share.

Processing modules (plug-ins) can be plugged into the output of a driver in a producer-consumer type design pattern. One plug-in can run beam diagnostics to find and output the beam position, size, profile and so on. Another plug-in can compress the image data into a media stream that can be viewed with a normal media/stream player. The limitation of processing plug-ins only depends on to the limitations of the platform on which it runs in terms of CPU cycles and available memory. This architecture lends itself very well to the many different requests for image processing routines that have been requested by users around DLS. As individual plug-ins can be dynamically enabled at run time, the same application can be built for all camera controllers across site as different users can enable only the processing they need for their specific situation.

An areaDetector driver plug-in firewireDCAM[7] has been developed at DLS along with mjpgServer[8]—a compression/streaming plug-in that converts the raw image data to a compressed media stream that can be viewed across network using various standard media player applications.

FUTURE DEVELOPMENT

The firewire areaDetector solution is still being tested in the lab and will be installed on beamlines when testing has completed with satisfactory results. Development is still ongoing to ensure backwards compatibility with the existing client side tools in use at DLS.

The image processing algorithms requested by various users will now be implemented as areaDetector plug-ins. The existing matlab routines will be translated to C/C++ plug-ins. As a future development, external image processing libraries may be integrated as plug-ins for specific requirements.

CONCLUSION

The switching of hardware platforms from the VME based VxWorks to x86 Linux has been successful. The wider range of available products provides greater flexibility in the system design. More powerful CPUs and a larger amount of memory on these servers improves the performance of the image processing and allows multiple cameras to run simultaneously. Cost savings have also been significant due to the use of commodity products.

The new areaDetector development and its consumer-producer design pattern works very well and has cut down the amount of code necessary to implement the driver support. The plug-in architecture makes it possible to design and implement additional image processing features, separate from the existing driver support and reusable for other camera drivers.

REFERENCES

[1] 1394 Trade Association, “TA Document 2003017 IIDC 1394-based Digital Camera Specification Ver.1.31”, February 12 2004 `http://damien.douxchamps.net/ieee1394/libdc1394/iidc/`

[2] D. Douxchamps, libdc1394, `http://damien.douxchamps.net/ieee1394/libdc1394` and `http://sourceforge.net/projects/libdc1394`

[3] M. Rivers, Area Detector, `http://cars9.uchicago.edu/software/epics/areaDetector.html`.

[4] R. P. Walker, “Overview of the Status of the Diamond Project”, EPAC’06, Edinburgh, June 2006.

[5] Experimental Physics and Industrial Control System EPICS, `http://www.aps.anl.gov/epics`.

[6] EPICS Asynchronous Driver Support ASYN, `http://www.aps.anl.gov/epics/modules/soft/asyn`.

[7] U.K. Pedersen, “FirewireDCAM EPICS Support Module”, `http://controls.diamond.ac.uk/downloads`.

[8] T. Cobb, “mjpgServer EPICS Support Module”, `http://controls.diamond.ac.uk/downloads`.