DOOCS CAMERA SYSTEM

G. Grygiel, V. Rybnikov, DESY, 22607 Hamburg, Germany

Abstract

The Free Electron Laser in Hamburg (FLASH) [1] with its complex accelerator diagnostics and user experiments requires for both, the operation and the experiments, a lot of different cameras. A common interface for simple USB cameras, for fire wire cameras and for high resolution cameras with e.g. multiple "region of interest" was developed. This system integrates the various camera types in a transparent way into the FLASH control system DOOCS [2]. In addition the cameras are connected to a fast data acquisition system (DAQ) [3]. The DAQ provides the synchronization with other diagnostics data, online processing of the images and a long time archiving.

INTRODUCTION

The main design idea was to develop a common approach for different camera types and frame grabbers used in FLASH. A camera base class implements all common functions in C++. This implies a common set of properties available to the control system.

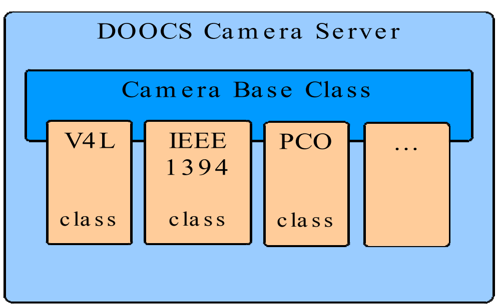

Figure 1: Camera Server Implementation.

DOOCS is a distributed control system that was developed for HERA [4] and FLASH applications. It is an object oriented system design from the device server level up to the operator console.

Any camera type or frame grabber has its own camera specific class which inherits from the base camera class. All camera common parameters (e.g. exposure, brightness, etc.) belong to the base class and therefore have the same DOOCS property name. The camera specific class for a particular camera type provides its additional features that do not exist in the base class. The DOOCS camera server makes use of camera specific classes to provide the access to a corresponding camera type. It's possible to combine the different camera types in one server. The camera servers are applicable for all kind of operations like changing camera settings, grabbing images etc. Some simple operations on images (histogram calculation, background subtraction, calculations on regions of interest, etc.) are also implemented.

CAMERA SERVER

Timing

Camera server computers are equipped with the FLASH/FNAL IP timing modules [5]. They provide eight channels of digital pulse outputs with programmable delays synchronized with the central timing system of FLASH. Every camera supporting external trigger mode is triggered by one of the timing pulses. The timing system distributes a programmable event number incremented with every linac shot. Images from triggered cameras include an event number as well as a time stamp required for synchronisation with other linac diagnostics data. All cameras can be used in a free-run mode also. This mode is mainly used for observation tasks without being synchronized to the shots of FLASH.

Controls

Every camera parameter can be controlled by a DOOCS property. The initial camera settings are stored in a camera server configuration file. This file is updated when configuration parameters are changed. Therefore parameters are preserved after a server restart. Some properties are read-only for presenting the information from cameras read out during the server start-up e.g. serial numbers, maximum resolution, etc. All cameras of the known types are detected during the start-up of the server process. There is a configuration flag per camera that controls the camera assignment. The flag defines either a particular camera should be taken into operation or any camera of this type.

Images

The camera server provides the access to the last grabbed image via a DOOCS property of type IMAGE. The IMAGE contains a header and a byte array as the image itself. The header includes an event number, a time stamp, and image parameters such as resolution, binning, image format, rotation, etc. The property IMAGE can be used by the client applications to read out the camera frames. To be able to process all frames (feedback or offline analysis) the FLASH DAQ system is used. There is also an option to use the local disk for image storage. Both, black and white and coloured images are supported.

Common Features

All camera independent features are implemented in the common camera base class. They are:

- Calibrations, background subtraction, offsets
- Image pixel histogram calculation
- X & Y Spectra with calculations of
  - min, max, mean
  - sigma, centre of mass

  All values can be calculated in physical units.
- Software region of interest operations
  - three types of shape (ellipse, rectangle or cross)
  - rotated shapes
  - different visualisation modes
  - similar calculation as for X & Y spectra
- Software image flipping X/Y
- JPEG Images, the user can switch from raw-data format to jpeg representation (to save network bandwidth).
- Central storage of all images via FLASH DAQ

Camera Specific Features

- Hardware region of Interest
- Hardware binning
- Loops, summing up images from sequential triggers
- Multiple, sequential images from one trigger

INTEGRATION INTO THE DAQ

For FLASH a fast data acquisition system to support the accelerator operation and the user experiments was developed. The system collects data from hundreds of ADC channels (in the range up to 2G samples per second) and digital video cameras. Software regions of interest calculated by the camera servers can be also stored in the DAQ if required. All the data is collected (in sync with the timing system) in a shared memory of a central multi-processor computer. The data flow from the camera servers to the DAQ is shown in figure 2.

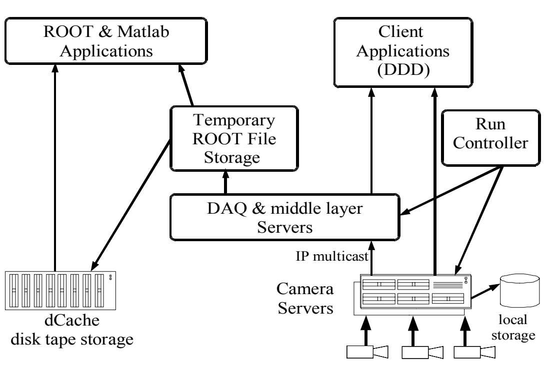

Figure 2: Data flow of images from cameras to storage and users.

The camera servers send images to the DAQ via IP multicast packets. One image is usually built of several packets. The packets belonging to one image including the header information are collected by the DAQ server. The full image information is made available to the middle layer servers by means of a shared memory facility. Synchronized with other data relevant to the same linac shot the camera data is stored in ROOT [6] files and in dCache [7]. For off line analysis and data browsing ROOT and Matlab [8] tools are available.

The DAQ configuration parameters are set in the camera servers as well as in all DAQ relevant processes by the Run Controller.

The camera server has an interface for writing images to the local storage. One can overwrite the same image or can produce a series of images with the time stamp.

IMPLEMENTED CAMERA SYSTEMS

The camera servers are in operation for the following diagnostics and measurement experiments at FLASH: Gun laser beam line, various electro optical beam diagnostics [9], transverse deflection bunch length measurements and other bunch diagnostics, beam position measurements, infrared spectra, beam energy measurements, etc. They are also used for the beam line diagnostics and Free Electron Laser (FEL) user experiments [10] [11]. There are two groups of cameras in operation.

Based on Standards

- v4l (Video for Linux) [12] compliant hardware (digital and analogue cameras, TV and video cards)
- IEEE1394 [13] (fire wire) based cameras
  - Basler
  - Orange Webcams
  - ...
- Gigabit Ethernet cameras [14]
  - Prosilica [15]

Special Cameras

- Andor DH740 [16]
- PCO dicam [17]
- uEye USB cameras [18]

OPERATING EXPERIENCE

The first camera type was adopted in January 2006. Currently 6 different types of cameras and in total 60 cameras are in operation. The cameras are served by 16 camera servers running under Debian GNU/Linux [19] OS. Cameras and computers are sensible to radiation damage. Therefore the hardware is placed as far as possible away from the radiation source and is shielded with lead or concrete. The maximum cable length currently exploited for USB cameras with use of hubs is 17 meters, and for IEEE1394 cameras - 20 meters. We are forced to use remote controllable switches/hubs for the camera connections to be able to reset cameras by power cycling. To solve the problems in case of radiation damage a script was developed to perform the following procedure: stop camera server, switch off cameras, reload drivers, switch on cameras and start camera server. This is a normal procedure during the linac operation for cameras near radiation sources.

On-line view, controlling and commissioning of all camera properties is done via DDD [20] panels included in the standard FLASH operator GUIs.

REFERENCES

[1] Free Electron Laser in Hamburg http://flash.desy.de/

[2] K. Rehlich et al, DOOCS: an Object Oriented Control System as the integrating part for the TTF Linac, Proceedings ICALEPCS '97, Beijing, China. Available: doocs.desy.de

[3] K. Rehlich et al, DAQ: Multi-Processor based Fast Data Acquisition for a Free Electron Laser and Experiments, Proceedings of 15th IEEE NPSS Real Time Conference 2007, Fermilab, Batavia IL, USA

[4] HERA http://hera.desy.de

[5] FLASH/FNAL IP timing http://tesla.desy.de/doocs/hardware/8ch_10MHz_ADC_timing.pdf

[6] R. Brun and F. Rademakers, ROOT - An Object Oriented Data Analysis Framework, Proceedings of the AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. Available: root.cern.ch

[7] P. Fuhrmann et al, dCache, a distributed Storage data caching system, Proceedings of CHEP 2001, Beijing, China., Available: dcache.desy.de

[8] The Math Works http://www.mathworks.com/

[9] G. Berden et al, Single Shot Longitudinal Bunch Profile Measurements at FLASH using Electro-optic Techniques, Proc. of the European Particle Accelerator Conference, EPAC'06, Edinburgh, UK, 26-30 June 2006

[10] Michael Wellhöfer, The Monochromator Beamline at the Free-Electron Laser Hamburg (FLASH), Dissertation, Universität Hamburg 2007

[11] Armin Azima et all., Jitter measurement by spatial electro-optical sampling at FLASH free electron laser, Proceedings of EPAC 2006, Edinburgh

[12] Video for Linux http://www.linuxtv.org/

[13] IEEE 1394 for Linux http://www.linux1394.org/

[14] GiGE Vision http://www.machinevisiononline.org/

[15] Prosilica GE-Series http://www.prosilica.com

[16] ANDOR iStar DH740 data sheet, http://www.andor.com, 2005

[17] PCO dicam http://www.pco.de

[18] uEye http://www.ids-imaging.de/

[19] Debian GNU/Linux http://www.debian.org/

[20] The DOOCS Data Display http://tesla.desy.de/doocs/ddd/ddd_intro.html