

## SPring-8/SACLA スクリーンモニタ更新に向けた GigE カメラ制御システムの開発

### DEVELOPMENT OF GigE VISION CAMERA CONTROL FOR UPGRADING SCREEN MONITOR OF SPring-8/SACLA

清道明男<sup>#, A)</sup>, 福井達<sup>B)</sup>

Akio Kiyomichi<sup>#, A)</sup>, Toru Fukui<sup>B)</sup>

<sup>A)</sup> Japan Synchrotron Radiation Research Institute (JASRI)

<sup>B)</sup> RIKEN SPring-8 Center

#### Abstract

In a part of the SPring-8 upgrade project, the SACLA linac will be used as the injector for the SPring-8 storage ring. We will upgrade the beam monitor system for beam transport, which consists of screen monitor (SCM), beam position monitor (BPM) and current monitor (CT). For the SCM, we adopted GigE Vision standard for the CCD camera. We have developed camera control software using open source libraries to integrate various vendors' GigE Vision cameras with the SPring-8 control framework. And we build a system to control SCM with one unit by installing PCI Express cards such as PoE type Ethernet and trigger counter. In this presentation, we will report on the functions and implementation of the Aravis library when integrating the GigE camera into the SPring-8 control framework.

#### 1. はじめに

SPring-8 のアップグレード計画では、SACLA Linac を SPring-8 ストレージリングの入射器としての使用を予定している。SACLA からのビーム輸送系は約 600 m で、最大繰り返し 10 Hz の入射を行う。そこにスクリーンモニタ (SCM) 約 30 台、BPM 約 30 台、CT 約 10 台の設置といったビーム輸送系モニタシステムのアップグレードを行う。

現在、SPring-8 加速器においてスクリーンモニタを始めとした画像診断系の多くは CameraLink 規格の CCD カメラを採用している。CameraLink I/F に対応したカメラ制御計算機は、PC サーバや MicroTCA をベースとした画像処理システム[1-3]を開発して運用している。

CameraLink I/F は高帯域幅のデータ転送能力をもち、トリガ同期ができ、シリアル通信によるカメラ制御も可能である。複数ベンダーからグラバボードが発売されており Linux に対応した API も利用可能であったことから、SCM のアナログカメラからデジタルカメラへの移行に際して採用された。しかし、伝送距離は 10 m と加速器施設で利用するには短く、光変換による送受信装置が必要となることから配線が複雑で高コストになる点がデメリットである。そこで、ギガビットイーサネットを通じて長距離伝送が可能となる GigE Vision 規格の CCD カメラの導入を検討した。

SPring-8 の一部の装置では GigE カメラの導入実績があるが、当時は Windows 用のソフトウェアでしか使用できなかったり、Linux 用の開発環境を提供するのも一部のベンダーのみであったりといった事情から、専用のソフトウェアとして開発された。そ

のため、SPring-8 制御系の中でも個別対応となり、カメラの選択筋も狭まり、長期の保守性が劣るという問題点があった。近年、汎用的な GigE Vision 用のオープンソースライブラリが利用可能となったことより、これを用いてさまざまなベンダーの GigE カメラに対応するカメラ制御ソフトウェアを開発することとした。そして SACLA からのビーム輸送系の SCM 更新において GigE カメラの導入を目指す。現在、SACLA で使用している JAI 製 CameraLink カメラの GigE 版を採用予定である。(Fig. 1)



Figure 1: JAI GO-2400M.

#### 2. 開発方針

##### 2.1 開発環境

GigE カメラをエンドユーザが利用するにあたり、多くのベンダーは何らかの形式の開発環境 (SDK) を提供している。カメラ制御システムを開発するにあたり、これらの SDK に依存することは幾つかの間

<sup>#</sup>kiyomichi@spring8.or.jp

題が生じる。

- 殆どのベンダーの SDK は自社カメラと結びついているため、他社のカメラを利用することが出来ない。
- SPring-8 の制御システムは UNIX ベースであるのに対し、カメラベンダーが提供する SDK の多くは Windows を対象としている。
- 多くのベンダー提供 SDK はクローズドソースであるため、問題が生じた際のデバッグが困難となる。

実際、SPring-8 においても GigE カメラの導入実績はあるが、多くは Windows でしか制御が出来ないカメラ（ストリークカメラなど）である。Linux 用の SDK が提供されており制御系に組み込むことができたカメラもあるが専用ソフトとなる。そのため更新に制約が生じるなど長期の保守には向かない事例が多い。そこで、様々なベンダーの GigE カメラに対し SPring-8 制御フレームワークへ組み込むための方策を検討した。

## 2.2 SPring-8 制御系への導入方針

SPring-8 の制御フレームワークは制御命令を行う GUI 部のレイヤーとハードウェアを制御する機器制御レイヤーに分かれており、Message Server を介してメッセージを送る。機器制御レイヤーでは Equipment Manager (EM) と呼ばれるソフトウェアが動作している。

現在、SPring-8 の画像診断系は CameraLink 規格の CCD カメラが主流である。カメラ制御の EM を整備して様々なベンダーのカメラに対応している。スクリーンモニターでは、EM はカメラ制御と画像保存までを担い、ビームプロファイル測定などの画像解析は GUI が行う。

GigE カメラ制御システムは、基本的なカメラ操作を EM の汎用関数として整備し、画像解析は既存の GUI を用いる。EM 関数は EM 構成ファイル内に関数とカメラの IP アドレスまたはホスト名を指定するだけで、新たなコード開発をすることなく制御できることを方針とした。そして CameraLink カメラに置き換え可能なシステムを実現することを目標とした。

## 3. GigE カメラ制御

### 3.1 GigE Vision 規格と Aravis ライブラリ

殆どの GigE カメラは GigE Vision 規格[4]に準拠している。そしてこれは GenICam (Generic Interface for Camera)規格[5]を使用することでカメラの機能の共通化が行える。GenICam 規格は、さまざまなベンダー間や物理接続にわたって、多種類のカメラに共通のインターフェースを提供している。カメラはそれがサポートする機能、およびそれらがデバイス上のレジスタにどのようにマッピングされるかを記述する XML ファイルを提供する。XML ファイルにはカメラの制御方法が記述されているため、さまざまなベンダーのカメラをサポートすることが可能となる。

Aravis[6]は GigE Vision を含む GenICam 規格に準拠し、glib/gobject ベースのオープンソースライブラ

リである。現在の最新バージョンは 0.6 で、GigE Vision と USB3 プロトコルのカメラをサポートしている。我々は Aravis ライブラリを導入するにあたり、OS に Ubuntu16 を選択した。カメラの設定は露光時間、ゲイン、ROI、ビニングといった基本的な操作が可能である。また、簡単なカメラシミュレータと画像ビューアも提供している。

Figure 2 にカメラとの接続試験の様子を示す。JAI 社 GO-2400M、Basler 社の acA780-75、Leutron Vision 社 PicSight P202M といった複数のベンダー製カメラにて、カメラ操作や画像取得ができた。一方、カメラの独自機能には対応していないことも確認した。現行バージョンでは例えば設定値の保存、IP アドレスの設定、カメラのテストパターン出力といった機能は使用できない。これらは、カメラの設置時の初期設定には必要な機能ではあるものの、ベンダー提供のソフトウェアで対応できる。そのため設置後の運転には問題ない。



Figure 2: Initial test using aravis.

### 3.2 カメラ制御 EM 関数

GigE カメラ制御用 EM を整備した。Aravis ライブラリでは露光時間、ゲイン、ROI、ビニングといった基本的なパラメータの設定および取得の関数が用意されている。これを利用して、各パラメータの設定 (put) および取得 (get) の EM 関数を用意した。外部トリガ設定はトリガソースを指定することで設定可能だが、カメラの種類によってトリガソース名が違うため手順が複雑となる。そこで、カメラからトリガソース名を取得する関数と組み合わせることで 1 つの EM 関数とした。

画像データは NFS ディスク上に 1 枚毎に書き出す。データフォーマットは HDF5 (Hierarchical Data Format release 5)[7] を採用した。HDF5 はデータをグループ毎に分類し階層的に扱うことができるので、画像の 2 次元配列データとカメラ名や画像サイズ・ゲイン・露光時間・トリガ番号・タイムスタンプといったメタ情報を 1 つの HDF5 ファイルに保存できる。本件の画像データファイルは SACL A スクリーンモニタの画像処理システム[2]と同じデータ構造とした。

Figure 3にEM構成ファイルの例を示す。引数部分にカメラのIPアドレスまたはホスト名を指定することで、カメラパラメータの設定・取得が可能となる。

```

put/sr_mon_gigecam_param
trigmode_id    em_cntl_gigecam_put_triggerrmode    "192.168.10.1"
               none
               em_std_ret
exposure_id    em_cntl_gigecam_put_exposuretime    "192.168.10.1"
               none
               em_std_ret
gain_id        em_cntl_gigecam_put_gain            test-camera-10
               none
               em_std_ret

get/sr_mon_gigecam_param
exposure       em_cntl_gigecam_get_exposure       "192.168.10.1"
               none
               em_cntl_gigecam_ret_int
gain           em_cntl_gigecam_get_gain            test-camera-10
               none
               em_cntl_gigecam_ret_int

get/sr_mon_gigecam_grab
image          em_cntl_gigecam_get_image_onefile   test-camera-10 [dir] [name]
               none
               em_cntl_gigecam_ret_int
               ↑
               HDF5 file
    
```

Figure 3: An example of config.tbl.

### 3.3 画像データ収集 MDAQ

加速器ビーム診断系ではスクリーンモニタの撮像をビームに同期して行う。ビームトリガに同期した画像データ収集はSpring-8 制御フレームワークの新しいDAQシステムのMDAQ[8]で実現した。

スクリーンモニタの画像データ収集は、

- 1) カメラ選択
- 2) run type を指定する
- 3) 録画枚数を指定する
- 4) 録画開始命令を送る

という手順で実行する。カメラ選択、run type、録画枚数、録画開始といった録画に関わる制御パラメータはShard Memoryを用いてEMとMDAQで共有する。MDAQはトリガに同期して撮像プロセスを実行し、GUIからEMを介して録画の制御指令が送られると、指定した録画枚数分のHDF5ファイルを保存する。Figure 4に録画したファイルのディレクトリ構造を示す。

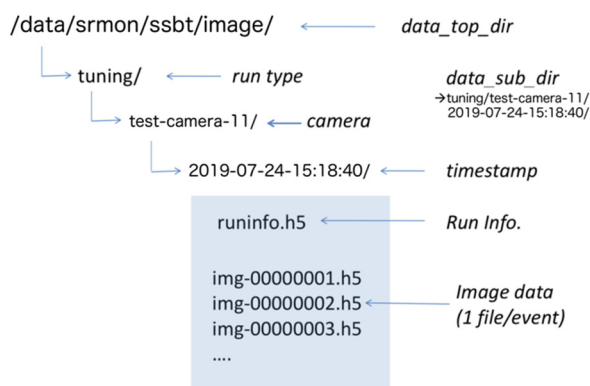


Figure 4: Directory structure of image file.

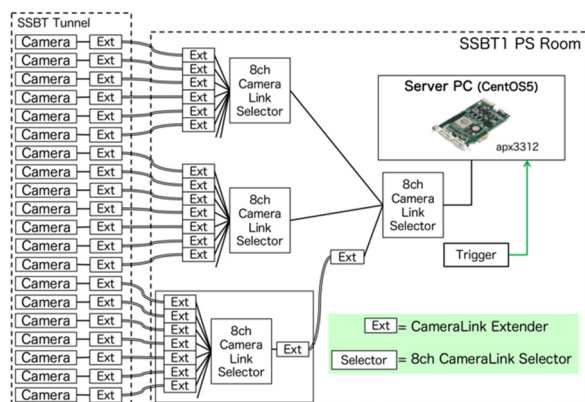
## 4. スクリーンモニタへの適用

### 4.1 現行の問題点と新システム構成

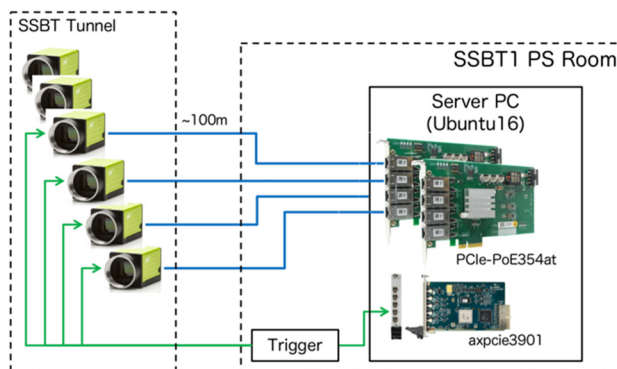
スクリーンモニタ (SCM) は破壊型の測定器であるため同時に使用できるモニタは一つである。よってカメラは同時に1台接続できればよい。現在のビーム輸送系SSBTのSCMはCameraLink延長器とCameraLink切替器を組み合わせた構成で運用している[3]。Figure 5(a)に現行のカメラシステムの構成を示す。このシステムは長期に亘って安定に動作しているが、以下のような課題が分かってきた。

- 光変換による延長器が高コストとなる
- カメラと計算機間の経路が多いため、トラブル時の問題箇所特定に時間がかかる。
- CameraLinkケーブルは太く、配線が複雑なため交換作業に時間がかかる。

長期の運用では、放射線損傷によってCCDカメラやトンネル内の延長器は故障するため定期的な交換が必要となる。GigEカメラによるスクリーンモニタを導入するにあたり、できるだけシンプルにして故障時の交換作業なども容易となる構成とした。Figure 5(b)に新システムの構成案を示す。GigEカメラ制御用計算機はPoE給電対応Ethernetボード



(a) current CameraLink camera connection



(b) GigE camera connection

Figure 5: Schematic view of camera connection.

(Neusys tech. PCIe-PoE354at) とトリガ入力用のカウンターボード (Gopher axpcie3901) を実装する。ギガビットイーサネットは最大 100 m の伝送距離を持つので GigE カメラは計算機と直結してローカルネットワークを組み、PoE 給電で運転する。

#### 4.2 デバイスセットアップ

Neusys tech. の PCIe-PoE354at[9] は PCI Express x4, Gen2 の 4 ポート Ethernet ボードで、PoE(Power-over-Ethernet)給電制御に対応する (Fig. 6)。ベンダーより Linux 用ドライバおよびライブラリ (libwdt\_dio.so) を提供してもらった。PoE 給電制御は PCI Express のスロットから SMBus を使用して信号を送っている。そのため計算機のマザーボードが対応している必要がある。我々が採用した Supermicro 社のサーバ計算機では「SMBus to PCI Slots」のジャンパーで有効に設定することでポート毎の PoE 給電 ON/OFF 制御が可能となった。

Gopher の axpcie3901 は 4ch の 32bit カウンタを備えた高速カウンタ CompactPCI ボード axpcie3901 [10] の PCI Express 版である (Fig. 6)。入力信号レベルは TTL または NIM で周波数 200 MHz までのパルス入力を計測可能である。これまでは CentOS5 - 32bit OS での利用実績があったが、このドライバを 64bit OS 対応の改造を行い Ubuntu16 で利用可能とした。

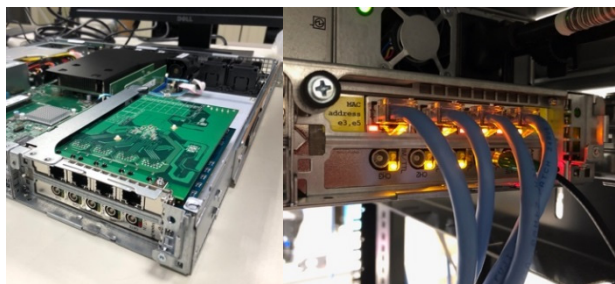


Figure 6: PCIe-PoE354at (up) and axpcie3901 (down).

#### 4.3 ソフトウェア整備

GigE カメラ制御 EM 関数に加えて、PCIe-PoE354at ボードの PoE 給電制御の EM 関数を開発した。そしてトリガ同期で動作する MDAQ を axpcie3901 ボードのトリガカウンタに対応させて、スクリーンモニタに対応した GigE カメラ制御 EM および MDAQ を整備した。

### 5. まとめと今後の予定

GenICam 規格用のオープンソースライブラリである Aravis を使用して、GigE Vision 規格に対応した汎用カメラ制御 EM を開発し SPring-8 の制御フレームワークに組み込んだ。SPring-8 において広く利用している CameraLink 規格のカメラに追加して選択の幅が広がることを期待される。

取得した画像ファイルはファイルサーバに保存するが、撮像時のカメラの設定やイベント番号などの

メタ情報は DB に保存することを検討している。画像と同様にデータサイズの大きい波形データ収集においてもファイルサーバにデータ、DB にメタ情報を保存する仕組みを構築しているの、これに合わせたフォーマットにする。

今夏に SSBT ダンプラインの SCM のうち 2ヶ所に GigE カメラを導入する予定である。EM に本制御システムを適用し GUI の移植を行うなどの運用を進めながら、今後の SACLA からのビーム輸送系での SCM 更新に適用していく。

#### 参考文献

- [1] A.Kiyomichi *et al.*, “Improvement of MicroTCA-based Image Processing System at the SPring-8”, Proceedings of the 12th Annual Meeting of Particle Accelerator Society of Japan, Tsuruga, Japan, Aug. 2015, pp.1303.
- [2] T.Matsumoto *et al.*, “Commissioning of Beam Profile Monitoring DAQ System for XFEL/SPring-8 SACLA”, Proceedings of the 8th Annual Meeting of Particle Accelerator Society of Japan, Tsukuba, Japan, Aug. 2011, pp.554.
- [3] T.Honiden *et al.*, “Construction of the Beam Profile Monitor System with CameraLink in the SPring-8 Booster Ring and Beam Transport Line”, Proceedings of the 10th Annual Meeting of Particle Accelerator Society of Japan, Nagoya, Japan, Aug. 2013, pp.1090.
- [4] GigE Vision – True Plug and Play Connectivity; <http://www.visiononline.org/gigevision>
- [5] GenICam – The Generic Interface for Cameras standard; <http://www.emva.org/standards-technology/genicam>
- [6] Aravis – A vision library for genicam based cameras; <http://www.github.com/AravisProject/aravis>
- [7] HDF5; <https://www.hdfgroup.org/solutions/hdf5/>
- [8] T. Fukui *et al.*, “Status of the control system for the SACLA/SPring-8 Accelerator complex”, Proc. of ICALEPCS2017, Barcelona, Spain, October 2017, pp.1995, FRAPL03
- [9] <https://www.neusys-tech.com/en/product/feature/poe-poe/pcie-poe354at-352at>
- [10] <http://gopher.co.jp/products/axpcie3901.html>