

電子光物理学研究センター加速器制御系の現状

STATUS OF ACCELERATOR CONTROL SYSTEM IN RESEARCH CENTER FOR ELECTRON PHOTON SCIENCE, TOHOKU UNIVERSITY

長澤育郎[#], 南部健一, 柏木茂, 日出富士雄, 武藤俊哉, 高橋健, 鹿又健, 濱広幸

Ikuro Nagasawa[#], Kenichi Nanbu, Shigeru Kashiwagi, Fujio Hinode, Toshiya Muto, Ken Takahashi, Ken Kanomata, Hiroyuki Hama

Research Center for Electron Photon Science, Tohoku University

Abstract

Research Center for Electron Photon Science, Tohoku University has 4 accelerators which are a 1.3GeV Booster-Storage ring (BST-ring), an injector linac for BST-ring, a high-intensity linac for RI production and a test accelerator. It is difficult to develop newly accelerator control system for our facility because the human resources are limited. Therefore, we have developed accelerator control system using LabVIEW. This paper reports current status and future of the control system.

1. LabVIEW の導入経緯

電子光物理学研究センターには 1.3GeV Booster-Storage ring (BST-ring)が設置され共同利用に供されている。少人数で加速器を維持・運用することを考慮して BST-ring とその入射器の制御系プログラムはほぼ全てが LabVIEW[1]で構築されている。

LabVIEW とは、C 言語や Python のようなテキストファイルでソースコードを作成するテキスト型言語とは異なり、グラフィカルプログラミング言語と呼ばれ GUI に対してマウス操作で各種コンポーネントを配置・配線することでプログラムを作成する言語である。より具体的には他言語において関数に対応する VI(Virtual Instrument)、変数に対応する制御器・表示器をブロックダイアグラムと呼ばれる領域へ Figure 1 のように配置・配線することでプログラムを作成する。



Figure 1: LabVIEW Block Diagram. (The left blue icon is a control, the center icon is a VI and the right icon is an indicator. In LabVIEW, users make programs by placing and wiring them in a region named Block Diagram.)

LabVIEW はデフォルトで高機能なライブラリが付属しており、それら全てが関数パレットと呼ばれる GUI からアクセス可能となっている(Figure 2)。このパレットには機能別に VI が分類、配置されており、どのような VI が存在するのか覚えずとも推測によって迅速に求める機能を探すことが可能である。他の

[#] nagasawa@lns.tohoku.ac.jp

テキスト型言語では高機能なライブラリを利用するには、ライブラリ自体を探し、選定し、利用方法を調べて、利用環境を整えるという作業が必要となるため、これらの作業を劇的に軽減する LabVIEW は開発効率の面では非常に優れているものと考えられる。

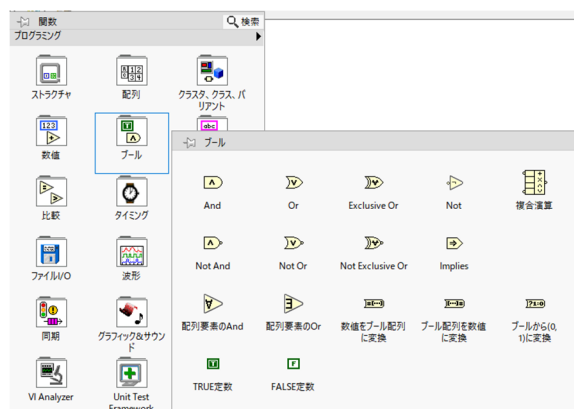


Figure 2: LabVIEW Functions Palette. The Functions palette contains functions to make programs. It is broken into various categories so it is easy to find a VI that suits each application.

また、基本的には配線で、データの入出力を行っているためテキスト型言語と比較してデータの流れやつながりが把握しやすく、プログラム実行時に配線をクリックするだけで配線を通るデータを表示でき、データのグラフ化もグラフ型表示器を配線に接続するだけで実現可能であり、その他デバッグ用の機能も直観的に利用可能となっている。テキスト型言語と比較してデバッグ時の環境整備や設定が非常に容易に行えるために、デバッグ環境も他言語と比較して優れていると考えられる。

以上より、LabVIEW によるプログラムの開発速度は他のテキスト型言語に比べて非常に速いと考

れ、またメンテナンスやエラー発生時の対応など管理の面からも非常に優れており、少人数で制御系を運用するためには非常に有用なソフトウェア開発環境であると考えられる。

電子光物理学研究センターの加速器を維持運用する職員は、加速器の運転とそのメンテナンスや、ユーティリティの保守管理など様々な業務を担当しているため、加速器制御系の構築と運用に専念することが出来ない。そこで制御系の構築と運用を可能な限り省力化することが求められ LabVIEW の導入という結論に至った。

2. EPICS CA と CA Lab の導入経緯

制御系の拡張性を確保するため、分散制御システムとして制御系を構築することになった。分散制御システムを構築する上で、通信プロトコルやデータフォーマットの開発者ごとの乱立を防ぎ、管理や開発の工数を削減する必要があったため、これらの課題を解決する分散制御システムである Experimental Physics and Industrial Control System (EPICS)[2]を導入することにした。

導入の決め手として、多くの加速器施設で運用実績があること、その通信プロトコルである EPICS Channel Access(CA)を LabVIEW 上で簡便に利用できる優れた VI ライブラリである CA Lab[3]が存在したことである。

ただし、あくまでも CA Lab は LabVIEW 上から EPICS CA で通信するための VI ライブラリであり、EPICS Input/Output Controller(IOC)開発に適した VI ライブラリが存在しないことは問題であった。ここで、EPICS IOC とはネットワーク上 Process Variable(PV)と呼ばれるデータを提供するプログラムで、PV の値は EPICS CA で変更することが可能であり、PV の変化に応じて EPICS IOC は担当する装置を制御する仕組みとなっている。

LabVIEW には EPICS I/O Server[4]と呼ばれる EPICS IOC を構築するための機能も存在するが PV の変化をプログラム上でポーリングすることでしか検出できなかったため、この導入は避けた。代わりに SoftIOC と呼ばれるネットワーク上に PV を提供するだけで特定の装置を制御する機能を持たないプログラムと、PV の値が変化した際に LabVIEW 上でイベント通知する CA Lab の VI を導入し、各 PV にアクセス制限を行って運用することとした(Figure 3)。これにより、プログラム上でのポーリングを避けイベント駆動型の実装が可能となった。通常の EPICS IOC は自身が所持する PV に対し直接アクセスしているのに対し、SoftIOC と CA Lab の構成では EPICS CA 経由で PV にアクセスしているため、その分オーバーヘッドによる負荷が大きくなってしまいう問題がある。しかし、今後 EPICS IOC を LabVIEW での構築に適した VI ライブラリが実装された際、ライブラ

リ導入時のプログラム修正の工数の削減を考慮して、上記の構成で EPICS IOC を構築することとした。

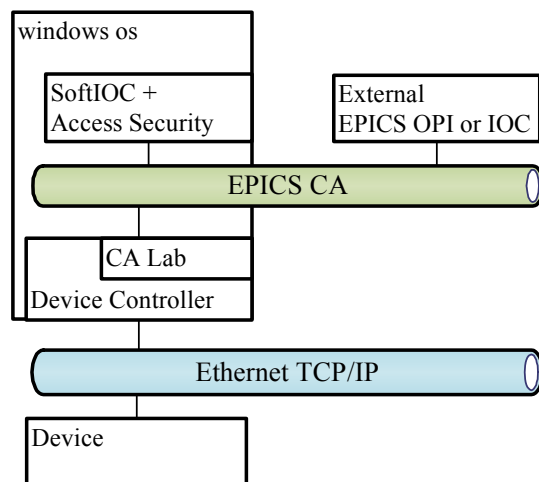


Figure 3: EPICS IOC configuration pattern in the BST-ring control system. In the BST-ring control system EPICS IOC is configured with SoftIOC and CA Lab.

3. 制御系の構成

Figure 4 に示すように電子光物理学研究センターの BST-ring 及びその入射器制御系のネットワークは BST-ring 制御関連のネットワーク、入射器制御関連のネットワークそして電気設備や冷却系などの施設のユーティリティ関連のネットワークの 3 つのネットワークから構成されており、加速器とその付帯設備(ユーティリティー)のほぼ全ての装置が EPICS CA 経由での操作・監視が可能となった。これによりシステムティックな運用が可能になった他、EPICS CA は言語に比較的依存しないため、様々なプラットフォームにおいて加速器装置の各パラメータの利用が容易になった。また、現在制御系とは切り離されているが、所内の放射線モニタリング用のネットワークも存在し、これも EPICS CA によって通信を行っている。基本的には異なるネットワークにまたがる EPICS IOC は存在しないが、異なるネットワークに存在する装置を統一的に扱う必要があったため加速器のタイミング系と入射器~BST-ring 間のトランスポートラインの EPICS IOC のみ両方のネットワークと通信できるように複数の NIC を所持している。

異なるネットワーク間は EPICS PV Gateway[5]によって接続されており、EPICS CA での通信が可能となっている。これにより、制御室内に存在する OPI とよばれるオペレータ用コンソールによって、制御系内の各装置の操作・監視が可能となっている他、Database Server によって、すべての EPICS IOC の PV のログを定期的に取り得、保存している他、主

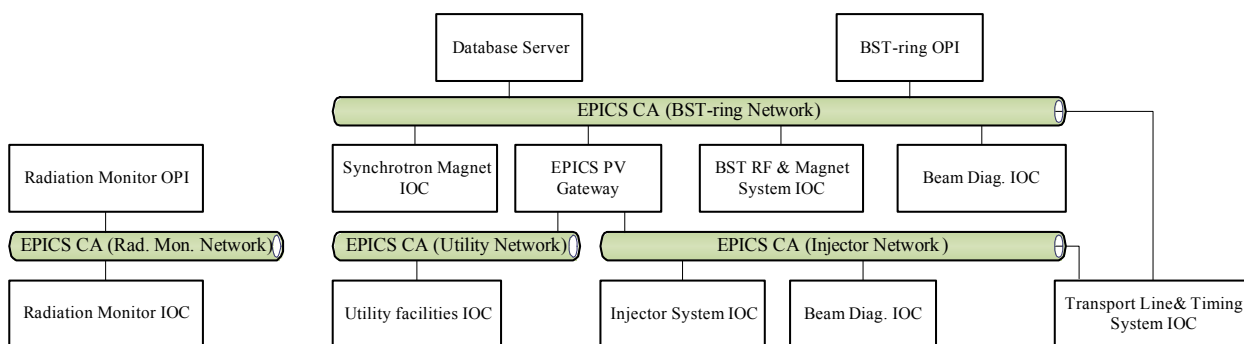


Figure 4: The network configuration of the BST-ring control system. The BST-ring control system has three local area networks; one is for BST-ring control, the rest for Injector and utility facilities. Almost all devices in the control system can be controlled with EPICS CA from BST-ring OPI.

要な PV の操作ログを収集している。

ネットワーク上の各 EPICS IOC は可能な限りデバイスドライバによる問題が発生しないよう制御装置を Ethernet 上で TCP/IP 通信できる装置を選定した。直接 Ethernet で制御できない GPIB や RS232 などの規格の装置は、GPIB-Ethernet 変換器や RS232-Ethernet 変換器を利用して Ethernet 経由で制御できる構成とした(Figure 5)。ただし、例外としてシンクロトロン電磁石電源の制御は National Instruments 製 PXI[6] の Field Programmable Gate Array (FPGA) モジュールを利用しているため、PXI の内部バスを経由して制御を行っている。

Figure 4 に示されている通り、EPICS IOC は制御系内に 8 台で、その合計 PV 数は 1345 個であり、制御されるハードウェアは Ethernet 経由で制御可能な各種装置に搭載された PLC が 13 台、電磁石電源が 32 台、PXI や Ethernet RIO[7] といった FPGA モジュール搭載型装置が計 7 台、特高変電所や冷却系といったユーティリティ設備関連の装置が数台の他、

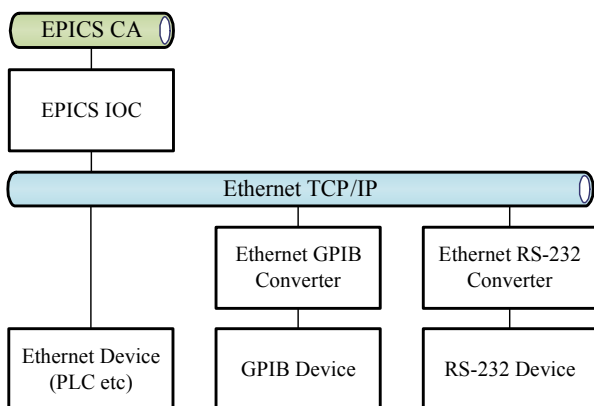


Figure 5: Control via the Ethernet. Almost all EPICS IOCs control device via Ethernet to reduce problems caused by device drivers in the BST-ring control system.

RS232 経由で制御可能なイオンポンプや真空計が計 24 台、GPIB 経由で制御可能な装置が数台という内訳になっていて、加速器施設としては小規模な構成となっている。

また、PXI 以外の EPICS IOC は全て、管理の工数削減と迅速な復旧を考慮して Hyper-V[8] 上の仮想マシンに構築されている。仮想マシンとすることで、物理マシンが故障した際にそのスナップショットを利用することで迅速な復旧が期待できる。また、仮想マシン以外の EPICS IOC が故障した際、もしくはプログラムが紛失した際に迅速に復旧するため、EPICS IOC 上のプログラムや設定ファイルをバックアップできる Git[9] を導入した。Git は、プログラムのソースコードなどの変更履歴を記録・追跡するための分散型バージョン管理システムである。すべての EPICS IOC からアクセスできる Git サーバーを構築し、そこで全ての EPICS IOC を構築するためのファイルを管理している。

4. 制御系の稼働状況

東日本大震災復旧作業完了後 2013 年 4 月の共同利用再開から現在まで、BST-ring の制御系は SoftIOC、CALab 自体の不具合で制御系が停止したことはなく、安定して動作している。ただし、Hyper-V 上の仮想マシンが複数の NIC を持つと極端にその処理性能が低下する現象が確認されたが、EPICS PV Gateway の導入により、仮想マシン上の NIC の数を可能な限り減らすことができたため、複数のネットワークにまたがる計算機を削減することができた。また、CALab は PV に対し同期的な書き込みを行う場合、極端な通信遅延を招くため実装方法には注意が必要であった。

その他、加速器運転中にシンクロトロン電磁石電源制御用 PXI のハードディスクが故障する事例が発生したが、Git によるバックアップを利用して迅速に復旧することができた。

5. 今後の展開

現在、管理を簡素化し、信頼性や処理速度を改善するために SoftIOC と CALab を用いた制御システムから、他の分散制御システム用のソフトウェアツールキットを利用した制御システムの移行を検討している。TANGO[10]と EPICS V4[11]を検討中で、TANGOは LabVIEW 用の環境が比較的整っているが、検討の結果、処理性能は現在の SoftIOC と CA Lab の構成より低下し、また利用方法に癖があり適切にプログラムを構築しなければプログラムが不安定になりがちであったため、経験の少ない職員にとっては扱いが難しいため、導入を見送った。

現在、EPICS V4 に LabVIEW 用の環境は整っていないが、これを LabVIEW から利用するために必要なプログラムを電子光物理学研究センターで開発中である。このプログラムは未だ開発中ではあるものの、SoftIOC と CA Lab の構成より処理性能が向上し、利用方法も簡素化されているため、これを導入することで開発時の作業工数や管理の容易さが向上するものと考えられる。

また、このプログラムにより具体的に SoftIOC と CA Lab の構成で生じている EPICS CA によるオーバーヘッドを評価することができるようになったため、Core i7-6500U 2.5GHz の windows マシンで PV の値を 10 万回変更するプログラムを SoftIOC と CA Lab の組合せによる構成と、新たに開発中の EPICS V4 用 VI で処理完了までの時間を比較したところ、SoftIOC と CA Lab の組合せでは 0.4s 要し、EPICS V4 用 VI では処理完了までおよそ 0.3s 要することがわかった。1 回あたりに換算すると 0.1 μ s のオーバーヘッドとなる。この結果だけで判断することはできないが、我々のような小規模な施設で PV 数が少ないケースでは LabVIEW で EPICS IOC 構築する場合、SoftIOC と CA Lab で構成しても、直接 PV にアクセスできる VI で構成しても処理性能にあまり差は出ないと推測できる。

その他にもログシステムの管理の簡素化、信頼性の向上そしてデータ利用時の利便性の向上、ネットワークの健全性向上を目指し、制御システムの更新計画を行っている段階である。

参考文献

- [1] LabVIEW; <http://www.ni.com/en-us/shop/labview.html>
- [2] EPICS; <http://www.aps.anl.gov/epics/>
- [3] CA Lab;
https://www.helmholtz-berlin.de/zentrum/locations/it/software/exsteuer/calab/index_en.html
- [4] LabVIEW EPICS I/O Server;
<http://www.ni.com/white-paper/14144/en/>
- [5] EPICS PV Gateway;
<http://www.aps.anl.gov/epics/extensions/gateway/>
- [6] PXI; <http://www.ni.com/pxi/whatis/>

- [7] Ethernet RIO; <http://www.ni.com/white-paper/14141/en/>
- [8] Hyper-V;
[https://technet.microsoft.com/ja-jp/library/hh831531\(v=ws.11\).aspx](https://technet.microsoft.com/ja-jp/library/hh831531(v=ws.11).aspx)
- [9] Git; <https://git-scm.com>
- [10] TANGO; <http://www.tango-controls.org/>
- [11] EPICS V4; <http://epics-pvdata.sourceforge.net/>