

EPICS を使った制御システム入門

中村 達郎*

Introduction to the Control System with EPICS

Tatsuro NAKAMURA *

Abstract

EPICS is the software toolkit to create control systems for scientific instruments such as accelerators. In this article, introductory review of EPICS for beginners is presented. Recent developments and activities in the EPICS collaboration are also reported.

1. はじめに

EPICS (Experimental Physics and Industrial Control System) は加速器や物理実験装置などの制御システムを構築するためのソフトウェア・ツールキットです¹⁾。そのルーツは 1980 年代末頃の LANL と ANL の共同開発に遡ります。日本では KEKB 加速器の制御システムで本格的に採用されたのが最初でした。その後 KEK の PF-AR, 理研の RIBF, J-PARC などでも採用されて来ました。

「EPICS」とはソフトウェア・ツールキットの名称であると同時に、それを開発する国際共同研究をも指しています²⁾。EPICS は世界中の加速器を運用する多くの研究所で採用されていますが、それらの研究所は EPICS の単なるユーザであるだけでなく、その開発・改良に大なり小なり関わっており、それら研究機関のコラボレーションによって EPICS は発展して来ました。現在 EPICS はオープン・ソフトウェアであり、無償でダウンロードして使うことができます。

EPICS は加速器の制御で多く使われ、開発の中心も加速器の研究所であるため、加速器に特化したソフトウェアと思われるかもしれませんが、そのようなことはなく、いろいろな装置の計算機制御に応用できる汎用性を備えています。後述するように、EPICS を利用可能なプラットフォームは近年広がりつつあり、ハードウェアの選択肢も充実して来ました。それに伴い

EPICS ユーザも大きな研究所だけでなく、小規模なシステムにも広がって来ています。本稿では EPICS の入門的な紹介をすると共に、最近の動向にも触れたいと思います。EPICS に興味を持って頂いて、EPICS に触れる足がかりに少しでもなれば幸いです。

2. EPICS の基本的構成

EPICS はネットワーク分散型の制御システムに適しています。適所に計算機を配置し、それらをネットワークで結んで全体として一つの制御システムを組み上げるとというのが EPICS の典型的な使い方です。ネットワークとしては広く普及している Ethernet がよく使われます。EPICS は標準的な IP (Internet Protocol) を使っており、特殊なネットワークを用意する必要はありません。

EPICS で使われる計算機は大きく 2 種類のカテゴリに分けられます。一つは IOC (Input/Output Controller) と呼ばれる計算機で、制御対象の装置の側に置かれ、何らかの信号線を介して装置を直接制御するものです。もう一つは OPI (Operator Interface) と呼ばれ、中央制御室などの運転員に近い所に置かれ、操作用のプログラムをはじめとする運転・制御用のアプリケーション・プログラムを実行する計算機です。IOC はハードウェア制御層を構成し、OPI は操作インターフェース層を構成します。これらを結び付けるネットワーク通信層を挟んだ 3 層構造が EPICS の標準的なモデルと言えます。以下では典型的な構成例を挙げて説明します。

* 高エネルギー加速器研究機構 KEK, High Energy Accelerator Research Organization
(E-mail: tatsuro.nakamura@kek.jp)

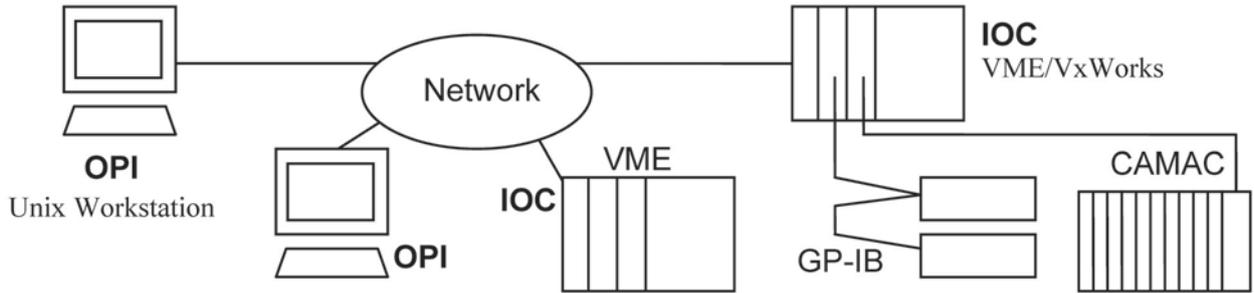


図1 VME 計算機を使った構成例

2.1 VME 計算機を使った構成例

図1はVMEシングルボードコンピュータをIOCとして使った例です。KEKB加速器の制御システムもこの構成で、約100台のVME計算機を使っています。デバイスとの接続にはVMEの各種インターフェースボードが用いられます。(図ではGP-IBとCAMACがインターフェースの代表例として単純化して描かれていますが、KEKBではARCNET, VXI/MXI, RS-232C等の多くの種類のインターフェースが使われています。)OSにはVxWorksがよく使われます。この場合ソフトウェアの開発には開発用のホスト計算機が別に必要でクロス開発となります。開発用ホストがOPIを兼ねることはよくあります。KEKB加速器ではOPI兼クロス開発用としてUNIX計算機を使っています。歴史的にはEPICSはVxWorksの使用を前提として開発されましたが、現在のバージョン(R3.14)では他のOSも使用可能です。

2.2 PC/Linuxを使った安価な構成例

ハードウェアにPC(PC-AT互換機)、OSにLinuxを採用したIOCが安価なシステムとして近年多く使われるようになってきました³⁾。PCはOPIも兼ねることができるため、開発も含めPC一台だけで完結するごく小規模な制御システムも構成が可能です。PCもLinuxも本来は制御用に作られたものではないため制約もありますが、安価で手軽なので、簡易システム、テストベンチ、EPICS学習用など用途によっては大変便利です。

2.3 PLCを使った組込みEPICSの例

近年コンピュータを組み込んでインテリジェント化した装置が多く使われるようになってきました。それならば組み込まれているコンピュータ上でEPICSを動かしてIOCとして機能させよう、という発想で生まれたのが「組込みEPICS」です。制御対象の装置の中にIOCが組み込まれている訳です。この場合、制御対象の装置がネットワークに直結されるので、外見上は

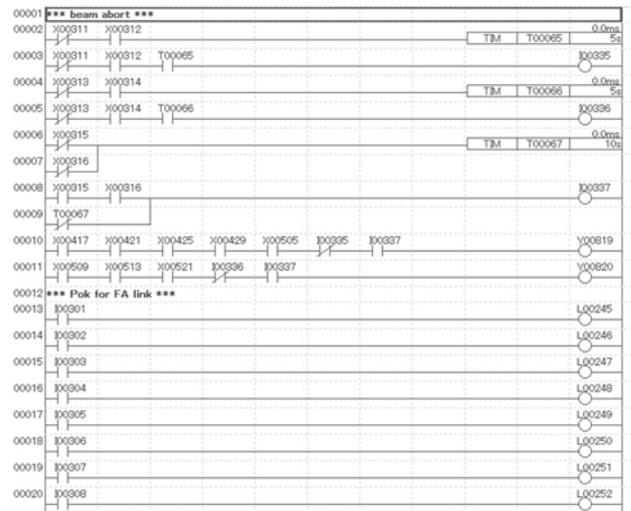


図2 ラダー言語の例。制御ロジックを図により視覚的に表現する。図は一部分を抜粋したもの。

VMEなどの制御計算機を省いた形となり制御システムのハードウェアを単純化することになります。

組込みEPICSは各種開発されていますが、ここではPLC(Programmable Logic Controller)を使った例を紹介しましょう。PLCはリレー回路の代替として使われる制御装置で、機械の自動制御などによく使われます。CPUモジュール、入出力モジュールなどを組み合わせて構成します。PLCは実体としてはコンピュータなのですが、通常は図2のようなラダー・ロジックまたはラダー言語と呼ばれるリレー回路を記号化した手法で制御ロジックを組むのが一般的です。しかし近年では一般のコンピュータと同様の自由なプログラミングが可能なものも出てきています。横河電機株式会社のPLC FA-M3シリーズのCPUであるF3RP61はLinuxをOSとして走らせることができます。この上でEPICSを走らせることでIOCとして使うことができます⁴⁾。図3は理研RIBFでの使用例です(写真提供: 込山美咲さん)。

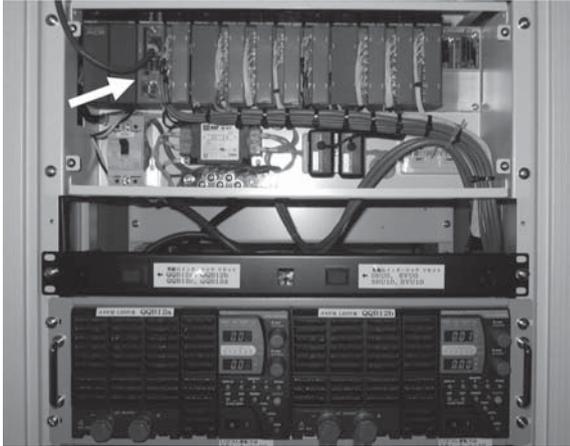


図3 F3R6P61 (矢印) を使った PLC の使用例

3. EPICS の基本的仕組み

EPICS はいろいろなソフトウェアの集合体であり、様々な機能があります。その中で最も重要で、EPICS の中心的な役割を担っているものが EPICS database と Channel Access です。この2つが EPICS を理解する上でのキーポイントになるので、ごく簡単に説明します。

3.1 EPICS database

EPICS database とは IOC に常駐するデータベースで、EPICS の中核となる部分です。データベースはレコードの集合です。典型的には一つのレコードは一つの入力点あるいは出力点に対応しますが、入出力点と対応せずに内部状態を保持したり、内部演算を行ったりするレコードもあります。レコードは次のような特徴を持っています。

- レコードには名前を付けて区別する。
- レコードには型があり、型ごとに固有の動作をする。
- レコードは型ごとに決められたフィールドを持っている。
- フィールドがデータアクセスの最小単位である。最も重要なフィールドは VAL フィールドで、入出力データ、演算結果などを保持する。
- レコード名とフィールド名でデータアクセスの対象を指定する。フィールド名はしばしば省略され、その時は VAL フィールドが仮定される。
- レコード間にはリンクと呼ばれる参照関係を定義することができる。リンクを通じてレコード間でデータを受け渡したり、動作を制御したりする。

3.2 Channel Access

Channel Access (CA) とは EPICS の基本的な通信

プロトコルで、レコードの情報にネットワーク経由でアクセスする仕組みです。CA はクライアント-サーバモデルによる通信を行ない、下位層 (IOC) がサーバ、上位層 (OPI) がクライアントとなります。また下位層同士でのデータアクセスにも用いられます。(したがって IOC はサーバであると共にクライアントにもなります。) CA の概要をごく簡単にまとめると次のようになります。

- データアクセスはチャンネル (Channel) とよばれる単位で行なう。
- 「レコード名. フィールド名」で指定されたデータ単位がチャンネルに対応する。この名前をチャンネル名という。
- クライアントはチャンネル名を指定してデータの読み (get) 書き (put) を行なう。このときアクセス先のデータ (レコード) がどの IOC にあるか (すなわち通信相手のサーバはどれか) を指定する必要はなく、チャンネル名のみで通信を行なえる。
- get, put のほかに monitor とよばれる機能もある。チャンネルの値が更新されたときサーバ側が通知する機能で、通知があるとクライアント側では予め登録したコールバックルーチンが呼び出される。

4. EPICS で制御システムを組むには

EPICS を使って実際に制御システムを組み上げていくのは、具体的にどのような作業になるのか見ていきましょう。

4.1 IOC の開発

とっておもおおまかに言えば IOC に制御システムを組み上げるには、EPICS database を定義することで行ないます。EPICS database の定義はテキストファイルで記述されるので、最低限テキストエディタさえあれば制御システムを組むことができます。あとは IOC の起動時に (OS を起動し、EPICS core を立上げ、そのあとに) このファイルを読み込んで EPICS をスタートすれば制御システムが動き出します。

テキストファイルを直接書き上げる以外に、グラフィカルな操作で EPICS database を定義していく VDCT というツールもあります。(歴史的には CapFast, GDCT, JDCT など何種類もありました。)

このようにプログラミング無しでも制御システムを組み上げることが可能なのが EPICS の大きな特徴です。ただし、ハードウェアと繋げる部分 (EPICS では Device Support, Device Driver と呼ばれる部分) は本来プログラミングが必要です。しかし、よく使われて

いるデバイスは既に誰かがプログラミングをしているのでそれを利用させてもらえば、全くプログラミング無しでIOCを組み上げることも可能なのです。一方、EPICSのコミュニティで使用実績のないデバイスを使う時は、残念ながら自分でプログラミングする必要があります。

IOCではこのほかにシーケンサと呼ばれるプログラムを走らせることがよくあります。シーケンサはSNL (State Notation Language) というC言語に似た文法でプログラミングするステートマシンで、イベント駆動型プログラムが容易に作れます。EPICS databaseだけでは記述しにくい複雑な制御を実現するためによく使われます。

4.2 クライアント・プログラムの開発

制御システムで一番よく使われるクライアント・プログラムは、状態を表示したりオペレータが操作したりするための操作パネル (ディスプレイ) です。操作パネルを作成・実行するアプリケーションはEPICSではDisplay Managerと呼ばれ、今までに何種類も開発されてきました。代表的なものにMEDM, EDMなどがあります。いずれもグラフィカルな操作でパネルを作成していくことができます。

EPICSではこのほか汎用に使える多くの種類のアプリケーション・プログラムが開発されています。例えば次のようなものがよく使われます。

- Channel Archiver … 指定したチャンネルの時系列変化を記録する。
- Strip Tool … 指定したチャンネルの時系列変化をグラフ表示する。
- Alarm Handler … 指定したチャンネルのアラーム状態を表示・通知する。

このような汎用のツールを活用することでOPI層でもプログラミング無しでの開発が可能になっています。しかし既製のアプリケーションでは不十分な場合は自分でアプリケーションをプログラミングすることになります。Channel AccessはC言語のライブラリを通して使うことができます。C言語以外にもJava, Pythonなど各種プログラミング言語のインターフェースが開発されていて、さかんに利用されています。KEKBではSAD⁵⁾ (Strategic Accelerator Design) によるアプリケーション開発が多くなされています。

5. CSS

EPICSの汎用アプリケーションは、いろいろな研



図4 CSSで作成した操作パネルの例

究所でそれぞれの必要に応じて独立に開発されてきました。そのためアプリケーション間の連携は無く、ユーザ・インターフェースのスタイルも統一性がありませんでした。そこで近年登場してきたのがCSS (Control System Studio) です。CSSは運転に必要なツール類を統一の枠組みに集めたもので、アプリケーション開発から実行まで一貫して行なえます。いわばオールインワンの汎用ツールと言えます。アプリケーション間の連携も図られています。CSSはEclipseをベースに作られています。CSSに標準で組み込まれているBOYと呼ばれるDisplay Managerは高機能であり、これを使う目的でCSSを使う人もいます。KEKBでも一部で導入を始めています⁶⁾。図4はCSSで作成した操作パネルの一例 (LLRF制御メイン画面) です。CSSはまだ整備の途上の面もありますが、今後の発展に注目したいと思います。

6. EPICSの歴史を振り返って

筆者はKEKB加速器の制御グループの一員としてEPICSに関わってきました。KEKB加速器の建設期の頃のEPICSはバージョンR3.12からR3.13に移行しつつある時期でした。R3.13以前のEPICSはVxWorksに強く依存しており、IOCのOSとしてはVxWorksしか選べませんでした。VxWorksは優れたリアルタイムOSですが、開発ライセンスは安くはなく、そのためEPICSはそれなりの規模の制御システム向けというイメージで、少数の比較的大きな研究所で採用されていました。

この状況が変わったのが次のバージョンのR3.14です。VxWorksに依存していたIOCのコードは見直され、

OS に依存する部分と OS に依らない部分とに分離されたことにより、いろいろな OS に対応でき、様々なプラットフォームに EPICS を移植できる道が開けました。その後はコラボレーションに参加する研究機関でそれぞれの必要に応じて（あるいは興味に応じて）いろいろな OS に移植されて行きました。プラットフォームの選択肢が広がると EPICS ユーザも増えていき、コラボレーションの輪も広がります。それがまた新たな開発の原動力にもなっています。EPICS では国際的なコラボレーションによる開発という手法がうまく機能していると言えます。

現在では加速器のような大型の装置だけでなく小規模の実験や組込みシステムにも使われ、様々な応用が試みられています。また研究機関だけでなく、企業の参加も進み始めています。自社の装置に EPICS を組み込むことで商品の付加価値を高め差別化を図る企業も出てきました。EPICS のエキスパートを擁して開発を請け負う企業もあります。

EPICS に広がりをもたらしたバージョン R3.14 は現在でも広く使われている安定バージョンです。次の R3.15 や更に先のバージョン 4 も開発が進められています。

7. EPICS の課題

EPICS の一番の弱点と筆者が考えるのは、学習に手間がかかる点です。EPICS は各種ツールの集合体からなっており、それらを開発した研究所も作者も様々です。商用ソフトウェアと比べてしまうと、全体の統一感に乏しく、雑然としていて初学者には取っ付きにくいことは否めません。中にはドキュメントも十分でないものもあります。また、同じ目的の似たようなツールが異なる研究所で別々に開発されていたりします。熟練者にとっては選択肢が広がって好ましい状況ですが、初学者にとっては迷いの元が増えるだけだったりもします。

制御ソフトウェア一般に言えることですが、システムを設計するに当たり、実現したい制御のロジックをどの層でどのように実装すべきかで頭を悩ませることはよくあります。EPICS ではその選択肢も多いため迷いは深くなります。EPICS database でロジックを実装するのが最も一般的ですが、よりハードウェアに近い Device Support に担わせることもできます。シーケン

サで実装することもできます。さらにはクライアント側（OPI 層）で処理させることも可能です。いろいろな解が可能ですが、適切な方法を選ばないと非効率になったり、保守性が悪くなったりするので、EPICS をきちんと使いこなすにはそれなりの知識や経験が要求されます。

学習に時間がかかるという EPICS の弱点を克服することはなかなか難しい課題で、これといった特効薬はありません。マニュアルや講習会を整備することは重要ですが、かなりの手間を必要とします。EPICS の発展が著しい昨今では、整備した端から内容が古くなっていくため、絶えざる改訂も必要になります。現実的には厳しいです。

最後に KEK で試みられている取り組みを 2 つ紹介したいと思います。一つめは 1 Disk EPICS と呼んでいる学習用の EPICS システムです。これは Disk（CD や DVD あるいは USB メモリ）に Linux と EPICS を入れ、Disk からブートして利用可能にしたものです。ノートパソコンなどでインストールすることなく EPICS を試してみることができます。（インストールして使うこともできます。）KEK では講習会で使う教材として幾度か作製されました。最新のものは Ubuntu Linux をベースに作られています。講習会と切り離して常時整備できると良いのですが、残念ながらなかなかそこまで手が回らないのが現状です。

もう一つは EPICS ユーザのメーリングリストです。（連絡先 <mailto:epics-users-request@ml.post.kek.jp>）これはマニュアルや講習会の不足をユーザ同士の口コミで補おうとするものです。疑問、質問、トラブルに陥って困っている事などを日本語で気軽に情報交換できるよう、最近（2012 年 5 月から）始めました。まだ KEK とその周辺の人々しか登録がありませんが、今後うまく活用していければと思います。

参考文献

- 1) <http://www.aps.anl.gov/epics/>
- 2) 山本昇, 「EPICS を巡る国際共同研究」, 加速器 Vol. 3, No. 4, 2006 (409-412)
- 3) 上窪田紀彦, 他, 「小型加速器向け EPICS 環境」, 原子核研究 49 巻 5 号 (2005) pp.1-7
- 4) 小田切淳一, 「EPICS on F3RP61 の開発と応用」, 加速器 Vol. 6, No. 4, 2009 (321-334)
- 5) <http://acc-physics.kek.jp/SAD/sad.html>
- 6) <http://www-linac.kek.jp/cont/epics/css/>