



## 5.15 原研 ERL-FEL 用制御系の更新 Upgrade of a Control System for the JAERI ERL-FEL

菊澤 信宏

日本原子力研究所 関西研究所 自由電子レーザー研究グループ

〒319-1195 茨城県那珂郡東海村白方白根 2-4

**Nobuhiro KIKUZAWA**

Free Electron Laser Laboratory, Advanced Photon Research Center, Kansai Research Establishment

Japan Atomic Energy Research Institute

2-4 Shirakata-Shirane Tokai-mura, Naka-gun, Ibaraki 319-1195 Japan

The accelerator control system used for the JAERI ERL-FEL is a PC-based distributed control system that has been in operation since 1992. Since an interface bus of the PCs is obsolete, interface boards for the PCs are difficult to obtain in recent years. Thus we have been developing the CAMAC controller with  $\mu$ ITRON operating system to replace the old PCs connected with CAMAC. We will introduce a Java and CORBA environment in the new control system. The control system upgrade, including hardware upgrading and applications rewriting, is described in this paper.

**Keywords : Accelerator Control System,  $\mu$ ITRON, Java, CORBA**

### 1. はじめに

原研自由電子レーザー用制御系 FELOWS (FEL Operator Window System) は独自開発された PC ベースのネットワーク分散処理型制御系であり、1992 年に運用を開始した[1]。しかしながら、運用開始から既に 10 年以上が過ぎているため PC などは生産中止となり、OS や開発ツールなどはサポート対象外となるなど、今後の機能の維持が困難となりつつある。また、最近の計測機器用のソフトが制御用 PC では使用できないなど、運用上も問題を生じている。これらの問題を解決するために制御系を更新することを計画し、 $\mu$ ITRON を OS として組込んだローカルコントローラを開発して既存の PC と置き換える方式にした[2]。本報告では、新しい制御系の概要について報告する。

### 2. 原研自由電子レーザー用制御系 FELOWS

FELOWS はオペレータが操作する 2 台の PC とローカルコントローラと呼ばれる CAMAC クレートに接続された 5 台の PC からなり、それらがネットワークでつながれている。ローカルコントローラは加速器室内に分散して配置され、CAMAC の DAC モジュールから各機器の電源をアナログ電圧で操作できるよう統一されている。コストなどの点から市販の PC を利用することとし、当時は広く用いられていた NEC 製の PC9801 シリーズを採用した。制御用ソフトウェアは Windows3.1 用ソフトとして Borland 社 Turbo Pascal で開発されており、サーバーおよびクライアントの両方の機能を持たせて同等な立場で相互に通信しながら処理を行う「P2P」型のネットワークとし、通信プロトコルには Ethernet を採用していた。

2000年に原研自由電子レーザーをエネルギー回収型に改造する際に、マグネットの増加に対応するため制御用ソフトウェアの変更を行った。この変更にあわせてPCのOSをWindows3.1からWindows95に更新した。ネットワーク接続がOSレベルでサポートされたため、通信プロトコルをそれまでのEthernetからTCP/IPに変更した。また、ファイアウォールを介してインターネットに接続したため、制御系ネットワーク内にウェブサーバを設置して制御系ネットワークの外部からウェブブラウザによる状態監視を可能とした[3]。また、電子メールを利用した自動警報発信システムを開発した。

これらの改良を行ったが、CAMAC用インターフェースボードは古い規格のままであり、現在のPCIバスに対応していなかったため、ハードウェアを更新することは困難であった。また、制御用ソフトウェアも16bitアプリケーションから32bitアプリケーションに変更するためには基本的な部分から大幅な変更が必要であり、新しいソフトウェアを新規に開発するのと同じぐらいの手間が必要であることがわかった。このため、現在まで古い制御用ソフトウェアを使用してきた。

### 3. 制御系の設計

#### 3.1 ローカルコントローラ

制御系の更新にあたって考慮した点は、(1)特定の計算機環境に依存しないこと、(2)将来の拡張性を考慮すること、(3)故障しにくいこと、(4)既存のCAMACモジュールを最大限に活用すること、であった。これらの条件を満たすため、リアルタイムOSとして産業用の組込みシステムに広く用いられている $\mu$ ITRONを組み込んだCAMACコントローラを開発し、CAMACの制御に用いているPCと置き換えることにした。産業用途として多くの実績を持つ $\mu$ ITRONをOSとして採用したコントローラ(ニチゾウ電子制御製ND-MCU)をLCUとして選択した。ND-MCUはCPUにSH7751(SH4)、動作クロック200MHzを採用している。このため、特別なシャットダウン操作無しに電源をON/OFFできる。また、OSとしてWindowsなどを使ったPCと比べてハングアップの心配などもほとんど無いと期待でき、将来にわたってOSのバージョンアップの必要などが無い。この方式ではCAMACモジュールは現状のまま使用しながらLCUを更新でき、CAMACモジュールから各電源までの配線作業は必要なく、しばらく新旧のシステムを共存させておいて切り替えながら動作確認することができるという利点がある。このCAMACコントローラはTCP/IPによってPCと通信が可能であり、PCの環境には依存しない。また、HDDや空冷ファンなどの駆動部を持たないため故障の確率も低くでき、拡張スロットとしてPCIスロットを3スロット装備している拡張PCIスロットを利用すれば将来の機能拡張にも対応できる。これによって既存のCAMACシステムを最大限に活用しながら、機能の向上を図ることができる。

LCUにはPCIスロットがあるため、CAMACクレートコントローラ(東陽テクニカ製CC/7700)用PCIインターフェースボード(東陽テクニカ製CC/PCI)をLCUに挿して使用し、LCUからCAMACにアクセスする。新しい制御系のハードウェアの構成を

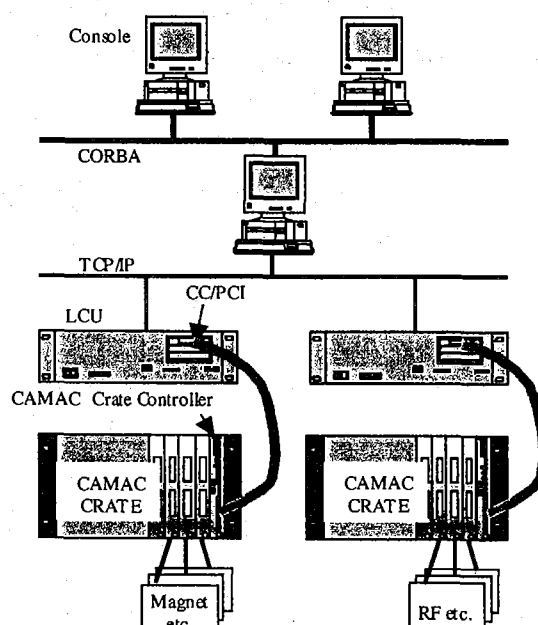


図1 新しい制御系のハードウェア構成

図1に示す。LCU から CAMAC クレートコントローラにアクセスするために CAMAC インターフェースボードのドライバを開発する必要がある。また、TCP/IP で送られた制御コマンドを実行し、バッテリーバックアップされたメモリ上の設定値テーブルに CAMAC モジュールなどの設定値を保存するためのコントロールソフトを開発する必要がある。このコントロールソフトによって LCU の起動時には設定値を自動的に復元できるような機能を持たせることを考えている。LCU は一種のネットワーククレートコントローラとして自律的に動作し、ホストコンピュータとは独立して制御を行えるようになる。図1にハードウェア構成を示す。

現在の制御系で CAMAC による制御を行っているのは、主に電磁石電源と電子銃であり、現在はマニュアルで制御している RF もリモートで計算機制御できるように変更を行っている。これらで5台の CAMAC クレートが使われており、5 台の LCU を使用する予定である。

### 3.2 制御用ソフトウェア

制御用ソフトウェアは Windows3.1 上の 16 ビットアプリケーションとして Borland 社 Turbo Pascal で開発され、その後、Borland 社の Delphi 1.0 に移植された。しかし、現在の Windows2000/XP 用に移植するためには、OS の仕様の違いによってインターフェースボードのドライバから共有メモリやタスク管理などに関係した部分や、開発ツールの違いによって GUI に関係した部分まで、ほとんど全ての部分で修正が必要であった。このため、LCU を更新するにあわせて制御用ソフトウェアは新規に開発することにした。PC の OS の選択であるが、ソースコードが公開され自由に利用できる Linux を利用する例が増えている。しかし、いくつかの計測器は Windows でしか利用できないものもあり、OS が混在することは避けられない。このため、さまざまな環境での互換性を最も重視し、Java を選択した。Java はオブジェクト指向言語の一つで機種依存性の非常に少ない言語であり、マルチプラットフォームに対応したソフトウェアが構築できる。Java は加速器制御の分野でも利用され始めており、ハードウェアに依存しない部分や、あまり制御速度を必要としない部分は Java で構築する予定である。これによって将来のアップグレードも容易になると期待できる。

通信関係のミドルウェアには分散オブジェクトである CORBA(Common Object Request Broker Architecture)を利用する予定である。CORBA とは米 OMG(Object Management Group)が規定する、分散システム環境でオブジェクト同士がメッセージを交換するための共通仕様であり、マルチプラットフォーム環境で OS や開発言語に依存しないクライアントサーバアプリケーションの開発が容易になる[4]。図 2 に電磁石電源制御用プログラムの表示例を示す。

### 3.3 ラッピングソフト

LCU は CORBA プロトコルによる通信ができないため、制御用ソフトウェアから CORBA プロトコルで送られた制御コマンドを TCP/IP に変換するラッピングソフトが必要である。新しい制御系のソフトウェア構成を図 3 に示す。コンソールの制御用プログラムから送られたコマンドは CORBA からラッピングソフトによって TCP/IP のコマンドに変換され、LCU 内部のコン

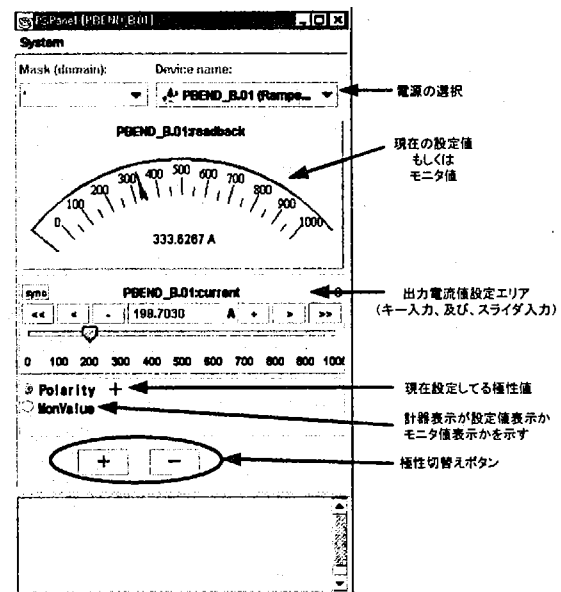


図 2 制御用プログラムの表示例

トロールソフトに送られる。このラッピングソフトは各 LCU に関するパラメータ定義ファイルを管理しており、コンソールの起動時に各パラメータを設定する。電源の数が増えた場合などはこのパラメータ制御ファイルを変更するだけで対応できる。

#### 4. まとめ

組込み OS として広く用いられている  $\mu$ ITRON を使った CAMAC コントローラを開発し、古い PC を更新する予定である。組込み OS を利用することにより、OS のアップデートやバージョンアップの必要がなく、安定して動作しつづけることが期待できる。また、ローカルコントローラとコンソールで機能分離を行うことにより、PC に依存せずにローカルコントローラだけで動作することができる。

制御用ソフトウェアは Java&CORBA で構築するため、機種依存性が少なくなる。このため、将来の計算機のアップグレードも容易になると考えられる。

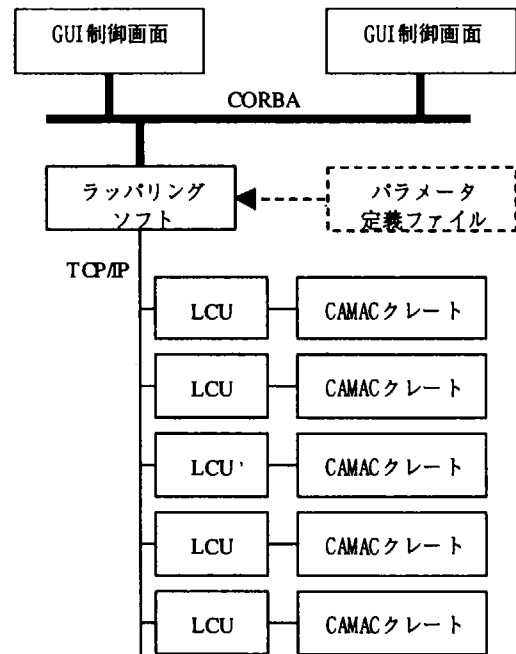


図3 新しい制御系のソフトウェア構成

#### 参考文献

- [1] M. Sugimoto, Nucl. Instr. and Meth. A331 (1993) 340
- [2] N. Kikuzawa, Proceedings of the 14th Symposium on Accelerator Science and Technology, Tsukuba, Ibaraki, Japan, Nov., 2003, p.269.
- [3] N. Kikuzawa, Proceedings of the 28th Linear Accelerator Meeting in Japan, Tokai, July 30- Aug. 1, 2003, p.437
- [4] T. Tanabe et al., Proceedings of the 14th Symposium on Accelerator Science and Technology, Tsukuba, Ibaraki, Japan, Nov., 2003, p.275.