

原研ERL-FELの制御系更新計画

菊澤 信宏¹

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2-4

概要

原研自由電子レーザー用制御系は、1992年から使用を開始し、何度かの大きな変更を行いながらも基本的な部分は変更せずに使用してきた。しかしながら、PCとインターフェイスボードが入手困難となり、ハードウェア、ソフトウェアの更新が必要となっている。本研究会では、今後の更新計画の概要とシステムについて述べる。

1. はじめに

原研自由電子レーザーは1992年から運転を開始し、その制御系として安価なPCを用いたネットワーク分散処理型制御系としてFELOWS(FEL Operators Window System)を開発してきた^[1]。機器とのI/OにはCAMACを用い、PCには当時広く普及していたNEC製PC-9801を採用した。しかしながら、PC-9801やPC-9801用の古い規格のインターフェイスボードが入手困難となり、ソフトウェアのバージョンアップも困難となっている。このことにより新しく導入した計測制御機器用のプログラムが動作しないなどの問題が生じはじめている。このため、現在使用しているローカルコントローラを更新し、それに伴って現在の制御系を更新することを検討している。

今後の制御系を改造するための要求として、できるだけ特定のプラットフォームに依存しないシステムとし、PCの進化やOSのバージョンアップからなるべく影響を受けないことを目的とする。このため、開発ツールやサーバソフトなどは広く一般的に用いられているものを利用することなどが必要である。

2. 現在までの制御系の概要

2.1 制御系インターフェース

原研自由電子レーザーでは、制御系として大きく分けて、電子銃、電磁石電源、超伝導加速器冷却系などがあるが、装置とのI/OにCAMACを使用している。レコーダなどのGPIB機器についてもCAMACのGPIBインターフェースモジュールを介して通信を行っていた。しかしながら、レスポンスが要求される電磁石電源の制御とI/Oの遅いGPIB機器が混在するなど問題点も多く、制御用PCに接続されるものは最小限に削られ、RF制御などはマニュアルで行われていた。またCAMACでは電圧出力の設定値などを読み出すことができないため、CAMACを管理するPCがハン

アップしたときなどは、設定値が失われるなどの問題が発生していた。

2.2 制御系プログラム

制御用PCとして7台のNEC製PC-9801(CPU: 486DX2 66MHz、RAM: 64MB)を使用したネットワーク分散型制御を行っているが、5台はCAMACやGPIBとのインターフェース用のローカルコントローラとして、2台は制御室でオペレータとのインターフェースをおこなうコンソールとして使用している。これらはEthernetで接続されているが、セキュリティなどの点から他のネットワークとは独立させていた。全体の構成の概念図を図1に示す。

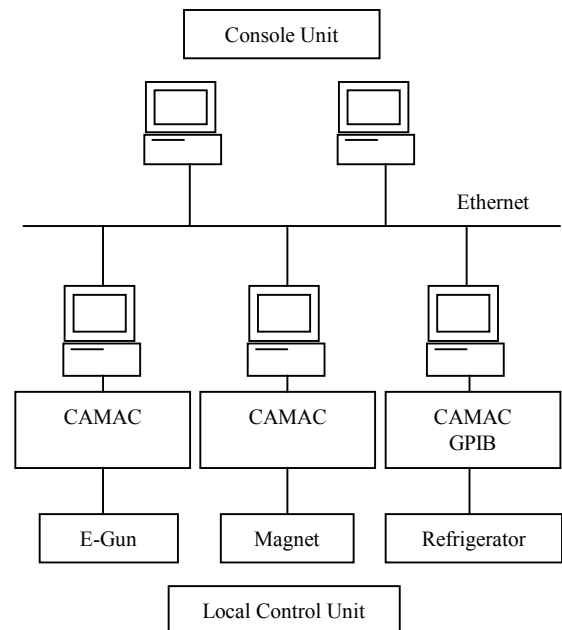


図1：FELOWSの概念図

制御プログラムは実行時に設定ファイルによってコンソールユニットとローカルコントロールユニットに分かれるが、全て同じバイナリプログラムで動作し、機能に制限をつけない構成となっている。このことにより、加速器室に設置されたローカルコントロールユニットからも全ての操作が行えるという利点があり、加速器室での機器の動作確認などの作業が容易になっている。一方で、全てのPCのスペックやOSのバージョンなどが同等である必要があり、最新のPCを導入することが困難であるという問題が

¹ E-mail: kikuzawa@popsvr.tokai.jaeri.go.jp

あった。

FELOWSではインターフェースを管理するドライバレイヤ、機器情報を管理するデータベースレイヤ、オペレータとのインターフェースであるアプリケーションレイヤの3層構造となっている。Object Pascalで開発されたこれらのプログラムはオブジェクト指向で構築されており、アプリケーションレイヤはWindows上のグラフィカルユーザーインターフェースを持っている。これらはWindows3.1時代の16bitアプリケーションであるため同一のメモリ空間で動作し、アプリケーション間でメモリ共有などが容易であった反面、メモリ保護が完全ではなかったため、ハングアップなどに対してシステムが保護されないという欠点があった。しかしながら、32bitプログラムへ変更するためにはメモリ管理やマルチタスクに関する根本的な部分から変更が必要であり、使用していた開発環境も32bit化するためには大幅に変更しなければならなかったため、電磁石電源の制御などの主要なプログラムに関しては現在でも16bitアプリケーションを使用している。

ネットワークに関して、開発当初のWindows3.1ではOSレベルではサポートされていなかったため、初期のFELOWSではネットワークボードのドライバを開発し、Ethernetによる通信を行っていた。しかし、Windows95以降にはTCP/IPによる通信がOSレベルでサポートされるようになってきたため、2001年に加速器をエネルギー回収型へ改造した際に制御系もネットワーク通信の部分でWinsockを利用するように変更を加え、制御用PCのOSをWindows95にバージョンアップした。通信プロトコルは通信速度の点からデータグラム通信(UDP/IP)を採用した。

測定データを制御系LANの外部から利用したり、状態を監視したいという要求があったため、制御用LANをルータを介してインターネットに接続するように変更した。これによって、個々のローカルコントローラからインターネットへのアクセスが可能となり、機器の異常などの緊急情報は電子メールによって携帯電話などのモバイル機器などへ送信することが可能になった。例えば、超伝導加速器用の小型冷凍機が故障した場合などは電子メールによって担当者に自動的に通報が行われ、早期の対応が可能となっている。

3. 新しい制御系の概要

3.1 ローカルコントローラの更新

PCがインターネットに常時接続されるようになったことによって、セキュリティの面では今まで以上に注意が必要になっている。適切に管理されていれば安全は確保できるであろうが、PCの数が増えてくると適切に管理していくことが困難になってくる。また、できるだけ故障などを避けるためにも、HDDなどを持たないハードウェアを選択することが望ましい。このため、ローカルコントローラとして多数のPCを利用することは好ましくないと考えている。

最近では加速器の制御用にシーケンサ(PLC)をI/Oに使用した例も増えており^[2]、今回の改造計画でもシーケンサへの変更を検討した。しかし、シーケンサに変更するためには現在の配線を変更しなければならないという欠点がある。他に、既存のCAMACとの互換性を考慮したローカルコントローラの採用も検討している。PCと同等の機能を持ったディスクレスのコントローラによってCAMACとのI/Oを行い、上位のPCとはTCP/IPによって通信を行う。このコントローラにCAMACの設定値を不揮発メモリに保存し、電源のON/OFFによって設定値が失われることを防ぐ機能も持たせることによって、設定値が読み出せないというCAMACの欠点を補うことができる。現在、これらのコントローラの比較検討を行っている。

3.2 GPIB機器との接続

現在でも古い記録計やオシロスコープなどとの通信ではGPIBが必要となることが多い。以前は、これらのGPIB機器との通信にCAMACのGPIBインターフェースモジュールを使用していたが、他のCAMACモジュールへの同時アクセスなどで障害が起こる確率が高いなどの問題があった。このため、現在ではLAN経由でデータをモニタするものに関してはイーサネット/GPIBコントローラ(National Instruments製 GPIB-ENET/100)を使用するように変更し、GPIB機器へのアクセスを独立させた。例えば、超伝導加速器用小型冷凍機の状態監視や温度モニタなどは記録計からGPIB経由でモニタしているが、GPIB-ENET/100に付属のドライバを利用することにより、ネットワーク通信を意識することなく複数のPCからデータの読み出しが可能となっている。

3.3 開発環境

開発のための環境は、当初からPascal言語を使用していたことから、現在でもボーランド社のPascal言語であるDelphiを中心に使用している。Delphiは統合環境によってWindows上のGUIプログラムの開発が容易であり、最近ではLinux上でもボーランド社のKylinxを使用すれば同様の開発環境が得られるようになってきた。DelphiとKylinxでソースコードの共有化も可能になるという利点がある。研究グループ内にもPascal言語に習熟した人間が多いことから、今後もPascal言語を中心に開発していくことになると考えられる。

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)に変更することを考えている。CORBAとは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。この場合、標準でCORBAをサポートしているJavaの利用を検討する必要があると考えられる。Javaはオブジェクト指向言語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

したシステムが構築できる。さらに、FELOWSでは全てのPCに制御プログラムをインストールしておく必要があったためプログラムのバージョン管理が煩雑であるという問題点があったが、Javaではネットワークを通じてプログラムの配布が可能であり、バージョン管理が楽になるという利点がある。

開発環境としてDelphi (Kylix)やJavaを利用することにより、WindowsやLinuxなどが混在した環境でもソースコードの共通化がある程度可能となる。このため、研究グループ内でソースコードを共有する必要があるため、バージョン管理ソフトウェア (CVS²)の導入を検討している。

3.4 ウェブサーバの導入

原研ERL-FELでは、超伝導加速器の冷却系に小型冷凍機を利用して24時間連続の無人運転を行っている。これを夜間などに自宅から状態を監視する目的でウェブサーバ (Apache³)を導入し、WWWによる状態表示を行っている。Webブラウザによる状態監視の例を図2に示す。セキュリティの点から、制御系LANとインターネットの間にFirewallを入れ、外部からのアクセスはHTTPプロトコルのみに制限している。

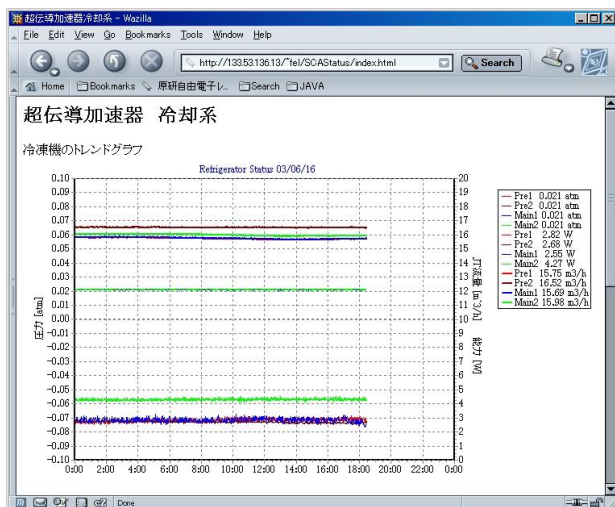


図2：状態監視用WWWモニタ例

3.5 データベースシステムの導入

現在は計測したデータや設定値などのログは個々のPCのHDDにファイルとして保存され、ファイルが保存されたディレクトリを共有ディレクトリとして公開し、参照している。しかしながら、効率的なデータ共有のためにはデータベースシステムの導入が必要であると考えられる。このため、データベースシステムとしてリレーショナルデータベースシ

テム (MySQL⁴)の導入を検討しており、データベースサーバに集中的に情報を集めることを考えている。これにより、情報の比較・検討が容易になると考えられる。

4. まとめ

ローカルコントローラを更新することにより、これまで大きく制限を受けていたPCを最新の機種に変更することが可能になる。これによってシミュレーションコードやデータ解析コードとの統合が容易になるであろう。

現在は制御用PCに接続されていない機器があり、今後はそれらを統合していく作業が必要となっている。また、加速器の改造を頻繁に行っていることもあり、わかりやすく、変更が容易にできる制御系を構築する必要がある。このため、できるだけデファクトスタンダードとなっている環境や開発ツールを選択することが重要であると考えている。

参考文献

- [1] M. Sugimoto., Nucl. Instr. and Meth. A 331 (1993) 340
- [2] R.Kato, et al., Proceedings of the 27th Linear Accelerator Meeting in Japan, Kyoto, Aug. 7-9, 2002

² <http://www.cvshome.org/>

³ <http://httpd.apache.org/>

⁴ <http://www.mysql.com/>

