

東北大学リニアック制御系改造

柴崎 義信、武藤 正勝、七尾 晶士、濱 広幸、小山田 正幸^{A)}

東北大学理学研究科附属原子核理学研究施設

〒982-0826 仙台市太白区三神峯一丁目 2 - 1

^{A)}National Synchrotron Research Center

111 University Ave., Muang District, Nakhon Ratchasima 30000 Thailand

概要

東北大学リニアック制御系は、昭和50年度にMELCOM-70を、昭和62年度に μ -VAX IIの導入を行い制御の計算機化により一定の成果をあげてきた^{[1][2][3]}。一方ここ数年の技術の進歩は目覚ましいものがあり、高性能のパーソナルコンピュータが安価に手に入る時代となった。ネットワーク技術が一般化しパーソナルコンピュータを簡単にネットワークに接続することも可能となっている。 μ -VAX IIを中心としたこれまでの制御系は、すでに15年ほど使用されており周辺機器の製造打ち切りなどにより、故障したときの問題解決が難しくなっている。そこで、平成12年度に μ -VAX IIを中心とした制御系をWindows2000が動作するパーソナルコンピュータへ置き換える作業を行った。開発環境は、これまで主だったC言語から、VisualBasicを開発言語とし、制御管理システムとしてCOACK^[4]を使用することによりプログラム生産性と安定性を確保している。更新作業は平成13年4月に終了し、5月から通常運転に入った。今回の研究会では更新作業、及び今後の計画について報告する。

1. はじめに

昭和62年度に行った改造により、リニアックの計算機制御化が一段と進展したが、一部の機器が手つかずに残ってしまった。プログラムの更新あるいは開発も、主にC言語を使用したため多くの人に開発を援助してもらうことが困難となり効率の悪い結果となった。また、器材に関して一部所内で製作しなければならなかったため、最近の新しい技術へ移行するためには新たな開発が必要であった。一方最近は、技術の進歩により様々な器材、ソフトウェアが開発され安価に提供されている。そのなかには、簡単にリニアックの制御のために応用できるものが数多く市販され、それらを使用することにより開発効率をよくし将来の変更にも柔軟に対処可能となっている。そこで、今回制御系を改造するにあたって次のような目標を立てた。

- ・ パーソナルコンピュータを全面的に使用
- ・ Windows2000 と VisualBasic による開発
- ・ 市販のプログラムコンポーネント利用
- ・ 制御管理ソフトの導入
- ・ 全てのリニアック機器の計算機制御への組み込み

これらの目標を段階的に達成することとし、平成12年度は主に、制御用サーバーとネットワークの更新および緊急を要する機器について更新を行った。

2. 更新機器の構成

2.1 制御サーバーとネットワーク

今回の更新により、複数台のパーソナルコンピュータがネットワークによって接続される。プログラム開発者あるいは、リニアックのオペレーターはどのコンピュータからでも同じように動作させることができる必要がある。そのためには、個別にコンピュータとユーザーを管理することは能率的ではない。そのためコンピュータとユーザーを管理するためにWindows2000のドメインサーバーを導入した。また、複数の人によるプログラム開発のため、あるいは単一プログラムを複数のコンピュータで動作させるためには共通の領域を使用してお互いのファイルを共有する必要がある。そのため、このコンピュータにはファイルサーバーの機能も持たせた。

制御管理ソフトには、昨年 KEK で開発されたCOACKを採用した。COACKの使用によりそれぞれのコンピュータからのデータ保存とメッセージの配信が簡略化され、プログラム開発が簡単になった。COACKサーバーとして動作するコンピュータは、全てのメッセージ処理を行うためある程度の負荷がかかる。そのため独立の速度の速いコンピュータを選択した。以下に、今回使用したパーソナルコンピュータの仕様を、図1に構成を示す。

ドメインサーバー

DellWorkStation PIII600MHz

512MB メモリ

OS-Windows2000Server

COACKサーバー

DellWorkStation PIII933MHz×2CPU

1GB メモリ

OS-Windows2000Server、SQL2000

クライアント

PIII300MHz 以上のコンピュータ

256MB メモリ

OS-Windows2000Professional

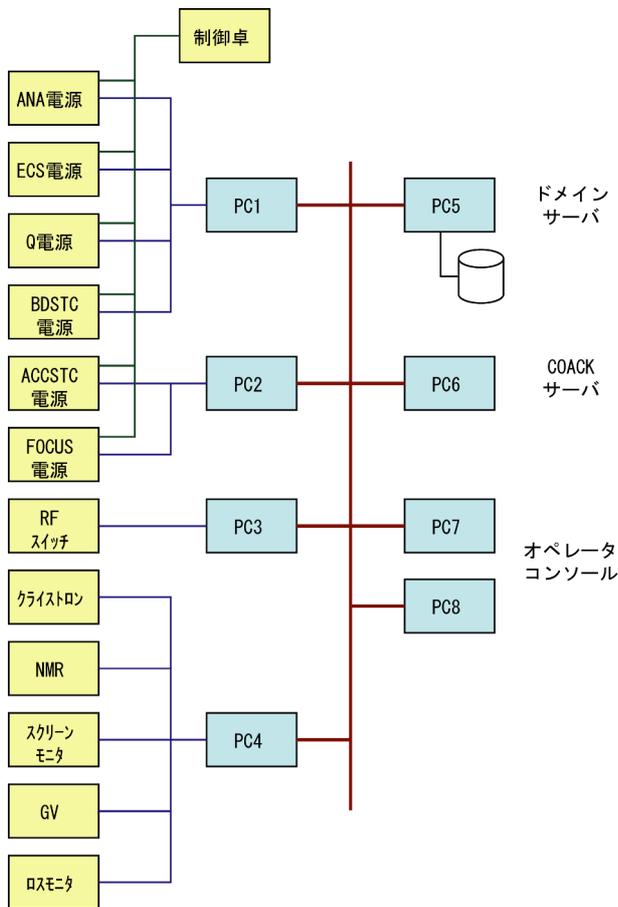


図 1 : システム構成図

2.2 デバイス接続

表 1 に今回の制御系に移行した IO のリストを示す。各機器 (電源等) は、昭和 62 年度の改造によって、 **GPIB** やその他の汎用インターフェースを持ったものに更新されている。そのインターフェースを使用して接続されたコントローラによって機器を制御していた。コントローラと制御用ネットワークは **RS232C**

デバイス名	デバイス数	I Oの種類			PC番号
電磁石電源					
アナライザー電源	1	電流値	ステータス	電流設定	1
ECS電源	1	電流値	ステータス	電流設定	
Q電源	8	電流値	ステータス	電流設定	
BDステアリング電源	10	電流値	ステータス	電流設定	
ACCステアリング電源	22	電流値	ステータス	電流設定	2
フォーカスコイル電源	6	電流値	ステータス	電流設定	
RF					
RF波形モータ切替スイッチ	1	ステータス	切替		3
クライストロンモジュレータ	5	電圧設定値	ステータス	電圧設定	
モニタ					
スクリーンモニタ	8	ステータス	ON/OFF設定		4
ビームロスモニタ	11	モニタ値			
アナライザ磁場測定	1	測定磁場	測定指示		
真空					
ゲートバルブ	5	ステータス			

表 1 : IO リスト

により接続していた。このコントローラも製造中止となっており更新する必要があったが、制御卓がコントローラをとおして接続されているため、単純に置き換えることが困難である。そのためコントローラは従来のものをそのまま使用している。コントローラとの接続は、マルチポートの通信ボードを使用した。コントローラが間に介在していない機器や PLC は直接汎用インターフェースによってパーソナルコンピュータへ接続した。

3. ソフトウェア

3.1 開発環境とコンポーネント

開発のための環境は、VisualStudio6.0 エンタープライズ版を用い VisualBasic により開発を行った。Basic 言語は初心者向けの言語として多くの人に使われている。最近ではオブジェクト指向であるクラス構築や ActiveX コンポーネント開発など高度なプログラミングが可能のように機能的にも強化されており、開発されたプログラムの実行速度もかなり改善されている。なにより Visual なツールを用いて簡単にプログラムを行うことができ、この言語のための様々なコンポーネントが市販されていることも魅力である。また独自のコンポーネント開発も Wizard の使用により容易に行うことができる。

このプログラム開発のために二つの市販コンポーネントも使用した。

MeasurementStudio は GUI の部品や、GPIB、DAQ などのボードを制御するドライバ等の ActiveX コンポーネントを提供している。特に GUI のプログラミングにはボタンやスライダー、グラフなどの表示を多用し、見易く操作しやすい画面を作る必要がある。これらの部品を使うことにより、誰でも簡単にコンソール画面を作ることが可能である。

FA-Engine は、PLC との通信を行うコンポーネントである。PLC と通信を行うためには、それぞれ固有のコマンドを使う必要がある。また、内部リレーの名前や取り扱いも異なっている。現在は、それらの違いを吸収するためにタグによってリレー番号を指定する方式になっている。そのようなソフトも数種類販売されている。今回は、ひとつのコンポーネントで複数のメーカーの PLC を扱うことが可能なこのコンポーネントを選択した。

この他に、COACK サーバと接続、通信を行う ActiveX コンポーネントを作成し COACK の利用をより簡単に行うことが出来るようにした。COACK 自身も接続コンポーネントを提供しており比較的簡単に使用することが可能である。しかし若干のコード追加や接続、切断での決められた手順等がある。今回作成したコンポーネントを使うことにより追加コードを減らし更に接続・切断の手順を意識する必要がなくなった。

3.2 デバイスクラス

COACK は内部に機器構造を DeviceClass として定義している。デバイスクラスは各デバイス名とノード名を"."を用いて接続しひとつのデバイスを表している。(例 Linac.Rf.Klystron.Kly01.Property) また、デバイスの値を入れるためのプロパティが各ノードに割り振られており、各クライアントはその定義されたデバイスのプロパティによってメッセージ交換を行う。また定義は XML ファイルとして保存されており、容易に変更ができる。今回の改造では、リニアック機器構成をそのまま DeviceClass として定義した。構成を表 2 に示す。

Device Name		Node Name	Node Number		
Linac	Gun	Cathode	1		
		Grid	1		
		Heater	1		
	Rf	Klystron	Kly01~Kly05	5	
			Driver	1	
			Switch	1	
			Driver~K5	6	
			Attenuator	1	
	Magnet	Phase	K1~B12	26	
			FocusCoil	G1~A13	6
			SteeringCoil	Acc1~Acc22	22
			Q	Q1~Q8	8
			Bd	Bd1~Bd10	10
			AnaMag	AnaMag	1
			Nmr	Nmr	1
			EcsMag	EcsMag	1
			M5	M5	1
			DegMag	DegMag	1
	Monitor	Screen	Screen1~Screen10	10	
			LmA12~LmEcs	11	
	Vacuum	IonPump	Ip1~Ip33	33	
			Gate Valve	Linac1~Exp3	6
SubPump			Exp1~K4	3	
Operation	Information	1			

表 2 : デバイスクラスとノード数

4. 稼動状況

更新作業は 4 月に終わり、5 月中旬からライナックの共同利用再開とともに稼動している。運転時に動作するプログラム数は、COACK のセッションマネージャーによりモニターすることが可能である。それによるセッション数はオペレーションコンソールのウインドウ数にもよるが、15 から 25 の間である。その時の、COACK サーバーの CPU 負荷は約 15% から 50% くらいの範囲で変動している。プログラム作成中のテスト段階でのこの値は、50% を超えていた。その原因は、ロスモニタの信号を AD 変換した値を時間的に制限せずに取り込み COACK へ書き込んでいたためであった。そこで、一定間隔でイベントを起こすように改善し、また COACK へデータを転送する場合も、直前に書き込んだ値と同じ値は書き込まないように作り直した。現段階ではこの CPU 負荷で特に問題とはなっていない。ライナックの共同利用を行っている時の GUI 画面を図 2 に示す。

5. まとめ

今回の作業で、旧制御系の更新作業は終了した。運転再開してから、約一ヶ月経過したが、メンテナンスのためのサーバー停止と、制御プログラム以外のタスクが引き起こしたと思われる一度の停止以外は問題なく安定に動作している。しかし、機器を直接制御しているコントローラが 15 年前の古いものであるため、新しい制御系とのデータ転送速度が遅く、旧制御卓に頼らざるを得ない状態である。また、それらコントローラの保守ができない状態となっており早急に改善することが必要である。

今年度は、上記のコントローラを廃止あるいは別な方式 (PLC 等) への変更と制御卓との接続方法の改善を予定している。

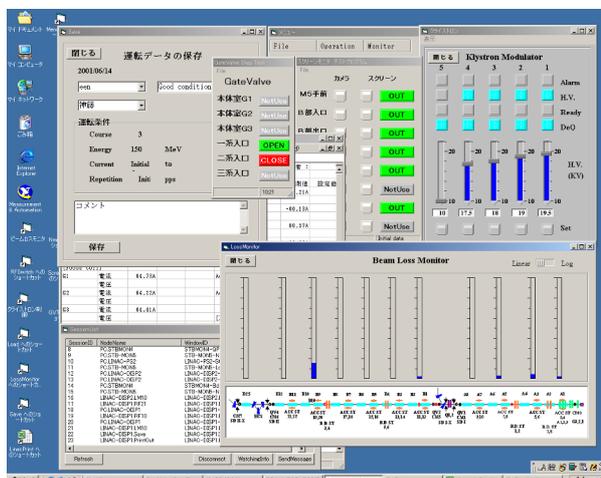


図 2 : オペレータコンソール GUI

参考文献

- [1] 柴崎義信、他、“第 4 回ライナック研究会報文集”，関西地区大学セミナーハウス，1979/7/19-21
- [2] Masakatsu Mutoh, et al., “Improvement of the control system for Linac and Pulse Beam Stretcher at Tohoku University”, Proceedings of the 13th Linear Accelerator Meeting in Japan”, Tsukuba, Sep. 7-9, 1988
- [3] Tadahiro Oonuma, et al., “The control of KLYSTRON MODULATER by VME”, Proceedings of the 14th Linear Accelerator Meeting in Japan”, Osaka, Sep. 7-9, 1989
- [4] I.ABE et al. “Status of Component ware Oriented Accelerator Control Kernel (COACK)”, 3rd International workshop on Personal Computers and Particle Accelerator Controls Oct 9-12, 2000, DESY, Hamburg, Germany