

DEVELOPMENT OF SOFTWARE USING EMBEDDED EPICS AT KEK INJECTOR LINAC

Shiro Kusano^{#, A)}, Naoyuki Toyotomi^{A)}, Shinji Ushimoto^{A)}, Kazuro Furukawa^{B)}, Masanori Satoh^{B)}, Yoshiharu Yano^{B)}

^{A)} Mitsubishi Electric System & Service Co., Ltd.

2-8-8 Umezono, Tsukuba, Ibaraki, 305-0045

^{B)} High Energy Accelerator Research Organization (KEK)

1-1 Oho, Tsukuba, Ibaraki, 305-0801

Abstract

Towards Super-KEKB project, we have a plan aiming for upgrade of control system in the KEK Linac. Recently, many embedded Linux-based I/O boards are commercially available. For the control system of the KEK Linac, we are considering the installation of many embedded devices. For this purpose, the timing watchdog system using ARM processor and klystron modulator control system using FPGA have been already developed. Furthermore, we develop the control panel using Tablet PC for klystron modulator control. These control devices are equipped with embedded Linux. In this paper, we report the development of software based on the embedded EPICS in detail.

入射器における組み込み EPICS を利用したソフトウェアの開発

1. はじめに

KEK 入射器 (入射器) では、Super-KEKB 計画に向けた機器の高度化が計画されており、これに伴う制御システムの更新および増強を検討している。現在、入射器制御システムでは、PLC、VME、CAMAC などが多く使用されている。近年、組み込み Linux を搭載した安価な I/O ボードが増えていることから、入射器制御システムにおいても、組み込み機器の導入を検討している。具体的には、ARM プロセッサ (Armadillo-9) を用いたタイミング監視用機器 (TDC: Time-to-Digital Converter)、FPGA (Xilinx: Vertex-4, FX, SUZAKU-V SZ410) を用いたクライストロンモジュレータ制御機器、およびタブレット PC (Android OS: ARM プロセッサ) を用いたクライストロン制御パネルなどを検討している。これらの制御機器は、組み込み Linux を搭載しているため、EPICS ソフトウェアの開発が容易である。本稿では、組み込み EPICS を用いたソフトウェア開発について、詳しく報告する。

2. 組み込み機器

2.1 タイミング監視機器 (TDC)

入射器では、タイミング監視機器の一つとして TDC を導入した^[1]。TDC は、複数のタイミング信号における相対的な時間差を測定するものであり、本機器は約 1 ns の測定精度を有する。

TDC で使用している I/O ボードは、アットマークテクノ社の Armadillo-9^[2]を採用している。Armadillo-9 は、USB2.0、コンパクトフラッシュ、ハードディスク、LAN、および Video 出力など小型ながら豊富な周辺機能を持つ CPU ボードである (図 1)。表 1 に、Armdillo-9 の主な仕様を示す。

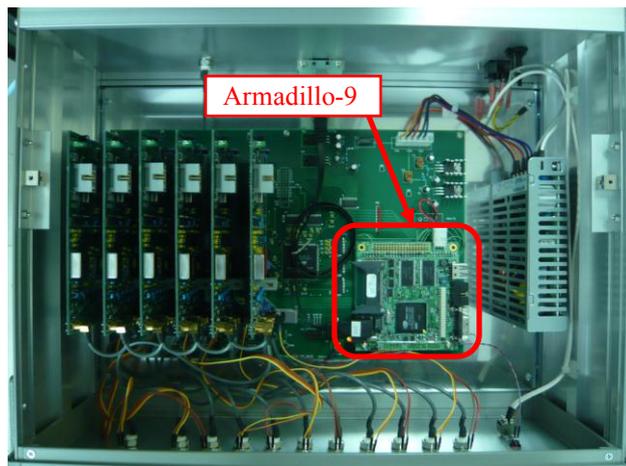


図 1 : TDC の内部

表 1 : Armadillo-9 の仕様

プロセッサ	EP9315
CPU コア	ARM920T
CPU/BUS Clock	200MHz/100MHz
SDRAM	64MB
FLASH	8MB
インタフェース	Ethernet、シリアル、USB など

2.2 クライストロンモジュレータ制御機器

現在のクライストロンモジュレータ制御システムは、CONTROL-1 および CONTROL-2 と呼ばれるモジュール群から構成されている (図 2)。CONTROL-1 は、現場における低電圧・高電圧電源の ON/OFF およびインターロックの表示をおこなう。

[#] skusa@post.kek.jp

CONTROL-2 は、PLC とタッチパネルから構成されており、現場でのステータス表示、自動制御、およびサーバ計算機との通信をおこなっている。両モジュールとも製造から 10 年以上が経過し、PLC の電源部やタッチパネル部の劣化が目立ってきている。このため、CONTROL-1 のダイオードマトリクス、CONTROL-2 の PLC、タッチパネルの代わりに、CPLD、FPGA、およびタブレット PC を用いた新制御システムの開発を検討している (図 3) [3]。

新制御システムは、I/O ボードとして、アットマークテクノ社の SUZAKU-V SZ410[4]を採用した。SZ410 は FPGA に高性能で実績の高い Virtex-4 FX を搭載し、CPU コアとして PowerPC405 を採用した CPU ボードである。表 2 に SZ410 の主な仕様を示す。

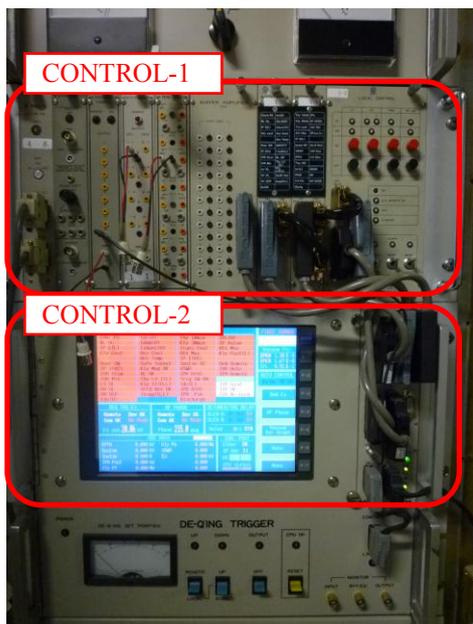


図 2 : クライストロン制御システム

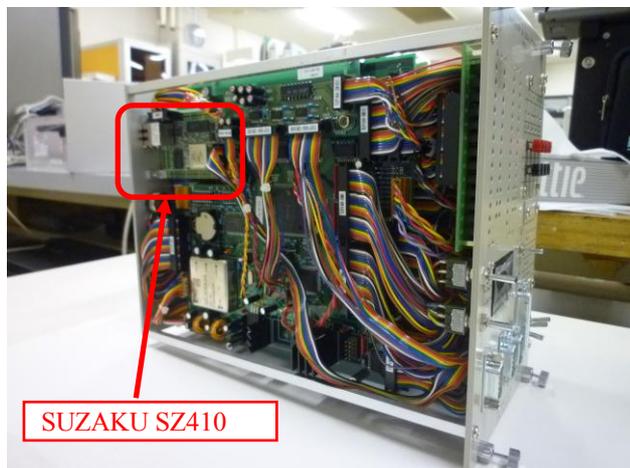


図 3 : 新制御システムのモジュール内部

表 2 : SZ410 の仕様

FPGA	Xilinx Vertex-4 FX
CPU コア	PowerPC405
CPU/BUS Clock	350MHz/87.5MHz
SDRAM	64MB
FLASH	8MB

新制御システムでは、CONTROL-1 および CONTROL-2 の機能が同じモジュール内にあるため、CONTROL-2 に備え付けていたタッチパネルが廃止されることになった。タッチパネルの代わりに現場でステータス表示や制御を可能とするため、持ち運びが容易なタブレット PC の導入を検討している。タブレット PC は、多くのメーカーから種々の OS および機能を搭載した機器が発売されている。本システムには、安価かつ種類の豊富な Android OS のタブレット PC を採用した。Android OS は、Google によりスマートフォンやタブレット PC などの携帯情報端末を主なターゲットとして開発されたプラットフォームであり、Linux をベースとしている。そのため、入射器制御システムとの親和性が高く、EPICS ソフトウェア開発の簡便性も期待できる。

表 3 : タブレット PC の仕様

CPU	NVIDIA Tegra 250
メモリ	512MB
ディスプレイ	10.1 型
FLASH	OS 用 512MB データ用 8GB
OS	Android 2.2

3. 組み込み EPICS ソフトウェア

3.1 開発環境

SUZAKU や Armadillo で動作可能な Linux カーネルのカスタマイズおよびソフトウェア開発をおこなうためには、Linux 計算機上にクロス開発環境を構築する必要がある。これらの開発環境としては、アットマークテクノから、ATDE (Atmark Techno Development Environment) と呼ばれる VMware を用いた仮想 OS 上で動作する開発環境、GNU クロス開発環境パッケージ、および Atmark Dist と呼ばれるカーネルの一部をカスタマイズするためのパッケージが提供されている。ATDE 上での開発は、仮想 OS 上でのソフトウェア開発効率がホスト OS の性能に大きく依存する上、共有ファイルを用いた複数ユーザーでの開発をおこなう場合、ATDE の運用管理が煩雑となる。このため、本開発においては、Linux 計算機上にクロス開発環境および Atmark Dist を用いたクロス開発環境の構築をおこなった。

Android OS 上で動作可能なソフトウェア開発をおこなうためには、Java Development Kit および Android Software Development Kit の開発環境を計算機に構築する必要がある。また、Eclipse^[5]と呼ばれる統合開発環境 (IDE) を導入し、Eclipse プラグインである Android Development Tools Plugin for the Eclipse IDE を利用することで、ソフトウェアの開発、デバッグ、およびパッケージングなどが容易となる。表 3 に開発環境の構成を示す。

表 3 : 開発環境の構成

I/O	Armadillo	SZ410/Android
Linux	Debian4.0	Debian5.0
GCC version	4.1.2	4.1.2

3.2 組み込み EPICS

EPICS IOC ソフトウェアを組み込み機器上で動作させるためには、EPICS Base パッケージをビルドする必要がある。今回の開発では、バージョン R3.14.11 を使用している。Armadillo-9 では、配布パッケージに含まれる標準のコンフィグレーションファイルを一切変更することなくビルドすることが可能であった。しかし SZ410 では、PowerPC 用 Linux のコンフィグレーションについて、いくつか GNU 関連について変更の必要があった。Android においても、ARM 用 Linux コンフィグレーションファイルの GNU 関連の定義変更の必要があった。これらの GNU 関連の定義変更部は、コンパイラ名、GNU のファイルパス、コンパイルオプションであった。

3.3 EPICS 用デバイスドライバの開発

組み込み機器を、EPICS レコードとして外部計算機から参照するためには、EPICS 用のデバイスドライバを開発する必要がある。EPICS 環境を用いた制御システムにおいては、二種類のデバイスドライバが必要となる。一つは、OS・ハードウェア間通信のための Linux kernel デバイスドライバであり、もう一つはデバイス・EPICS 間通信のための EPICS 用デバイスドライバである。Linux kernel 用デバイスドライバに関しては、機器メーカーより提供されているため、今回は、TDC および CONTROL-1 のための EPICS 用デバイスドライバのみ開発した。

3.4 クライストロン制御機器パネルの開発

図 4 は、Android OS 上で動作するクライストロン制御パネルの画面を示している。本ソフトウェアは、クライストロンモジュレータのインターロックやステータス表示を 1 Hz で更新表示する。また、Es および RF 位相の設定、ならびに低電圧および高電圧の ON/OFF 制御が可能である。



図 4 : クライストロン制御パネルの画面

4. 問題点

4.1 EPICS IOC の起動方法

Armadillo-9 および SZ410 では、8MB のフラッシュメモリ内のディスクイメージファイルを読み込むことにより起動する。Linux 起動時には、kernel および userland が SRAM 上に展開される。EPICS IOC などのユーザーが開発したソフトウェアは、userland に書き込む必要がある。しかしながら、機器毎に異なるイメージを作成すると管理が煩雑になるため、共通のイメージを用いた運用を検討している。機器毎の動作の違いは、随時書き換えが可能な config 領域に書かれたパラメータおよび NFS マウントされた共有ファイル内を参照して管理する。しかしながら、現在の起動方法では、ネットワークもしくは NFS サーバが停止した場合には、EPICS IOC を起動することができないことが欠点となる。

4.2 Android OS 上のセキュリティ

Android の linker は、"/system/lib" および "/lib" にあるライブラリを参照する仕組みになっており、セキュリティの関係上、それらのディレクトリは、書き込み禁止になっている。したがって、ユーザーが開発した共有ライブラリを利用するソフトウェアを動作させることができない。このため、EPICS Base パッケージをビルドする際には、静的ライブラリでビルドする必要がある。ただしこの方法では、実行ファイルの容量が大きくなりやすく、ディスク容量が小さい機器への導入には注意が必要である。

4.3 Android と Java

Android 用のソフトウェアの多くは、Java 言語で記述されているが、動作は Dalvik VM と呼ばれる仮想マシン上でおこなわれる。しかしながら、Dalvik VM は Java と完全互換ではないため、EPICS の拡張パッケージである CAJ (Pure Java Channel Access Client) および JCA (Channel Access client for Java using JNI) を動作させることができない。本ソフトウェアでは、EPICS レコードの値を参照する caget コマンドおよび EPICS レコードの値を変更する caput コマンドを外部コマンドとして実行することにより機器の制御をおこなっている。

4.4 Android NDK

Android の開発環境には、Android SDK の一つである Android Native Development Kit (NDK) と呼ばれるパッケージも用意されている。Android 用ソフトウェアは、Dalvik VM 上で動作するが、NDK を用いてソースコードを C/C++ 言語で記述することにより、高速化を図ることが可能である。EPICS Base は C/C++ で記述されていることから、NDK を用いた EPICS Base のビルドを試みた。しかしながら、多くのエラーが出力されたため、現在のところビルドに成功していない。エラーへの対処については、今後の課題である。

5. まとめ

組み込み Linux を搭載した I/O ボードを用いて、EPICS IOC の開発と動作の確認をおこなった。近年、組み込み Linux を搭載した安価な I/O ボードが増えつつあり、今回開発したような組み込み Linux および EPICS を活用することにより、ハードウェアのコストパフォーマンスの改善やソフトウェア開発の大幅な効率化が期待できる。

今後、入射器においては、Super-KEKB 計画に向けた機器の改造、改修が計画されており、これに伴う制御機器の更新および増強が要求されている。組み込み機器の導入も検討されているため、本開発の経験を生かし、さらなる組み込み EPICS IOC の導入を進める予定である。

参考文献

- [1] S.Kusano, et al., “組み込み EPICS を利用したタイミン
グ監視ソフトウェアの開発”, Proceedings 7th Annual
Meeting of Particle Accelerator Society of Japan, Himeji,
Aug. 4-6, 2010
- [2] <http://armadillo.atmark-techno.com/>
- [3] Y.Yano, et al., “クライストロンモジュレータの新制御
システム”, Proceedings of 8th Annual Meeting of Particle
Accelerator Society of Japan, Tsukuba, Aug. 1-3, 2011
- [4] <http://suzaku.atmark-techno.com/index.html>
- [5] <http://www.eclipse.org/>