

YOKOGAWA F3HA12 を使用した高速サンプリングシステムの開発

DEVELOPMENT OF HIGH SPEED SAMPLING SYSTEM USING YOKOGAWA F3HA12

亀田吉郎^{#, A)}, 帯名崇^{B)}

Yoshiro Kameta^{#, A)}, Takashi Obina^{B)}

^{A)} East Japan Institute of Technology Co., Ltd.

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

Abstract

In order to control the accelerator, it is important to remotely monitor various kinds of equipment. For example, a pen recorder, a paperless recorder, etc. are often used to record analog voltage, temperature, etc. Due to the balance between the analog bandwidth and the sampling speed, it is not so easy to achieve contradictory requirements: "to capture fast signal change" and "to save/show averaged data for a long time". Therefore, we have developed a high-speed sampling system with a combination of EPICS and F3HA12 (high speed data collection module) manufactured by Yokogawa. As an initial target of the system development, we selected the sampling speed faster than several hundred Hz, while the statistical data of the specific time window are transferred to the CPU at the rate about 1 Hz. The CPU module, F3RP61, controls the data acquisition module F3HA12 via backplane of the system. We use Linux OS and use the control software framework based on EPICS. On Linux, PREEMPT_RT which is the configuration of the kernel is applied in order to improve the real-time performance, and unnecessary service has been stopped and parameter tuning are performed in order to operate the system stably. A supervisory process was implemented to recover the FIFO error of F3HA12 caused by delays in data transfer. We have developed a high speed sampling system that can measure 12CH data at the rate of 170Hz using F3HA12 / F3RP61. In this presentation, we report details of the equipment and results obtained in the performance test.

1. はじめに

加速器を制御するにあたって様々な機器を遠隔から監視／制御することは重要である。アナログ電圧値や温度等を記録する目的にはペンレコーダやペーパーレスレコーダなどがよく使用されるが、アナログ帯域とサンプリング速度との兼ね合いで“高速な信号変化をとらえる”ことと“平均的なデータを長時間にわたって保存・表示する”ことを両立させることは容易ではなかった。

そこで我々は多チャンネルのペンレコーダの置き換えとしてアナログ電圧値を切れ目なく高速でデータ収集出来るシステムを制作することを目指し、開発目標を以下に定めた。

- 一般的なペンレコーダより高速・広帯域であること：アナログ帯域は数 kHz 以上、サンプリング速度は 100 Hz 程度以上であれば良い。
- データ更新速度は 1 秒程度とし、サンプリング波形の表示をすると同時にその区間内の統計データ(最大・最小等)を Archiver に保存する。
- 長期間のデータを表示可能であること。
- EPICS[1]に対応すること。

サンプリング速度を 100 Hz にすることによってペンレコーダよりも短い間隔の変化をとらえることが可能になる。しかしながら、制御ソフトウェア側は OS を含めたチューニングが必須になり、本開発の大きなテーマになっている。また、EPICS に対応することで、CSS Archiver による長時間のデータ保存、参照が可能になる。これらを踏まえたシステムの概要を Figure 1 に示す。

本発表では装置に関する詳細および性能試験で得られた結果について報告する。

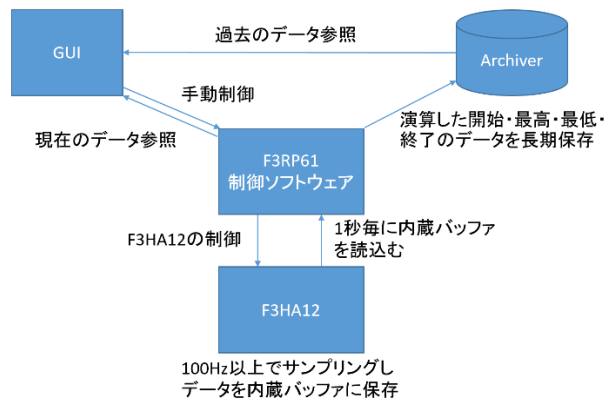


Figure 1: System overview.

2. ハードウェア構成

ハードウェア構成は YOKOGAWA 製 F3HA12(高速データ収集モジュール)[2]を YOKOGAWA 製 F3RP61(Linux 対応 CPU モジュール)[3]で制御する方法を採用した。F3RP61 は KEK における多くの加速器で運用されており実績も多く信頼性も高い。また、高速・高分解能でアナログ入力も 12 チャンネルあることから採用することにした。また、制御ソフトウェアを開発する上で F3RP61 Device/Driver Support[4]を使用することが可能であり開発コストを抑えられることも大きな要因である。F3HA12 の主な仕様を Table 1 に示す。

[#] hig-kame@post.kek.jp

Table 1: Main Specifications of F3HA12

Resolution	16 bit ADC
Number of input channels	12 Channel
Sampling rate	Max 200 kSPS
Cutoff Frequency	40 kHz

F3RP61 と F3HA12 はベースモジュール[5]と呼ばれるバックレーンに接続することでモジュール間の通信が行われる。このベースモジュールは 5/9/13/16 スロット用が用意されており柔軟なシステム構築が可能である。F3RP61 と F3HA12 がベースモジュールにセットされた状態を Figure 2 に示す。

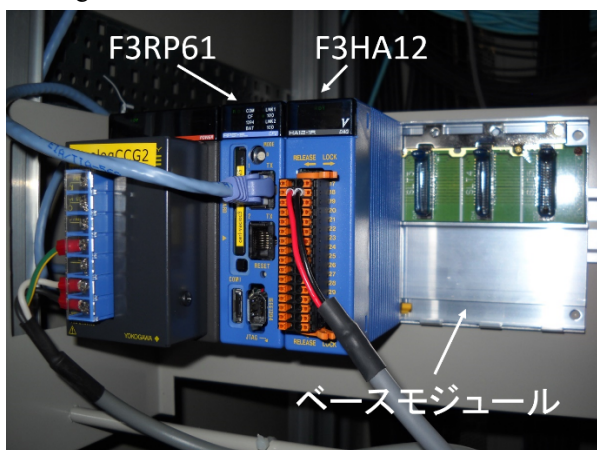


Figure 2: Photograph of F3RP61 and F3HA12 connected to the base module.

3. F3HA12 の制御

3.1 収集動作モードとパラメータ設計

F3HA12 には 3 つの収集動作モードがある。

- シングル(運転指示毎に実行する収集モード)
- レペティティブ(データ収集後、再度トリガ条件が成立した場合に、データ収集を繰り返し実行する収集モード)
- コンティニューアス(データ収集後、継続してデータ収集を実行する収集モード)

今回はトリガに関係なく継続してデータ収集を行うことが目的なのでコンティニューアスモードで動作させることにした。

収集動作モードによってサンプリングを保存する内蔵バッファサイズが異なってくる。シングルでは 2 M ワード、レペティティブでは 1 M ワード、コンティニューアスでは 2048 ワードである(内部では 4096 ワードになっており、ソフトウェア側が読み込みを行いながらデータ収集を継続して行うダブルバッファ構造となっている)。コンティニューアスモードで収集対象を 12 チャンネルとした場合、収集点数は最大 170 点まで指定できることになる。

サンプリングレートについては動作モードを内部の定周期タイマ(5 μ s)をベースとし、この分周比を指定することで指定できる。例えば 100 Hz とするならば収集周期を 2000 とすれば良い。

100 Hz で収集点数を 170 とした場合、内蔵バッファが満杯になる時間は 1.7 sec となる。制御ソフトウェア側ではこれより早い周期で読み込めば良い。サンプリングレートを上げると内蔵バッファが満杯になる時間が短くなるので制御ソフトウェア側でどれだけ短い間隔で読み出すことができるか把握することが重要になる。

3.2 設定、操作するレジスタとリレー

F3HA12 では設定や操作の手続きに入力リレーと出力リレーを用いる。継続して出力リレーを操作する場合は、必ず出力リレーの操作に対する入力リレーの応答を確認してから、次の出力リレーの操作を行う必要がある。今回使用した主な入力リレー、出力リレーを Table 2 に示す。

Table 2: Main Input Relay and Output Relay of F3HA12

X05	収集データ書込み完了応答
X17	自己診断 NG
Y34	運転/停止
Y35	強制収集開始
Y38	収取データ読み出し完了通知
Y49	自己診断 NG 了解

モジュールの動作にかかわる設定にはモードレジスタを使用し、運転中の操作とデータの読出しには入出力データレジスタを使用する。これらのレジスタは、いくつかの例外を除いてデータ長が 16 bit のレジスタになっている。その読出しと書込みには、モードレジスタ、入出力データレジスタいずれも READ 命令および WRITE 命令を使用する。カウンタ機能や、収集に係る設定用レジスタの一部には、データ長 32 bit のレジスタがあるが、これらについても READ 命令と WRITE 命令を使用する。同様に今回使用した入出力データレジスタ、モードレジスタを Table 3 に示す。

Table 3: Main Input Output Data Register and Mode Register of F3HA12

17~2064	収集データ 1~2048
3041	収集周期
3042	収集動作モード
3043	収集対象選択
3045	収集点数指定 L
3046	収集点数指定 H

3.3 測定開始までの手順

F3HA12 の電源投入後、連続測定状態にするには一連の手続きでモジュールの設定を行う必要がある。また、制御ソフトウェアの再起動時も考慮して起動シーケンスを設計する必要がある。起動シーケンスは大きく 3 つに分かれて順にリセット、パラメータの設定、運転開始になっている。リセットではデータ収集中の制御ソフトウェアの再起動や本体の自己診断 NG による動作停止を考慮

し、リカバリを含めたリセットを行う。パラメータの設定では、各種パラメータを設定したのち、パラメータ設定要求を ON にし X01,X02 で応答を確認する必要がある。データ収集開始時には運転状態にしたのちに強制収集開始を ON/OFF することによってモジュールの収集動作が開始される。また、後述するが FIFO エラー時に制御ソフトウェア側で自動的にモジュールをリカバリするかどうかの設定を行う。測定開始までの手順を Figure 3 に示す。

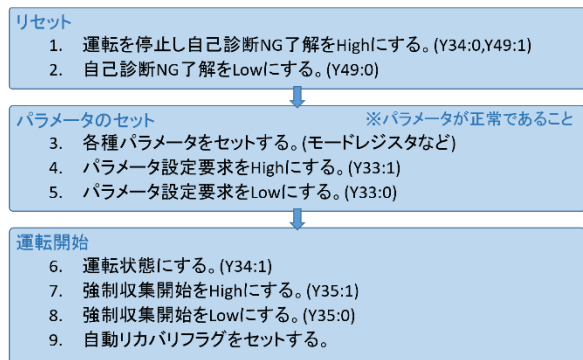


Figure 3: Procedure to start measurement.

3.4 測定データの読み込みシーケンス

F3HA12 は運転状態設定することで自動的に測定を開始する。内臓バッファが満杯になると入力リレーの X4(収集完了)が ON になり、入出力データレジスタ 17~2064(収集データ)に書き込まれると入力リレーの X5(収集データ書き込み完了応答)が ON になる。制御ソフトウェアはこれをトリガに入出力データレジスタ 17~2064 の 2048 ワードを読み込み、完了した後に出力リレーの Y38(収集データ読出し完了通知)を ON/OFF にする必要がある。F3HA12 は制御ソフトウェアが読み込み中でもダブルバッファで異なる領域に測定データを保存していく。この時に制御ソフトウェアが読み込みを完了する前に収集データが満杯になると FIFO エラーとなってしまい、入力リレーの X17(自己診断 NG)が ON となってデータ収集が中断する。制御ソフトウェアは安定して高速に一連の動作を行う必要がある。測定データの読み込みシーケンスを Figure 4 に示す。

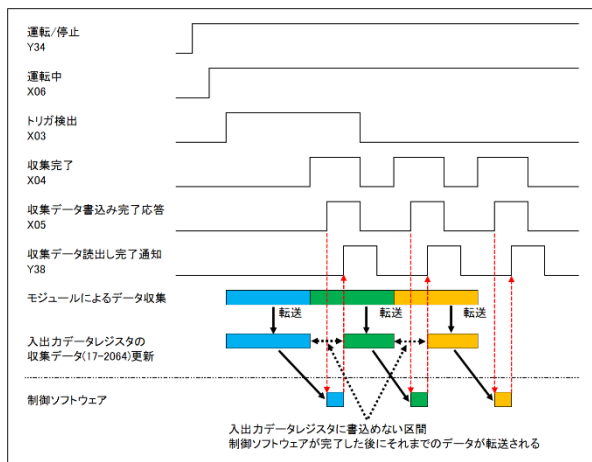


Figure 4: Measurement data reading sequence.

3.5 FIFO エラー時の復帰シーケンス

前述したように FIFO エラーが発生した場合は運転停止状態に戻ってしまう。そこで自動でリカバリする機能を制御ソフトウェア側で実装した。この機能は任意で ON/OFF 出来るようにしている。具体的な動作としては入力リレーの X17(自己診断 NG)をモニタし、ON になった場合に運転を停止し出力リレーの Y49(自己診断 NG 了解)を ON/OFF にする。これで FIFO エラーがリセットされる。その後、運転再開のために運転状態にし、出力リレーの Y35(強制収集開始)を ON/OFF にすることで連続運転を再開することが可能になる。FIFO エラー時の復帰シーケンスを Figure 5 に示す。

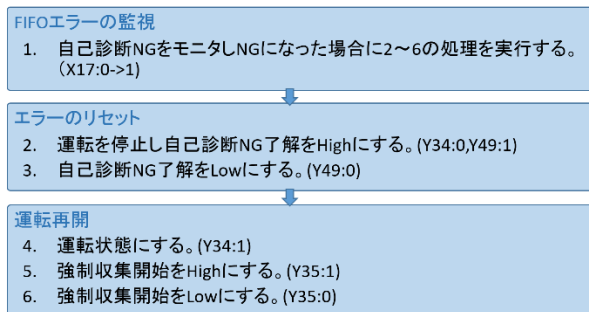


Figure 5: FIFO error recovery sequence.

4. リアルタイム性の向上

F3RP61 は OS として Linux を使用し EPICS を用いて制御ソフトウェアを動作させている。Linux においてリアルタイム性を向上させるために Kernel の configuration である PREEMPT_RT を適用する必要がある。EPICS についてもプロセスの優先度を変更するために configuration の変更を行い、Linux システムにおいても安定して動作させるために不要なサービス停止やパラメータの調整を行う必要がある。

Kernel については、リアルタイム性を向上させるために PREEMPT_RT を適用したものを新たに作成した。これは KEK の加速器で使用している F3RP61 の標準環境には含まれていないため、新たに作成する必要があった。PREEMPT_RT を適用するには GUI を使った menuconfig から configuration を変更することが可能である。また、ハードウェアクロックの設定も無効にしている。これは、Kernel の安定性を高めることと時刻の同期はソフトウェアクロックで行う方針であるため不要になるからである。configuration については menuconfig から変更することが出来ないため直接 Kconfig の CONFIG_GENERIC_CMOS_UPDATE を修正することで対応している。

EPICS については、SCHED_FIFO リアルタイムポリシーを使用するため、CONFIG_SITE の USE_POSIX_THREAD_PRIORITY_SCHEDULING[6] を有効にして EPICS base を再構築している。制御ソフトウェアを起動する際にもスケジューリングポリシーに FIFO を指定している。

Linux システムについては、Kernel に PREEMPT_RT が適用されたことによりデバイスドライバである m3iodrv.ko

を m3iodrvRT.ko に変更する必要がある。これは YOKOGAWA で提供している Linux BSP に含まれているものを使用している。

次に crond の停止である。これは不要なサービスを停止することで、リアルタイム性を低下させる要因である不要な I/O アクセスと CPU 負荷を低減している。標準で動作している機能を確認しても不要なものを除くと logrotate がある。1 日当たりのデータ増加量を考慮してもわずかなので停止している。

最後に ntpd の時刻補正方法である。通常は step モードによる動作になっているが、slew モードに変更することによって時刻が大きく補正されることを避けるためである。

5. 性能評価

性能評価を効率的に行うために、F3HA12 のコントロールパネルと演算した値をプロットする GUI を CSS[7] で作成した。これを Figure 6 に示す。

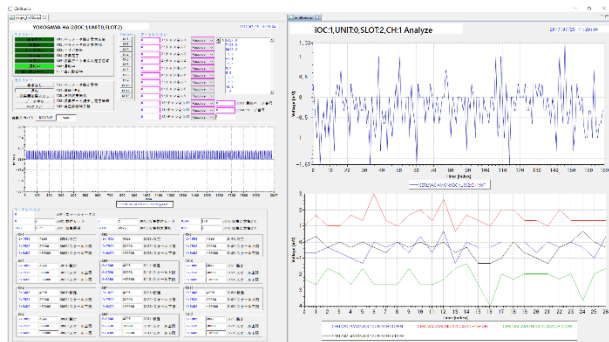


Figure 6: Control panel of F3HA12.

次に、Linux システムのリアルタイム性の安定度を見る。方法としては F3HA12 に変えて YOKOGAWA 製 F3XD32(DC 入力モジュール)[8]を使い 200 Hz の矩形信号を入力し信号の立ち上がりの間隔を見ることにする。F3XS32 に変える理由としてはハードウェア割込みで値の変化を見ることが可能で、制御ソフトウェアの性能にあまり左右されずに評価できるからである。また、制御ソフトウェアも信号の読み込みだけの簡略したものに変えることで問題の切り分けを容易にするためである。周期実行の安定度を Figure 7 に示す。

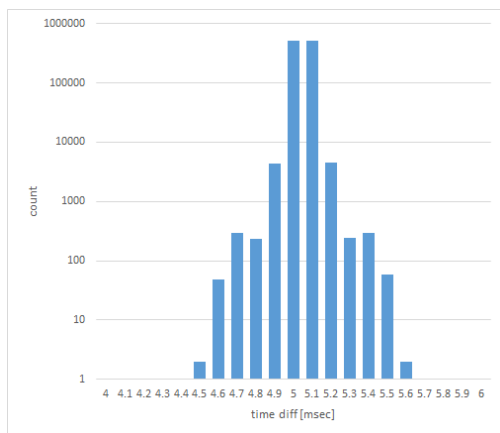
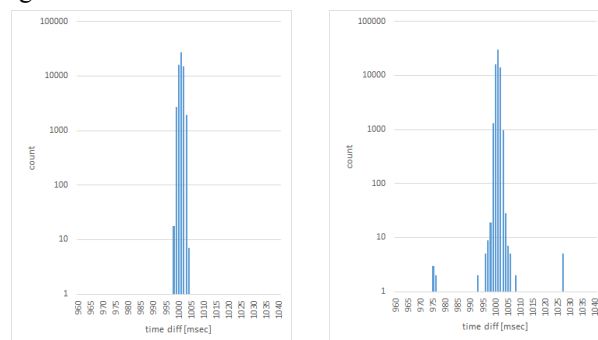


Figure 7: Periodic-scan stability of linux system.

このデータより±1ms 内でモニタ出来ており Linux システムは安定して動作していることがわかる。

次に、リアルタイム環境と非リアルタイム環境で同じ制御ソフトウェアを動作させ安定度を比較する。設定は目標とした“サンプリング速度は 100 Hz 以上で 1 Hz 程度の一定周期間隔でその区間の開始・最高・最低・最終データを転送するシステム”なので収集点数は 12 CH の限界である 170 点とし、収集周期は 1 秒で内蔵バッファが満杯となる 1177 とした。これは約 170 Hz でサンプリングする計算となる。環境毎の制御ソフトウェアの安定度を Figure 8 に示す。



(a) Real-time environment

(b) Non real-time environment

Figure 8: Comparison of stability for each environment.

リアルタイム環境を適用したものは、非リアルタイム環境と比較してばらつきが少なく動作していることがわかる。また、リアルタイム環境を適用したものであっても前述の F3XD32 の安定度と比較してばらつきが大きくなっている。これは F3HA12 ではハードウェア割込みが使用できず、ポーリングを使ってモニタする以外に方法がないためである。しかしながら概ね±3 ms 以内で動作していることから妥当な結果といえる。課題は残るが当初の目標は達成することができた。

6. まとめと今後の予定

目標とする“サンプリング速度は 100 Hz 以上で 1 Hz 程度の一定周期間隔でその区間の開始・最高・最低・最終データを転送するシステム”を構築することができた。cERL ではこの装置を用いて一部の真空度をモニタするシステムを構築し、安定して動作している状況である。

今後の予定だが、モジュール構成についてはハードウェア割込みによって値の変化を見ることが可能な F3XD32 と F3HA12 を組み合わせて実現可能か検討する。ソフトウェアについては更なる高速化やモジュールを増やしての多チャンネル化に対応しつつ、安定して動作するように対応していく。また、一定周期間隔でその区間の開始・最高・最低・最終データを演算していたが、最高・最低・平均・分散に変更する予定である。これは開始・終了はほぼ同じデータになってしまうからである。ハードウェアについては Linux 対応 CPU モジュールである F3RP61 が生産終了になったので、後継機種である F3RP71 に対応することが必須である。

参考文献

- [1] EPCIS;
<http://www.aps.anl.gov/epics/>
- [2] Yokogawa Electric Corporation 高速データ収集モジュール;
<https://www.yokogawa.co.jp/itc/Products/ADDA/itc-prdad15-ja.htm>
- [3] Yokogawa Electric Corporation Linux 対応 CPU モジュール;
<https://www.yokogawa.co.jp/itc/Products/ADDA/itc-prdad15-ja.htm>
- [4] EPICS Device and Driver Support for Yokogawa's F3RP61;
<http://www-linac.kek.jp/cont/epics/f3rp61/>
- [5] Yokogawa Electric Corporation RTOS コントローラ ベースモジュール;
<http://www.yokogawa.co.jp/rtos/Products/Base/rtos-prdbase1-ja.htm>
- [6] https://wiki-ext.aps.anl.gov/epics/index.php/How_To_Use_Posix_Threading_Priority_Scheduling_under_Linux
- [7] CSS (Control System Studio);
<http://controlsystemstudio.org/>
- [8] Yokogawa Electric Corporation 入出力モジュール;
<https://www.yokogawa.co.jp/itc/Products/DIO/itc-prddio1-ja.htm>