F3RP61 での STARS の利用とビームライン制御への応用

小菅隆

高エネルギー加速器研究機構 物質構造科学研究所

1. 概要

これまで、Linux を OS として採用した横河電機株式会社製 PLC（Programmable Logic Controller）用 CPU モジュール F3RP61 上で、高エネルギー加速器研究機構 放射光科学研究施設（KEK-PF）においてビームライン制御等に使用されているソフトウエア「STARS」(Simple Transmission and Retrieval System)[1,2]の動作確認[3]を行ってきた。

F3RP61 の利用は、ビームライン制御についても有用であり、今回、ビームラインへの応用を目指して C 言語による実践に向けた STARS クライアントの開発を行った。今後、本クライアントは低速陽電子ビームラインの制御システム等へ応用される予定である。ここでは F3RP61 上での STARS の利用と、今回開発を行った C 言語によるクライアントプログラムの詳細および、実践導入の予定について報告する。

2. STARS

STARS は比較的小規模な制御システム用に設計された非常にシンプルかつ柔軟なソフトウエアで、放射光ビームラインの制御をはじめ様々なシステムへ導入が進行中である。KEK-PF では表 1 に示すビームラインおよび様々なシステムで利用されている。また、KEK-PF 以外でも山口大学[4,5]や核融合科学研究所[6]をはじめとする機関において利用されており、名古屋大学シンクロトロンのビームライン制御システムへの採用も決定している。

表 1　KEK-PF において STARS が導入されているビームラインまたはシステム

カテゴリー	ビームラインまたはシステム名
PF-2.5GeV Ring X 線ビームライン	BL-1A、BL-3A、BL-4B、BL-4C、BL-5A、BL-6A、BL-6C、BL-7C、BL-8A、BL-8B、BL-9A、BL-9C、BL-12C、BL-14A、BL-17A、BL-18B
PF-2.5GeV Ring VUV ビームライン	BL-2A、BL-11B、BL-11D、BL-13A、BL-16A、BL-19A、BL-20A
PF-AR ビームライン	NE-1A、NE-3A、NW-2A、NW-10A、NW-12A、NW-14A
その他のビームライン	低速陽電子ビームライン
その他のシステム	ビームラインインターロック集中管理システム、 PF 実験ホール入退室管理システム、キー管理システム、 ミラー評価システム等

STARS は STARS サーバと複数の STARS クライアントから構成され、それぞれの STARS クライアントは TCP/IP ソケットを使用して STARS サーバに接続する。接続の際にはある程度のセキュリティーを保持するために、簡単なキーワードチェックのプロセスが設けられているが、基本的にそれぞれのクライアントおよびサーバの通信は、すべてテキストベースのメッセージを用いて行われるため、STARS クライアントの開発は TCP/IP ソケットおよびテキスト処理を行うスキルさえあれば容易に行う事が出来る。また、更にプログラミングを容易にするために Perl、C、Visual Basic、C#などのライブラリが用意されている。なお、STARS サーバ自体は Perl を使用して作成されており、様々な OS 上で動作可能である。

図 1 は STARS のサーバおよびクライアントの構成例である。それぞれの STARS クライアントはそれぞれユニークなノード名を持ち、たとえば「Term1」のノード名を持った STARS クライアントが「Dev1」のノード名を持った STARS クライアントを宛先としたメッセージ「Dev1 xxx」のようなテキストを STARS サーバに送信すると、STARS サーバはこのメッセージを「Dev1」に送信する。STARS では基本的にこのようなメッセージのやり取りを、それぞれの STARS クライアントが STARS サーバを介して行う事で制御システム等を構築してゆく。

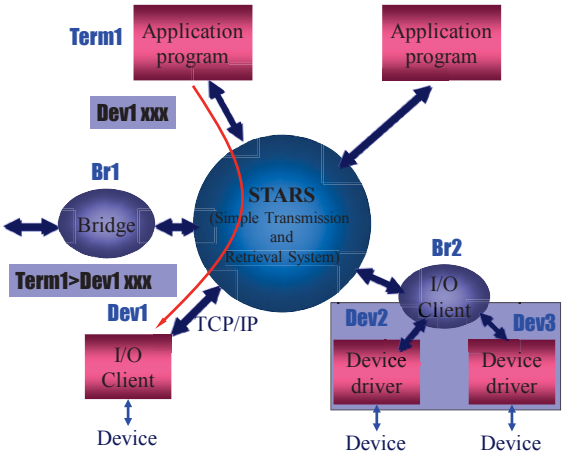

図 1　STARS におけるメッセージの送受

STARS においてはグラフィカルユーザインターフェース（GUI）プログラムも、実際にハードウエアを制御するためのデバイスドライバ的なプログラムも基本的には STARS クライアントであり、それぞれ「User Client」、「IO Client」と呼ばれている。今回のプログラム開発は、F3RP61 用にこのうちの「IO Client」の作成である。なお、これらのクライアントは、STARS の動作中いつでも接続や切断が行えるので、機能拡張や変更はシステム全体を停止することなく行う事が可能であり、今回のプログラム開発においても STARS サーバおよび TEST 用 GUI を停止することなく F3RP61 用 IO Client のテスト及び修正を行う事が出来た。

3.　F3RP61

F3RP61 は横河電気株式会社製 PLC である FA-M3 用の e-RT3 2.0 Linux に対応した CPU モジュールであり、高エネルギー加速器研究機構では加速器の制御等に利用されている。実際には加速器等用の制御システムである EPICS（Experimental Physics and Industrial Control System）の IOC（Input Output Controller）等として応用[7,8]されており、導入および動作実績は様々な場で報告されている。

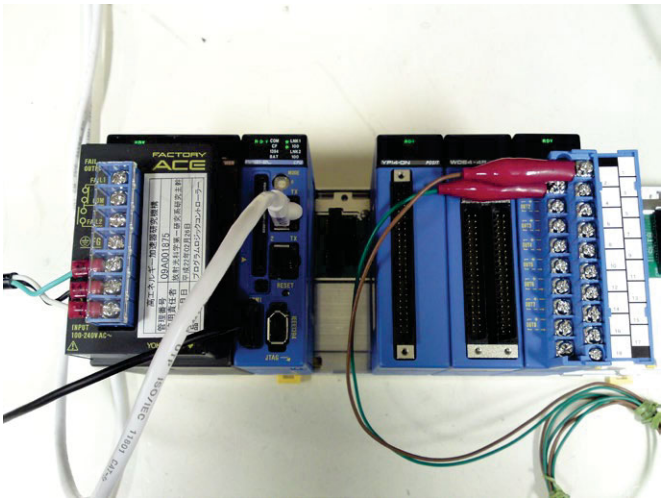

写真 1　FA-M3 および F3RP61

FA-M3 および F3RP61 の利用は加速器だけではなく、ビームライン制御に於いても有用であり、放射光科学研究施設のビームライン制御システムとして広く利用されている STARS の動作テストが進められてきた。

STARS においては F3RP61 上で Perl の利用可能であることから、STARS サーバおよび Perl による STARS クライアントを動作させる事が可能であり、この STARS サーバおよび Perl による STARS クライアントについては動作確認が既に行われている。また、C 言語を用いた簡単なテストプログラムを作成し、FA-M3 上のパラレル入出力モジュールに STARS からアクセス出来ることも確認された。今回開発を行った C 言語による F3RP61 用 STARS クライアントは実践導入に向けたプログラムである。なお、今回開発に使用した FA-M3 および F3RP61 を写真 1 に示す。

4.　C 言語による STARS クライアントの開発

STARS クライアントの開発は前述の通り TCP/IP ソケットの利用およびテキストの操作が可能であれば容易であるが、接続時のセキュリティー確保のためのキーワードチェックのプロセス等を複数のキーワードを用いて行おうとすると若

干複雑となる。このため STARS では予めこれらのプロセスを自動化するためのライブラリをいくつかのプログラミング言語用に用意している。また、これらのライブラリを利用すると TCP/IP ソケットについても、特に意識する必要がなくなるので更にプログラミングが容易となる。なお、F3RP61 から FA-M3 上の各モジュールにアクセスするための C 言語用のライブラリが横河電気株式会社から提供されているので、今回は C 言語を利用した STARS クライアントの開発を行う事とした。実際の F3RP61 のプログラム開発は F3RP61 とは別に RPM パッケージを利用可能な Linux を使用したクロス開発環境が必要であり、コンパイル等の作業はこの RPM パッケージを利用可能な Linux マシン上で行う事になる。

STARS C ライブラリを使用したプログラム開発

STARS C ライブラリを利用する際にはヘッダファイル starsif.h および starsfunc.h をインクルード（図 2: 17 行および 18 行目）する必要がある。また、キーワードチェックの際に使用されるキーワードを列挙した「キーワードファイル」を予め準備しておく必要がある。次に STARS サーバの動作するホスト名や自ノード名等を予めセットし、stars_open 関数を

```
14 #include<sys/types.h>
15 #include<sys/stat.h>
16 #include<fcntl.h>
17 #include<starsif.h>
18 #include<starsfnc.h>
19 #include<asm/fam3rtos/m3iodrv.h>
20 #include<ert3io.h>
21
22
23 #define IODEV           "/dev/m3io"
24 #define MY_NODE_NAME    "ert3io"
25 #define STARS_SERVER    "localhost"
26 #define STARS_PORT      6057
27
28
29 int sthandler(STARS_CONNECTION *scp);
30 int get_node_id(char *node_name);  /* 100 main, 201-216 I/O dev
31 int get_in_value(char* rtbuf, int slot);
32 ERT3STATUS get_ch_number(int* chnum, ERT3_DEVICE* ert3, char* n
33
34 int fdio;                          /* File handle of IO devi
35 char my_node_name[STARS_SVRN_MAX];  /* Node name of this clie
36 ERT3_DEVICE* ert3_iodev[17];       /* Inforamtion of IO devi
37 char child_node_list[256];         /* Node list */
38
39
```

図 2　starsif.h および starsfunc.h のインクルード

```
94    (int)ert3_iodev[lp]->devtype,
95    minfo.unitno, minfo.slotno, minfo.msize,
96    minfo.num_xreg, minfo.num_yreg, minfo.num_dreg);
97 #endif
98    }
99    }else{
100 #ifdef DEBUG
101    printf("device not found.¥n");
102 #endif
103    ert3_iodev[lp] = NULL;
104    }
105 }
106
107
108
109 /* Open connection of STARS. "satars_open" tries connection first then allocates
110 memories for structure of STARS_CONNECTION. */
111 if((svr1 = stars_open(my_node_name, stars_server, stars_port, key_file))
112   == NULL){exit(-1);}
113
114 /* "sthandler" will be called when this client receives a message from STARS ser
115 svr1->cbfunc = sthandler;
116 stars_add_callback(svr1);
117
118
119 /* Wait for input */
120 while(stars_mainloop(50) >= 0){
121   for(lp = 1; lp <= 16; lp++){
122     if(ert3_iodev[lp] == NULL){continue;}
123     if(ert3_iodev[lp]->interval == NULL){continue;}
124     rtst = (*ert3_iodev[lp]->interval)(ert3_iodev[lp]);
125     if(rtst == ERT3STATUS_EVENTDETECTED){
```

図 3　stars_open 関数

呼び出すと stars_open 関数（図 3: 111 行目）は自動的に STARS サーバへの接続、キーワードのチェックのプロセスを行った後、STARS_CONNECTION 構造体用のメモリを確保した上でそのポインターを返す。STARS サーバへのメッセージの送信および受信は STARS_CONNECTION 構造体と stars_send 関数、stars_receive 関数などを使用して行う。なお、STARS C ライブラリには（Windows 上では動作しないが）、コールバックの機能が設けられている。予め呼び出される関数を設定しておけば、STARS サーバがメッセージを送ってきた際に自動的にメッセージを読み込んだ上で設定した関数が呼び出される。このコールバック機能は今回のような IO Client を開発する際には非常に有効であり、今回のプログラムでは STARS サーバから送られてくるコマンドをもとに必要な関数を呼び出し、ioctl 関数を使用して FA-M3 上のモジュールにアクセスしている。

FA-M3 上のモジュールへのアクセス

F3RP61 から FA-M3 上のモジュールへのアクセスは ioctl 関数（図 4: 382 行目）を利用して行う。FA-M3 上のモジュールへのアクセスのためには「m3iodrv.h」をインクルードする必要があり、ioctl 関数を使用するために必要なヘッダファイルも同様にインクルードしておく必要がある。なお、今回の開発ではこれまでにテスト的に使用したパラレル入出力のユニット WD64-4P 用のプログラムをも

```
H3 ERT3STATUS DA08_GetRegister(unsigned short* rddata, int fd, int slotno, int ch){
374   M3IO_ACCESS_REG rly;
375
376   rly.unitno = 0;
377   rly.slotno = slotno;
378   rly.start  = ch;
379   rly.count  = 1;
380   rly.u.pwdata = rddata;
381
382   if(ioctl(fd, M3IO_READ_REG, &rly) < 0){
383     return(ERT3STATUS_IOERROR);
384   }
385 #ifdef DEBUG
386 printf("Slot = %d, Ch = %d, Data = %u¥n", rly.slotno, rly.start, *rly.u.pwdata);
387 #endif
388
389   return(ERT3STATUS_PROCESSED);
390
391 }
```

図 4　ioctl 関数によるモジュールへのアクセス

とに、プログラム全体の見直しおよび DA コンバータ DA08-5X 用のプログラム部分の新たな開発を行った。

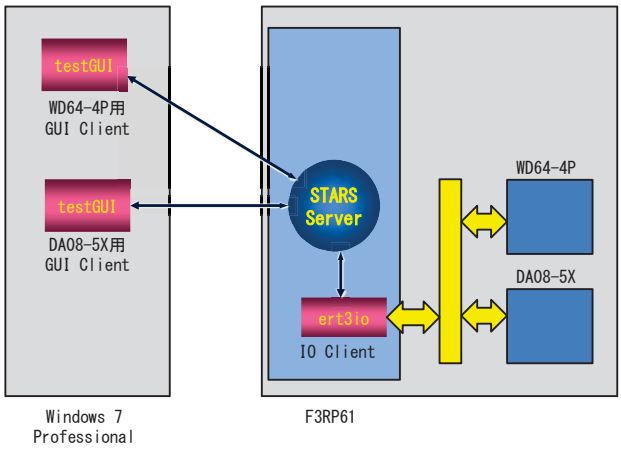

図 5　テスト時のソフトウエア構成

5.　F3RP61 上での動作確認

今回開発した IO Client のテストを行った構成を図 5 に示す。今回のテスト環境では、以前にテストを行った際に作成したユーザクライアントである WD64-4P 操作用 GUI も同時に接続している。F3RP61 用 IO Client 自体は以前のものとは違うが、ノード名やチャンネル構成は同じとしてあるので、WD64-4P 操作用 GUI には一切手を加える事無く使用可能となっている。更に、DA08-5X 用のテスト用 GUI プログラムを作成し、それぞれのプログラムから WD64-4P および DA08-5X に正常にアクセスできることを確認した。なお、WD64-4P 操作用 GUI および DA08-5X 用のテスト用 GUI プログラムのスナップショットを図 6 に、モジュールへの書き込みが正常に行われている事を確認している様子を写真 2 に示す。

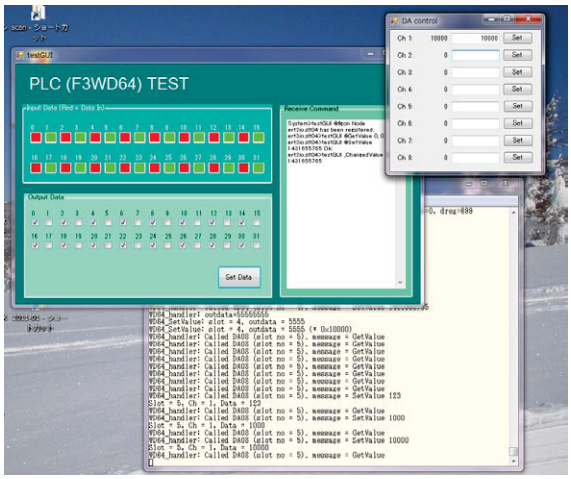

図 6　テスト用 GUI プログラム

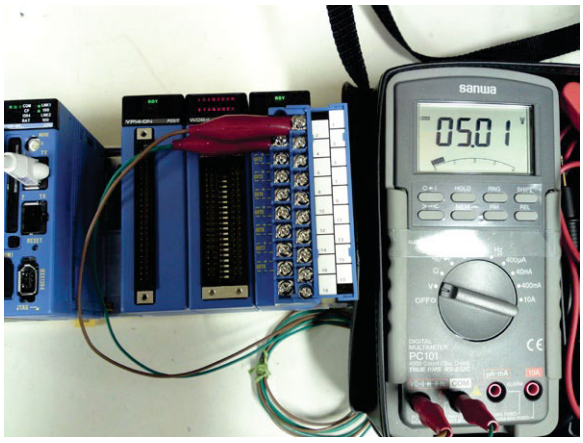

写真 2　パラレル出力および DA 出力の確認

6.　低速陽電子ビームラインへの導入予定

今回開発を行った IO Client は KEK-PF での低速陽電子ビームラインにおいて使用する予定である。これまで低速陽電子ビームラインで利用されていた STARS に関しては再度構成について見直しを行う。なお、DA08-5X は電源の制御用として使用する予定であるが、対象となる電源が老朽化している等の問題があり、導入後に電源の機種変更が行われる可能性が高い。STARS の利用は、その際にもソフトウエアの改造を最小限に抑えることができるなど、大きなメリットが期待できる。

写真 3　今回制御予定の電源

7. まとめ

これまでの動作確認を踏まえて、今回実践投入を目指した F3RP61 用 STARS IO Client の開発に成功した。また、動作テストでは十分な結果を得ることができた。実際の低速陽電子ビームラインへの導入は対象となる電源の老朽化等の問題により若干遅れているが、今後随時導入作業を進めてゆく予定である。

8. 参考文献等

[1] T. Kosuge, Y. Saito, “RECENT PROGRESS OF STARS”, Proceedings of PCaPAC2005, Hayama, Japan, 2005.

[2] http://stars.kek.jp

[3] T. Kosuge, K. Nigorikawa, “STARS ON PLC”, Proceedings of PCaPAC2010, Saskatoon, Canada, 2010.

[4] 田内 康, “STARS による Web カメラ画像の再送”, 平成 20 年度九州工業大学第 4 回情報技術研究会

[5] 田内 康, “STARS を用いた環境データ計測システムの構築”, 平成 21 年度京都大学総合技術研究会

[6] 小川 英樹, et al, “LHD における 20 バレル固体水素ペレット入射装置の制御開発”, 平成 22 年度高エネルギー加速器研究機構技術研究会

[7] J. Odagiri, et al., “APPLICATION OF EPICS ON F3RP61 TO ACCELERATOR CONTROL", Proceedings of ICALEPCS2009, Kobe, Japan.

[8] J.-I. Odagiri, et al., “Development of Image Processing System on Embedded EPICS for Beam Diagnostics”, Proceedings of PCaPAC2010, Saskatoon, Canada, 2010.