Linux PLC のための汎用的な Equipment Manager の開発

DEVELOPMENT OF GENERAL-PURPOSE EQUIPMENT MANAGER FOR LINUX PLC

植田倉六#, 増田 剛正
Souroku Ueda #, Takemasa Masuda
Japan Synchrotron Radiation Research Institute (JASRI/SPring-8)

Abstract

Equipment Manager (EM) is one of the software components of MADOCA control framework used at SPring-8 and controls equipment such as magnet power supplies, ion pumps, beam position monitors and so on. Since EM accesses equipment directly, we have to customize the EM software according to the equipment. New source code is written to create the new EM whenever a new kind of equipment is installed. If general-purpose EM that can be applied to a wide variety of equipment is prepared, we can start control of newly installed equipment immediately. As the first target, we have newly developed general-purpose EM for the Linux PLC e-RT3 by YOKOGAWA Electric Corporation, which is started introducing to the SPring-8 control system. In order to generalize control functions using in EM, a lot of information required for customization according to the equipment is listed in a configuration file (config.tbl) and passed to the EM. As the result it becomes difficult for us to grasp the configuration file. We have therefore prepared the Web-base user interface not only to understand the configuration file easily but also to edit the configuration file. The use of the e-RT3 chooses control commands, module to be controlled and functions to be used, and creates and updates the config.tbl fie by using the Web interface.

1. はじめに

SPring-8 制御系フレームワークである MADOCA[1] を構成するソフトウェア群の１つに、電磁石電源、イオンポンプ、ビーム位置モニター等の機器を制御するための Equipment Manager(EM)がある。EM は機器に直接アクセスする性質上、他のソフトウェアと異なり機器毎のカスタマイズが必要で、新しい種類の機器が追加されると、新しい EM の作成が必要になる。これまでは、その都度新しいソースコードを記述しソフトウェアを作成してきた。

SPring-8 制御系に導入が始まっている横河電機製 Linux 搭載 PLC モジュール(e-RT3) [2]では、各モジュールへのアクセス方法がデバイスドライバレベルで共通化されており、その機能を使用すれば制御に使用するモジュールに依存しない汎用的な EM の作成が可能である。汎用的な EM を作成すれば e-RT3 に新たな種類のモジュールが追加されても、EM を作成することなく、利用者がすぐに制御することが可能となる。そこで、今回 e-RT3 をターゲットに汎用的な EM の作成を行った。

2. EM

2.1 処理関数

EM は C/C++言語で実装されている。EM は機器の制御を行うにあたり主に３つの処理を介して機器の制御を行っている。それは変換処理・機器制御処理・結果処理である。変換処理は利用者が入力した物理データを、機器を動かすために必要なバイナリデータに変更するための処理で、例えば倍精度のアナログ値を整数型バイナリ値に変換する処理が行われる。機器制御処理は、ioctl 関数を使用しデバイスドライバ経由でモジュールにアクセスし、データ取得や設定を行うための処理である。結果処理は、機器制御処理で設定・取得した値を利用者に返信するための処理である。EM を作成するに当たり、主に必要となるのが機器制御処理関数の作成である。

2.2 EM 設定ファイル

MADOCA はメッセージ駆動型のフレームワークであるため、EM は、受け取ったメッセージ(SVOC コマンド)と３つの処理関数との対応付けが必要になる。それを行っているのが設定ファイル(config.tbl)である。また config.tbl には、SVOC コマンドと処理関数の対応付け以外に、関数に渡すパラメータ(引数)も設定される。config.tbl の一例を Figure 1 に示す。

```
get/srvacf1wcnt-c21_slot6
 use_mode1      em_cnt1_ert3_io_read_reg /dev/m3io 0 6 16 83 1
                none
                em_cnt1_ert3_io_ret_status
 use_mode2      em_cnt1_ert3_io_read_reg /dev/m3io 0 6 16 84 1
                none
                em_cnt1_ert3_io_ret_status
 set_mode       em_cnt1_ert3_io_read_reg /dev/m3io 0 6 16 71 1
                none
                em_cnt1_ert3_io_ret_status
```

Figure 1: An example of config.tbl file for EM.

2.3 汎用的な関数

e-RT3 には、それぞれを識別するためのモジュール番号、モジュール内のスロット番号やチャンネル番号等多くの情報を持つ。また温度計モジュール等の特定のモジュールでは、初期化や値の取得・設定

ueda_s@spring8.or.jp

のための手順等が必要になる。従来であれば、初期化や終了処理は SVOC コマンドを必要としない初期化・終了処理専用の関数内で、手順等は専用のソースコードを作成し記述してきたが、汎用的な EM の作成するために、それらの情報を config.tbl に記述し、ソースコードには記述しないこととした。

初期化・終了処理は、通常の SVOC コマンドと同じ config.tbl に記述し、初期化・終了処理のための専用の関数の中で SVOC コマンドを呼び出すこととした。SVOC コマンドの中にホスト名入れることで、汎用性を高めている。また使用するために設定手順が必要なモジュールのために、1 つの SVOC コマンドから複数の SVOC コマンドを連続して呼び出す関数を用意した。そのため、作成したソースコードは単純な機能だけのシンプルなものとなった。

e-RT3 は様々な IO モジュールを実装でき、モジュール上のレジスタへのアクセスタイプも数多く存在するが、今回ターゲットとしたのは DI・DO 等の入出力接点、データを設定取得するための入出力レジスタ、IO リレーにデータが設定されたことを知るための割込み、シーケンスデバイス、及び他の CPU モジュールと通信する共有メモリである。それぞれの種類に対応するハードウェア制御関数の作成を行った。変換処理に属する関数としてアナログ値をデジタル値に変換する関数を、結果処理の関数として、ハードウェアから取得した値をそのまま返信する関数、取得したデータを配列として返信する関数、デジタル値をアナログ値に変換し返信する関数の作成を行った。

Figure 2: Structure of the general purpose EM for e-RT3.

2.4 MADOCA II[3]対応

SPring-8 では現在 MADOCA フレームワークから MADOCA II フレームワークに移行を行っている。そこで、e-RT3 でも MADOCA II フレームワークを動作させるための調査及び移行を行った。MADOCA II フレームワークでは、通信に ZeroMQ[4] 及び MessagePack[5]を使用しているが、MessagePack においてメモリのバイトオーダがビッグインディアン時の処理に不具合があったため、２０１４年２月時点での最新のバージョンにアップデートを行い、移行を行った。

3. 設定ファイル変更システム

3.1 システム構成

前述の通り EM には機器を制御するために必要な情報を渡すための config.tbl が存在するが、汎用的な関数にしたことで、config.tbl に設定する情報が多くなった。情報量が多くなると内容の把握が難しくなる。そこで、簡単に内容を把握できるように Web ベースのユーザーインターフェースも同時に用意した。Web ブラウザを利用することとしたのは、EM が遠隔にあることや、クライアントとなる端末の OS が複数種類あるためである。Figure 3 に設定ファイル変更システムの構成図を記す。

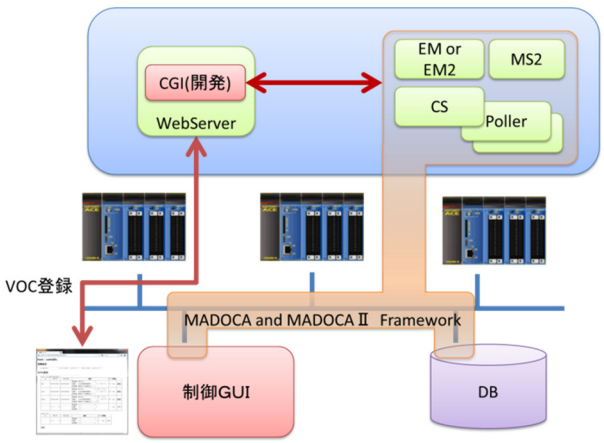

Figure 3: Schematic Diagram of configuration file modification system for e-RT3 general purpose EM.

3.2 サーバ処理

e-RT3 には Web サーバとしてデフォルトで thttpd[6]がインストールされており、それを使用することとした。Web サーバ側の処理は CGI 形式を利用し、C 言語で実装することとした。PHP や Python と比べに必要なリソースが少なくて済み、低速な CPU でも EM の処理に影響することなく動作させるためである。

サーバはクライアントが表示を行うのに必要な config.tbl の内容を、JavaScript とそのサブセットである JSON 形式(Figure 4)に変換をおこない、クライアントに送信する。汎用性を高めるために JSON 形式に変換するプログラムを単体のプログラムとして作成し、クライアントに送信する CGI と変換プログラムに分けることとした。またトラブル発生時の原因の切り分けを簡単行うためでもある。

クライアントに送信する情報として、config.tbl に書かれている関数や引数などの情報・説明文等を情報ファイルに記述し、それを送信することとした。情報ファイルは、フォーマットとして XML 形式(Figure 5)を利用している。これは、人が編集することを想定した場合、XML の要素名や属性名を記述

するため JSON と比べるとデータ量は多くなるが、区切り文字の記述ミス等による間違いを少なくするためである。また、XML であれば、タグのつけ忘れ等によるフォーマットの間違いはブラウザで簡単に見つけることができる。

```
"signal" : [
{
  "type" : "get",
  "name" : "srvacflwcnt-c21_slot6",
  "code" : [{
    "name" : "use_mode1",
    "func" : [
      { "name" : "em_cntl_ert3_io_read_reg", "arg" : [ "/dev/m3io", "0", "6", "16", "83","1" ] },
      null,
      { "name" : "em_cntl_ert3_io_ret_status" }
    ]
  },{
    "name" : "use_mode2",
    "func" : [
      { "name" : "em_cntl_ert3_io_read_reg", "arg" : [ "/dev/m3io", "0", "6", "16", "84", "1" ] },
      null,
      { "name" : "em_cntl_ert3_io_ret_status" }
    ]
  },
}
```

Figure 4:　An example of json file for EM.

```
<board device_type="/dev/m3io">
  <name>e-RT3</name>
  <em_func name="em_cntl_ert3_io_read_reg" type="control">
    <desc>入出力レジスタの値を取得する。</desc>
    <inputs />
    <args>
      <fixed_arg>
        <value name="device" type="device" visible="false" >
          <desc>制御するデバイスを指定</desc>
          <default>/dev/m3io</default>
        </value>
        <value name="unit" type="int" visible="false" >
          <desc>ユニット番号</desc>
          <min>0</min>
          <max>7</max>
          <default>0</default>
        </value>
        <value name="slot" type="int">
          <desc>スロット番号</desc>
          <min>1</min>
          <max>16</max>
          <default>2</default>
        </value>
```

Figure 5:　An example of XML file.

サーバはクライアントであるブラウザ側で作成された JSON ファイルから config.tbl を作成し、現在使用している config.tbl のバックアップファイルを作成し、置き換える。置き換えた config.tbl をブラウザに送信している。またクライアントから取得した JSON ファイルを config.tbl に変換するプログラムも作成している。サーバとクライアント間のデータフローを Figure 6 に記す。

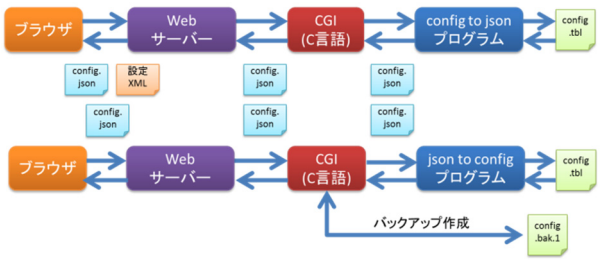

Figure 6: Data flow of the configuration file modification system.

3.3　クライアント処理

クライアントは HTML/JavaScript で実装を行った。クライアントはサーバから受け取った config.tbl から変換した JSON ファイル及び XML 情報ファイルから設定用画面を作成する。JavaScript では JQuery[7]及び Bootstrap[8]を利用し、構築している。これらのライブラリを使用しているのは、ダイアログボックス、ツールヒント等のユーザーインターフェースを作成するのに必要性の高い部品が用意されているためである。操作として、追加・削除・取り消し操作をマウス操作で行えるようにしている。変更が行われた場合は、その箇所の色を変更することで、変更が行われた箇所の確認を行いやすくしている。登録を行う場合は、画面から JSON ファイルを作成し、サーバに送信を行っている。登録が正常に行われた場合、サーバ側で生成された新しい config.tbl 自体が送信され、それを画面に表示して、間違いがないことが確認できるようになっている。

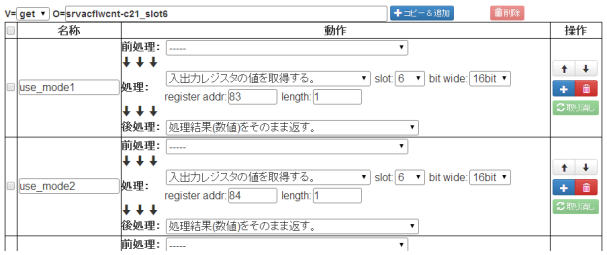

Figure 7: Web interface of the client side.

また設定中に SVOC コマンドが重複している等の問題が発生した場合には、その箇所のエラー表示を行い、エラー内容を表示する機能を搭載している。画面上部に状態表示欄を設け、現在エラー項目数、変更項目数、削除項目数を表示し、またそこから各変更が行われた場所へリンクを張り、利用者が簡単に変更箇所や問題が発生している箇所を表示できるように構築を行った。

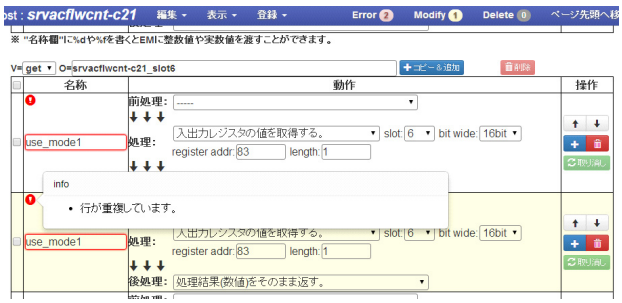

Figure 8: Error display.

4.　まとめ

今回で、汎用的な EM の基本部分は作成できた。今後は、更に汎用的になるように SVOC コマンドの連続実行時に次の SVOC コマンドにデータを渡せるように改良してく予定である。また、設定ファイル変更システムを構築したことにより、情報量が多くなった config.tbl の内容確認等が行いやすくなった。

参考文献

[1] R. Tanaka, et al., “The first operation of control system at the SPring-8 storage ring”, Proceedings of ICALEPCS97, Beijing, China, 1 (1997).
[2] http://www.e-RT3.com
[3] T.Matsumoto, et al., “Development of New Control Framework MADOCA II at SPring-8”, Proceedings of the 10th Annual Meeting of Particle Accelerator Society of Japan, Nagoya, Japan, Aug. 2013, 14.
[4] http://zeromq.org/
[5] http://msgpack.org/
[6] http://www.acme.com/software/thttpd/
[7] http://jquery.com/
[8] http://getbootstrap.com/