

MADDOCA-to-EPICS ゲートウェイの開発 DEVELOPMENT OF THE MADDOCA-TO-EPICS GATEWAY

清道明男[#], 増田剛正

Akio Kiyomichi[#], Takemasa Masuda

Japan Synchrotron Radiation Research Institute (JASRI)

Abstract

For the new device evaluation towards the SPring-8 upgrade project, we were faced with a request to temporarily integrate the new BPM signal processing system (Libera Brilliance+) with built-in EPICS IOC into the SPring-8 storage ring control system. We have developed the MADDOCA-to-EPICS gateway. In the equipment control layer of MADDOCA, Equipment Manager (EM) is deployed on the front-end computers. The MADDOCA-to-EPICS gateway is implemented as general-purpose EM to deal with EPICS IOCs. The gateway consists of general EM functions for interacting with IOC using Channel Access (CA), which are corresponding to the command such as caput, caget and camonitor in the EPICS system. We can build the gateway by specifying EM functions for CA and PV names in the EM configuration file. We will report on the development of the MADDOCA-to-EPICS gateway and an application to the evaluation system of Libera Brilliance+ installed into the SPring-8 storage ring.

1. はじめに

SPring-8 では基幹制御フレームワークとして MADDOCA (Message And Database Oriented Control Architecture)[1]を採用している。MADDOCA はクライアント・サーバモデルに基づいた分散型制御システムであり、遠隔の VME 等のフロントエンド計算機に対してテキストベースのメッセージを送り、VME はその応答を同じくメッセージとして送り返す。MADDOCA は加速器の高度化などに対応するため、画像や波形データなどの可変長データの取り扱いや Windows 対応などの機能拡張をして MADDOCA II [2]として発展している。

一方、加速器施設でよく利用されている分散型の制御システムには EPICS[3]や TANGO[4]といった制御システムがある。加速器向けの市販機器にはこれらの制御システムに準拠した機器があり、例えば BPM 信号処理システムの Libera はあらかじめ EPICS と TANGO が組み込まれている。このような EPICS 準拠デバイスを MADDOCA 制御システムの前で簡便かつ迅速に導入できることを目的に、MADDOCA-to-EPICS ゲートウェイの整備を行った。

2. 開発方針

MADDOCA はメッセージ駆動型のフレームワークである。制御命令を行う GUI 部のレイヤーとハードウェアを制御する機器制御レイヤーに分かれており、Message Server を介してメッセージを送る。機器制御レイヤーでは Equipment Manager (EM) と呼ばれるソフトウェアが動作している。EM は上位から抽象化された命令を受け取り、解釈して、物理デバイスとの対応付けを行い、デバイスドライバを呼び出して命令を末端の機器に伝える。そして機器からの応答を受取りこれを論理的な値へと抽象化して上位に返す。これら一連の制御を行うにあたり EM は変換処理・機器制御処理・結果処理の3つの処理関

数を持つ。そして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を EM 構成ファイル (config.tbl)で行う。配列などの可変長データを取り扱えるよう MADDOCA II をベースとする。

EPICS は通信モデルとしてクライアント・サーバモデルと publish-subscribe (出版・購読) 型モデルを採用している。各コントローラに用いられる通信プロトコルは EPICS IOC (Input/Output Controller) によって CA (Channel Access) プロトコルとして統一的に扱われる。IOC 上にはレコードのデータベースがあり個々のレコードがデバイスやデバイスの制御を表している。よって EPICS のクライアントシステムは CA プロトコルに基づいて開発を行えば良いことになる。

MADDOCA-to-EPICS ゲートウェイの基本構成を Figure 1 に示す。MADDOCA 制御システムにおいて EPICS 準拠デバイスを導入するために、EPICS の CA プロトコルに基づいた処理を EM の汎用関数として整備し、EM 構成ファイル内に関数と EPICS レコード名を指定するだけで、新たな開発をすることなく制御できることを方針とした。

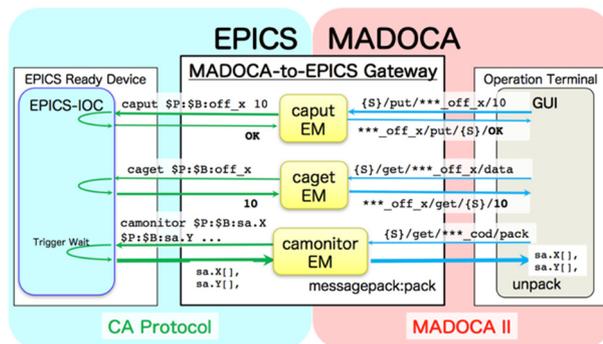


Figure 1: Scheme of MADDOCA-to-EPICS gateway.

[#] kiyomichi@spring8.or.jp

3. MADOCA-to-EPICS ゲートウェイ

EPICS 準拠のデバイスはレコード名で指定される EPICS レコードに対して、CA プロトコルに基づいたコマンド (caget, caput, camonitor) を使用すれば基本的には全ての制御を実現できる。以下にコマンドの役割を示す。

- caput : レコードに対して値をセットする
- caget : レコード状態を返す
- camonitor : 常にモニターし、状態に変化があればレコード状態を返す

MADOCA-to-EPICS ゲートウェイでは、EPICS における caget, caput, camonitor コマンドに相当する処理を EM 関数として整備した。作成した関数を Table 1 に示す。

Table 1: EM Functions of MADOCA-to-EPICS Gateway

Type	EM functions
caput	em_cntl_epics_caput_char.c em_cntl_epics_caput_double.c em_cntl_epics_caput_int.c
caget	em_cntl_epics_caget.c em_cntl_epics_caget_msgpack_arrayint32.c
camonitor	em_cntl_epics_camonitor_msgpack_arrayint32.c (em_cntl_epics_camonitor_msgpack_arraydouble.c) em_cntl_epics_camonitor_msgpack_wfminarrayint32.c
return	em_cntl_epics_ret_string.c em_cntl_epics_ret_clock.c

3.1 開発環境の構築

EM は Solaris10 の仮想ホストで運用する。MADOCA の開発環境において CA プロトコル通信機能を使用するために、EPICS の base パッケージ (base-3.14.12) より以下のライブラリを導入した。

- libca.a : EPICS CA Client Library
- libCom.a : EPICS Common Library

3.2 caput/caget 対応 EM 関数

EPICS の caput/caget の動作は MADOCA における設定 (put) / 取得 (get) の命令と同等である。MADOCA-to-EPICS ゲートウェイとして設定関数・取得関数は、caput/caget のソースコードを参考にして EM 関数を作成した。設定関数は C 言語の文法に沿った型指定で値を入力するため、int 型、double 型、char 型それぞれの関数を用意した (em_cntl_epics_caput_int, 他)。取得関数 (em_cntl_epics_caget) は char 型とし、結果処理関数 (em_cntl_epics_ret_string) にて char 型のまま処理する。レコード “\$(P):ioc:date” によって得られる IOC の日時情報については unix 時間に変換する結果処理関数 (em_cntl_epics_ret_clock) を用意した。MADOCA のツールコマンドの tellms2 により、コマンドラインから設定・取得が可能となる。

配列データの取り扱いには MADOCA II で実現した可変長データ対応の機能を用いて実装した。メッセージ交換に用いるデータは MessagePack[5]を用いてシリアライズ化 (pack) する (em_cntl_epics_caget_msgpack_arrayint32)。可変長データ対応の場合は、コマンドラインで取得することは出来ないが、GUI 等の受け取り側のプログラムで復元 (unpack) することが出来る。

Figure 2 に EM 構成ファイルの例を示す。caput/caget 対応 EM 関数の引数部分にレコード名を指定することで、レコードに対する値の設定・取得が可能となる。

3.3 camonitor 対応 EM 関数

データ収集用の関数として camonitor 対応 EM 関数 (em_cntl_epics_camonitor_msgpack_arrayint32) を整備した。EPICS ではデータを IOC 側からストリーミング配信しており、push 型のデータ収集系である。camonitor は登録したレコードをモニターし続けることができるコマンドである。

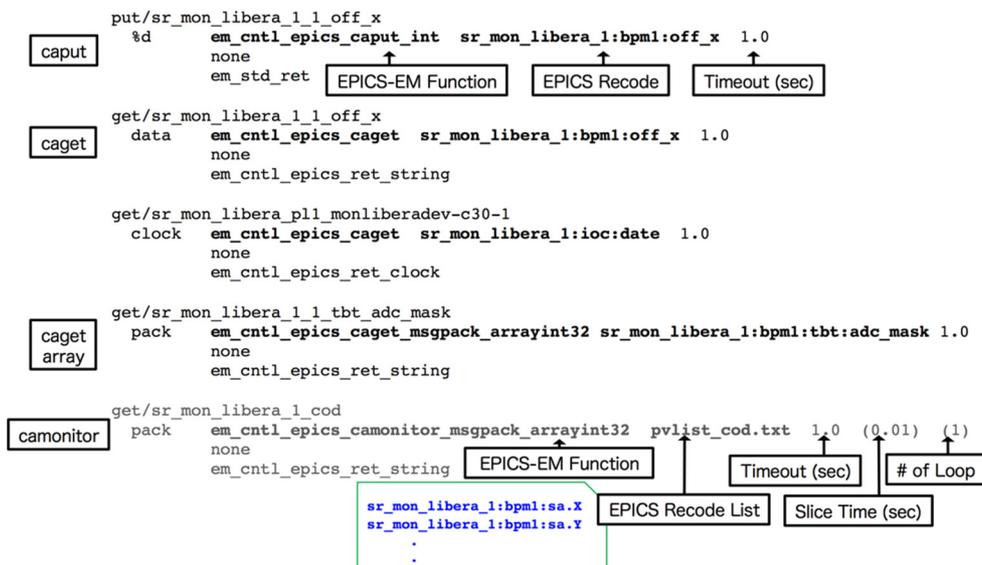


Figure 2: An example of config.tbl for caput/caget/camonitor EM.

一方、MADOCA のデータ収集は上位から取りに行く pull 型を基本としている。そのため camonitor 対応 EM 関数は、レコードの登録・IOC との接続・データ取得・レコードの解除、という一連の処理で pull 型のデータ収集系を実装した。一回のデータ取得は、指定回数分の更新を EM 側のメモリに溜めた上で引き出す形となる。データは int 型の配列データで MessagePack を用いてシリアル化(pack)する。

また、複数のレコードから同期したデータを収集するために、レコード名はリスト化したファイルで与えることとした。Figure 2 下に camonitor 対応 EM 構成ファイルの例を示す。データレートやまとめる回数に応じて、タイムアウト値や検知周期を決める。

4. BPM 信号処理システムへの適用

SPring-8-II に向けた BPM 信号処理システムの候補のひとつに Libera Brilliance+ [6] がある。これの長期安定性の評価を行うにあたり、SPring-8 蓄積リングへ設置し現行の BPM 信号処理システムと同時に運転しデータ収集を行う。そのために MADOCA 制御システムに組み込む必要がある。また SACL A において、コンパクトで PoE (Power-over-Ethernet) 対応の BPM 信号処理回路 Libera SPARK [7] の試験を行いたいとの需要もある。

Libera の制御は、開発キットを用いてシステム上で直接動作する EM を開発する方法と、ベンダーが提供する EPICS または TANGO のインターフェースを利用する方法がある。今回は評価試験として簡便かつ迅速に導入したいことから、MADOCA-to-EPICS ゲートウェイを適用することで MADOCA 制御システムへの組み込むこととした。

4.1 Libera ソフトウェア

Libera Brilliance+ のソフトウェア構成を Figure 3 に示す。Libera のソフトウェアは幾つかの階層に分かれており、ユーザは MCI (Measurement and Control Interface) を通して制御することができる。また、MCI を介した EPICS-IOC が提供されている。

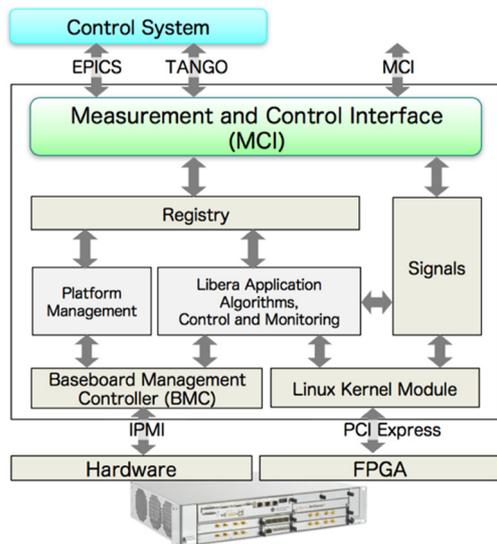


Figure 3: Libera Brilliance+ software structure.

Libera Brilliance+ はハードウェアとしては MicroTCA の筐体をベースに作られており、CPU は Intel 系で Ubuntu Linux を搭載している。Libera SPARK の CPU は ARM である。もし、MCI を使用してシステム上で直接動作する EM を開発する場合には、それぞれの CPU に合わせた MADOCA の開発環境を作成する必要がある。この点からも、MADOCA-to-EPICS ゲートウェイは迅速な導入に役立つ。

4.2 Libera の制御およびデータ収集

Libera Brilliance+ を SPring-8 蓄積リングへ導入し MADOCA 制御システムへ組み込んだ。各種パラメータの設定や読み返しは EPICS に登録されているレコードに対して caput/caget EM を用いて制御する。ファンや電源のステータスといった監視データは数百に及ぶが caget EM で取り扱うことが出来る。収集の対象となるデータは以下のとおりである。

- COD: 8pt, 9.708Hz
- Single Pass (SP) : 16384 turns, 0.1Hz
- Fast Acquisition (FA) : 10kHz, 128kpt(10sec)

COD は約 10Hz で camonitor EM から直接データベースに書き込む。FA は 10kHz のデータであるが、Libera からは 1024 ずつまとめて 10Hz のレートで取得出来る。これを camonitor EM で約 10 秒分溜めて、配列データとして取得する。FA および SP データは一回の測定でのデータ量が大きいので GUI 経由でファイルに保存することになっているが、将来はデータベースへ直接書き込むことも検討している。データベースおよびファイルに保存したデータは GUI 上で表示も行う。Libera Brilliance+ の制御およびデータ収集の構成を Figure 4 に示す。

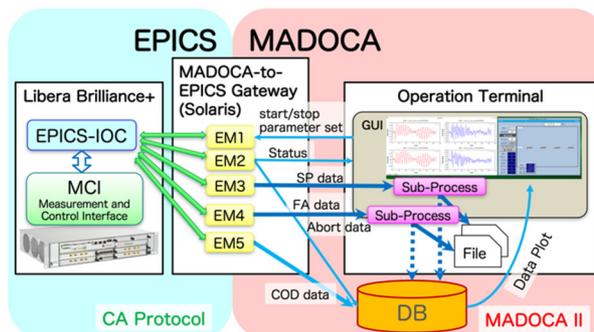


Figure 4: Scheme of control and data acquisition for Libera Brilliance+.

4.3 性能および課題

MADOCA 制御システムの下での Libera Brilliance+ の制御は MADOCA-to-EPICS ゲートウェイを用いて実現し、COD、SP、FA の各データは継続的な読み込みに成功した。データ収集において camonitor EM は接続処理が最も時間がかかり、30~50ms ほど費やしている。他の処理は COD 測定の場合 1ms 以下である。安定にデータ収集できるレートは 10Hz 程度までであり、COD 測定は 10Hz のデータ収集が安定に動作している。

データ収集において 10Hz 以上の高レートが必要な場

合は、今回実装した pull 型のデータ収集系では困難であり、push 型ストリーミングによるデータ配信の方法を検討している。

5. まとめ

MADDOCA 制御システムにおいて EPICS 準拠デバイスを簡便かつ迅速に導入できるようにするために、MADDOCA-to-EPICS ゲートウェイを開発した。SPring-8 II に向けた BPM 信号処理システムの評価試験に適用し、その機器の制御やデータ収集を行うことができた。

今後は未整備の型への対応を行い、汎用化を進める。また、push 型のストリーミングデータへの対応を検討している。

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, 1997, p.1.
- [2] T. Matsumoto *et al.*, “Development of New Control Framework MADDOCA II at SPring-8”, Proceedings of the 10th Annual Meeting of Particle Accelerator Society of Japan, Nagoya, Japan, Aug. 2013, p.14.
- [3] <http://www.aps.anl.gov/epics/>
- [4] <http://www.tango-control.org/>
- [5] <http://msgpack.org/>
- [6] <http://www.i-tech.si/accelerators-instrumentation/libera-brilliance-plus/>
- [7] <http://www.i-tech.si/accelerators-instrumentation/spark-eh/>