ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA149

MADOCA TO EPICS GATEWAY

A. Kiyomichi, T. Masuda
Japan Synchrotron Radiation Research Institute (JASRI), Sayo, Hyogo, Japan

Abstract

MADOCA-to-EPICS gateway has been developed for easy and rapid integration of EPICS-ready devices into MADOCA, the control software framework for SPring-8 and SACLA. MADOCA uses equipment control software called equipment manager (EM) in the device control layer. The MADOCA-to-EPICS gateway is implemented as a general-purpose EM to handle EPICS IOCs. The gateway consists of EM functions that interact with IOCs using channel access (CA) protocol corresponding to EPICS commands such as caget, caput and camonitor. We can build the gateway for the target EPICS device by editing the EM configuration file, without any programming. We have applied the gateway to the Libera Brilliance+, installed in the SPring-8 storage ring, to be evaluated towards the SPring-8 upgrade project. In addition, it has been applied to the Libera Brilliance Single Pass and Spark installed in beam transport line, and the Libera Spark and Cavity installed in SACLA. The gateway is helpful in minimizing the installation time and effort, even for the different platform (CPU and OS) devices. We will report on the development and advantage as well as the performance improvement of the MADOCA-to-EPICS gateway.

INTRODUCTION

The MADOCA (Message And Database Oriented Control Architecture) [1] control framework was developed at SPring-8 and utilized in the control systems of SPring-8 and SACLA. MADOCA is a message-oriented control system based on a client-server model. It sends a text-based message to a remote VME or another front-end computer and the VME sends back the response as a message.

In the meantime, there is EPICS (Experimental Physics and Industrial Control System) [2] and TANGO [3], which are popular control systems for many accelerator facilities. There are commercially available devices compliant with these control systems. For example, Libera, the BPM signal processing system, has in-built EPICS and TANGO.

We have developed MADOCA-to-EPICS gateway for easy and rapid integration of EPICS-ready devices into MADOCA control system.

DESIGN POLICY

The design policy of MADOCA-to-EPICS gateway development is as following.

- Provide as a general-purpose EM functions.
- Build the gateway EM by editing the configuration file without any programming.

In order to realize this, we have developed based on the features of MADOCA and EPICS described below.

MADOCA is a message driven client-server model control framework covering layers from presentation to equipment control. In the equipment control layer, the equipment manager (EM) receives a command from the operating terminal, interprets it, associates with the physical devices, and sends a command to the target device via the device driver. Therefore, it accepts the response from the device, converts it to a logical value, and returns it to the operating terminal. In performing these control schemes, EM has three processing functions: a function to interpret a message, a function to control a device, and a function to abstract as a message. The received message (SVOC command), three processing functions, and the control target device are mapped in the EM configuration file.

EPICS employs a client-server model and a publish-subscribe model as a communication model. The EPICS input/output controller (IOC) runs on each device and communication is handled uniformly by a channel access (CA) protocol. There is a database of records on the IOC with each record representing a device or its control. Essentially, all controls can be realized with commands based on CA protocols such as caput which sets a value to a record and caget which returns a record status.

In order to integrate EPICS-ready devices into the MADOCA control system, we have developed a general-purpose EM function for processing based on EPICS CA protocol. This allows us to build the gateway EM by editing the EPICS record name in the EM configuration file, without any programming. The basic scheme of the MADOCA-to-EPICS gateway is shown in Figure 1.

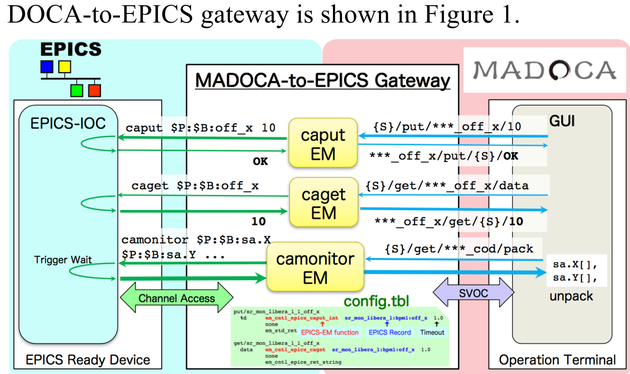

Figure 1: Scheme of MADOCA-to-EPICS gateway.

GATEWAY EM

In EPICS, all controls can be realized using the CA protocol-based commands (caget, caput, camonitor) to the EPICS process variable (PV) specified by the record name. The role of the command is shown below.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA149

- caput: Put value to PVs
- caget: Get value of PVs
- camonitor: Set up monitors and continuously update incoming value for PVs.

The MADOCA-to-EPICS gateway has been developed as EM functions corresponding to the caput, caget and camonitor commands in EPICS. The developed EM functions list is shown in Table 1 and an example of EM configuration file (config.tbl) is shown in Figure 2. In the EM configuration file, we can 'put' and 'get' the PVs by specifying the recode name as the argument of the EM function.

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
	em_cntl_epics_caput_arraydouble.c
	em_cntl_epics_caput_arrayint.c
	em_cntl_epics_caput_msgpack_arraydouble.c
	em_cntl_epics_caput_msgpack_arrayint32.c
caget	em_cntl_epics_caget.c
	em_cntl_epics_caget_waveform.c
	em_cntl_epics_caget_msgpack_arraydouble.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_wfmarrayint32.c
	em_cntl_epics_camonitor_msgpack_wfmarraydouble.c
return	em_cntl_epics_ret_string.c
	em_cntl_epics_ret_clock.c

Development Environment

The MADOCA-to-EPICS gateway EM is operated on VME or container (virtual host) of Solaris 10. In order to use the CA protocol functions in the development environment of MADOCA, the following libraries were installed from EPICS base-3.14.12 package.

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

The Caput/Caget EM Functions

The caput/caget commands of EPICS are equivalent to 'put/get' operation in MADOCA. As a MADOCA-to-EPICS gateway, 'put' and 'get' EM functions were created with reference to the source codes of caput and caget commands, respectively. Because the 'put' EM function inputs the value by type specification according to the C language grammar, functions for each integer type, double type, and character type are prepared. The 'get' EM function (em_cntl_epics_caget) is of type char, and it is processed as char type by the return function. We prepared a special return EM function (em_cntl_epics_ret_clock) that converts the date and time information of the IOC obtained by the record "$(P):ioc:date" into UNIX time.

MADOCA implements pull-type data collection which takes data from host computers. Because the first version of the 'get' EM function executed CA connection and disconnection every time, there was a limit to the data collection rate. This was due to an overhead of approximately 30 ms per access by CA reconnection processing. Therefore, we improved the data collection rate by modifying the gateway to avoid the CA reconnection.

The first time PV is accessed, a CA channel is created and the connection status is stored in a linked list. The list is declared as a static variable, and information is saved even if it exits the function. It takes approximately 30 ms to call the record by the EM function for the first time; however, it can reduce it to less than 1 ms after the second access. We achieved high-speed data collection by reducing the overhead to 1 ms or less.

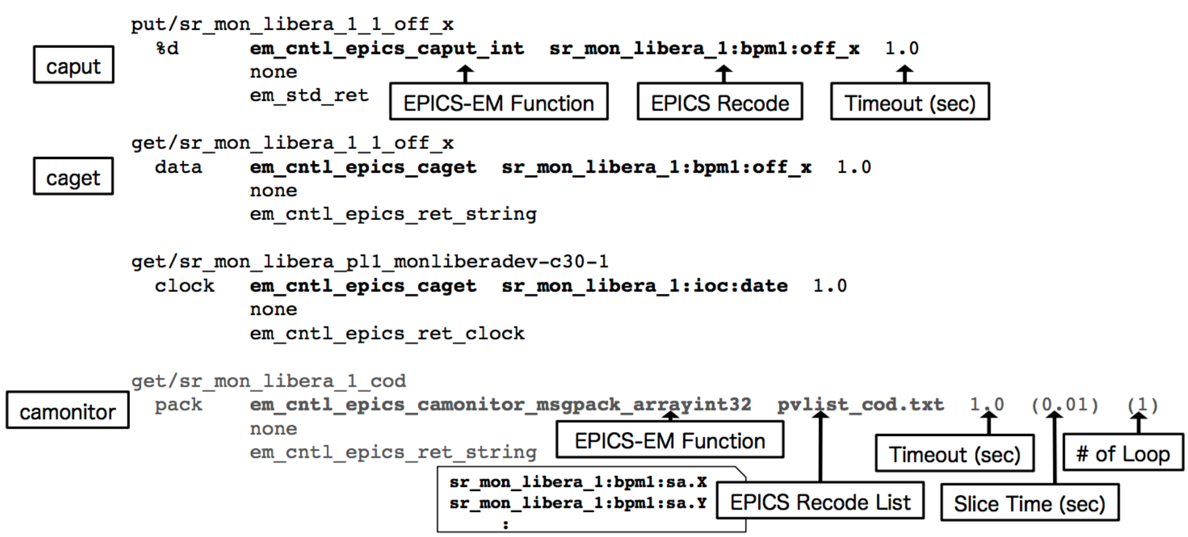

Figure 2: An example of EM configuration file (config.tbl) for caput/caget/camonitor EM.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA149

The Camonitor EM Function

The camonitor compatible EM function was prepared as a function for data collection. EPICS is a push-type data collection system that distributes data streaming from the IOC side. However, MADOCA implements a pull-type data collection system, which takes data from host computers. As a result, the camonitor EM function implements a pull-type data collection system by a series of processes such as registration of records, connection with IOC, data acquisition, and release of records.

The camonitor EM function was developed as a function dedicated to MADOCA II [4] that can handle variable length data by messaging. This was used at SPring-8. The array data is serialized using MessagePack library [5]. Therefore, the GUI or client process receives and unpacks it via messaging, and performs calculation processing and database writing.

APPLICATIONS

In SPring-8 and SACLA, we installed several types of the BPM processing systems such as Libera Brilliance+, Libera Brilliance Single Pass, Libera Spark and Cavity BPM. These devices are products from the same manufacturer; however, the CPU and OS are different depending on usage or manufacturing age. Therefore, EPICS IOC and development kit dedicated to each platform are provided. If we develop the EM that runs directly on the platform without using IOC, it would take a lot of time and effort to build MADOCA development environment and EM operating environment for each platform. However, the gateway offers us the benefits of minimizing the installation time and effort even for the different platform (CPU and OS) devices.

Evaluation system towards the SPring-8 upgrade project

We have applied the gateway to the BPM processing system, Libera Brilliance+ [6], installed in the SPring-8 storage ring, to be evaluated towards the SPring-8 upgrade project. Libera Brilliance+ has Intel CPU and Ubuntu Linux and the software is divided into several layers. The user can control through the Measurement and Control Interface (MCI), and an EPICS-IOC via MCI is also provided. We applied the MADOCA-to-EPICS gateway and realized the control and data acquisition via EPICS-IOC as shown in Figure 3. The MADOCA-to-EPICS gateway operates on the Solaris 10 container, and there are six EMs and three data collectors (DC), which make up the data collection process of MADOCA II [7], are run as follows.

- EM1, DC1: 0.1 Hz, platform status data, clock
- EM2, DC2: 1 Hz, maximum adc, gain, phase, etc.
- EM3, DC3: 10 Hz, Slow Acquisition (SA) data
- EM4: Single Pass (SP) data by GUI
- EM5: Fast Acquisition (FA) data: 10 kHz, 128 kpt.
- EM6: ADC data

Periodic data collection is written to the database by DC for each cycle. DC1 and DC2 collect device monitoring data such as fan and power supply status, gain and phase data. DC3 collects slow acquisition (SA) which is COD (Closed Orbit Distortion) data at 10 Hz. The beam position data is obtained by acquiring the voltage data of each electrode inside the Libera via the EM3. The COD data is calculated on DC3 using the nonlinear map function of BPM, and written it to the database together with the voltage data. Because the amount of fast acquisition (FA) and single pass (SP) data is large in one measurement, it is saved in a file via the GUI. Although the FA data rate is 10 kHz, one array data containing data of 128k points for approximately 10 seconds is acquired by camonitor EM, and it is acquired by the GUI and saved in a file.

The system was operated from the Fall of 2016, and data collection has bend stabled.

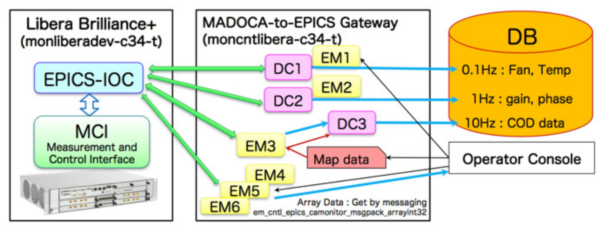

Figure 3: Scheme of control and data acquisition for Libera at the SPring-8 SR.

SPring-8 SSBT Strip Line

In the SPring-8 SSBT (Synchrotron to Storage ring Beam Transport) strip line, three Libera Brilliance Single Pass [8], which is the old type of Libera Brilliance+, are used for the BPM signal processing system. Libera Brilliance Single Pass has Ubuntu Linux on ARMv5 CPU, and users exercise control through Libera Control System Programming Interface (CSPI: previous version of MCI). EPICS-IOC via CSPI is also provided. When we originally installed them several years ago, we used ARMv5 cross-compiler and CSPI development kit to develop and operate EM directly running on the Libera.

From 2017, we added two Libera Spark [9] and decided to operate a total of five BPM signal processing systems. Libera Spark's CPU is Zynq (ARMv7), OS is equipped with i-tech Linux, MCI and EPICS-IOC are provided for user development. If we develop the EM directly running on the Libera Spark, there is no binary compatibility with Single Pass. Thus, we need to prepare ARMv7 cross compiler and the MCI development kit in addition to the CSPI development kit. Furthermore, as the CPU is different, it is not compatible with Brilliance+, which uses the same MCI. For that reason, we decided that it would be costly to maintain this development system, and thus, shifted to a method of applying MADOCA-to-EPICS gateway. Figure 4 shows the scheme of control and data acquisition for Libera at SPring-8 SSBT strip line.

We prepared EM and DC for each of the five Libera. The DC process acquires 1024 points of AD data via EM,

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA149

performs calculations such as x and y positions, and writes them to the database at 1Hz. In addition, we prepared the monitoring process, DC6, for all Libera and collected the status information at the cycle of 10 seconds.

In the future, we will inject beams from SACLA into the storage ring and we plan to operate at a maximum of 10 Hz. The data acquisition test of 10 Hz was successfully performed.

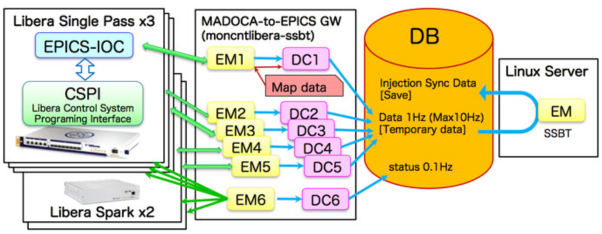

Figure 4: Scheme of control and data acquisition for Libera at SPring-8 SSBT strip line.

SACLA BPM

To use Libera Spark and Libera Cavity BPM in SACLA, MADOCA-to-EPICS gateway was applied. Because the SACLA control system is MADOCA, we only use the caput/caget EM functions without the camonitor EM functions which are based on MADOCA II. Figure 5 shows the control and data collection scheme of Libera Spark and Libera Cavity BPM at SACLA. The MADOCA-to-EPICS gateway operates on two VMEs with Solaris 10 installed.

One VME is built in the synchronous data acquisition system of SACLA, which inputs the trigger signal synchronized with the beam to VME and Libera. The data collection process 'poller' runs for each Libera. Because the data on EPICS DB is updated by reading with camonitor, caget EM alone cannot obtain the latest value. Therefore, by executing camonitor on Libera, it is possible to acquire the latest value with caget EM function. The system was operated from January 2017, and a stable data collection has been achieved.

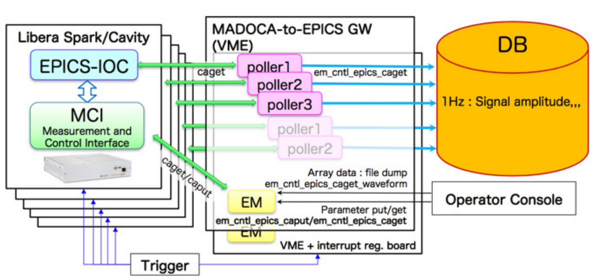

Figure 5: Scheme of control and data acquisition for Libera at SACLA.

Regarding the capability of the upgrade project to integrate both SPring-8 and SACLA control systems, the redesign of the control framework is underway [10]. In the test of the new control system from the Summer of 2017, MADOCA-to-EPICS gateway was also applied to the new data acquisition system MDAQ and the new MQTT-based EM for Libera control.

PROTOTYPE OF EXTENDED GATEWAY

The current MADOCA-to-EPICS gateway EM has overhead, owing to inquiring the IOC every time. By managing the CA connection status, several hundreds of data were steadily collected up to the rate of approximately 10 Hz, but it would be caused data loss if the rate is further increased. Because EPICS can also distribute data from IOC by a publish-subscribe model, we studied an extension of speed enhancement that realizes collection of several thousand points array like waveform data at 10 Hz or more using this function, and built a prototype of extended gateway.

In the extended gateway, all Channel Access operations are replaced with callbacks and subscriptions. The gateway EM stores PV values in local memory, and keeps track of disconnected PVs, while the CA buffer is periodically flushed in a separate thread. This design also gave us an option for a proper monitor implementation as PV updates can be stored in a buffer on the gateway. When the client program asks for monitored values, it receives all the data that was acquired from previous monitor commands. This enables clients to receive updates when they are ready without missing or duplicating values. An array of values is implemented as a ring buffer and, in case clients do not ask for an update, the oldest data will be overwritten first. The scheme of the prototype is shown in Figure 6.

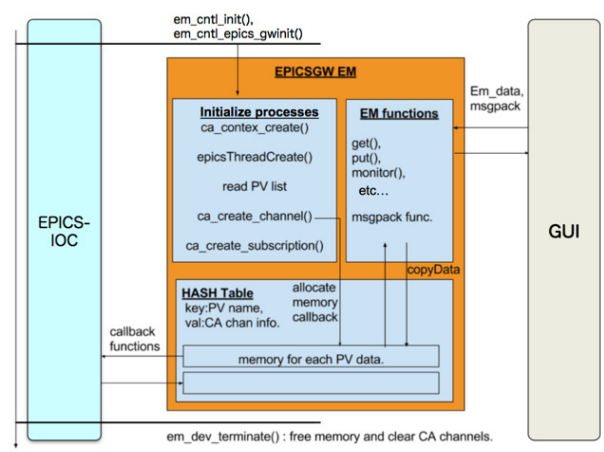

Figure 6: Scheme of extended MADOCA-to-EPICS gateway.

It is possible to update and read data asynchronously in the margin of the ring buffer. For example, it is supposed to use a method of buffering data at 60 Hz trigger period by 60 times, and to read it at 1 Hz. In order to evaluate the prototype, a readout test was performed using an EPICS simulator with 10000 array data updated at 1 Hz, 10 Hz, and 50 Hz cycles. The 50Hz update is the limit of simulator. In the data collection of the current version, it was possible up to 10 Hz, however data omission occurred at 50 Hz. Nonetheless, with the new method of asynchro-

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA149

nous data collection using buffers, it was possible to stably collect 50 Hz data.

CONCLUSION

The MADOCA-to-EPICS gateway has been developed so that EPICS-ready devices can be integrated easily and rapidly in the MADOCA control system. We have solved the overhead problem of CA connection and realized high-speed data collection. For faster data collection, we created a prototype that stores data in local memory using the callback function and confirmed its effectiveness.

We applied control and data collection to 11 Libera installed on an evaluation test of BPM signal processing system for SPring-8 upgrade project and on a current system of SPring-8 SSBT and SACLA. We achieved installation with minimum time and effort for three different platforms.

REFERENCES

[1] R. Tanaka *et al.*, "The first operation of control system at the SPring-8 storage ring", in *Proc.ICALEPCS'97*, Beijing, China, 1997, p.1.

[2] http://www.aps.anl.gov/epics/

[3] http://www.tango-controls.org/

[4] T. Matsumoto, et al., "Next-Generation MADOCA for the SPring-8 Control Framework", Proceedings of ICALEPCS2013, San Francisco, USA, 2013, p.944-947.

[5] http://msgpack.org /

[6] http://www.i-tech.si/accelerators-instrumentation/libera-brilliance-plus/

[7] T. Matsumoto, et al., "MADOCA-II Data Collection Framework for SPring-8", In this Proceedings of ICALEPCS2017, Barcelona, Spain, 2017, MOBPL04.

[8] http://cerncourier.com/cws/product/P000015259

[9] http://www.i-tech.si/accelerators-instrumentation/spark-el-hl/

[10] T. Fukui, et al., "Status of the Control System for the SACLA/SPring-8 Accelerator Complex", In this Proceedings of ICALEPCS2017, Barcelona, Spain, 2017, FRAPL03.