

RaspberryPiを用いた測定器データ収集システムの構築

- 安価に手軽に DAQ 構築 -

大阪市立大学

岩崎 昌子

masako@sci.osaka-cu.ac.jp

大阪大学

南條 創

nanjo@champ.hep.sci.osaka-u.ac.jp

KEK 加速器研究施設

帯名 崇, 上窪田 紀彦, 山本 昇

takashi.obina@kek.jp, norihiko.kamikubota@kek.jp, noboru.yamamoto@kek.jp

2018 年 (平成 30 年) 2 月 9 日

1 はじめに

RaspberryPi (RPI)[1] は, ARM プロセッサを搭載した手のひらサイズのシングルボードコンピューターである (図 1)。小型サイズながら, 1GB RAM を搭載し, 4 ポートの USB, Ethernet, GPIO インターフェースを備え, 無線 LAN も使用可能, という優れたものである (RPI3 モデル B)。しかも, 非常に安価である (~5000 円)。この RPi を用いると, 安価で可搬性に優れた測定器制御・データ収集システムが構築できる。限られた予算とスペースで運営している大学の研究室にとって, 安価で小型であることは, 重要なポイントである。

本稿では, RPi を用いた測定器データ収集システムについて, 1) 大阪市立大学, 2) 大阪大学, 3) J-PARC での事例を紹介する。また, RPi とは別の小型シングルボードコンピューターを用いた例として, 4) KEK-PF・cERL での事例についても紹介する。

2 大阪市立大学での例

大阪市立大学, 宇宙素粒子物理学実験研究室では, 学部 4 年生の卒業研究として, RPi (RPI3 モデル B) を用いたネットワーク分散型機器制御システムを構築した。RPI の OS は, 公式ホームページから提供される Raspbian(Linux) を使用している。RPI は microSD カードから OS を読み込み起動するため, Raspbian のイメージファイルを microSD カードに書き込んで導入した。

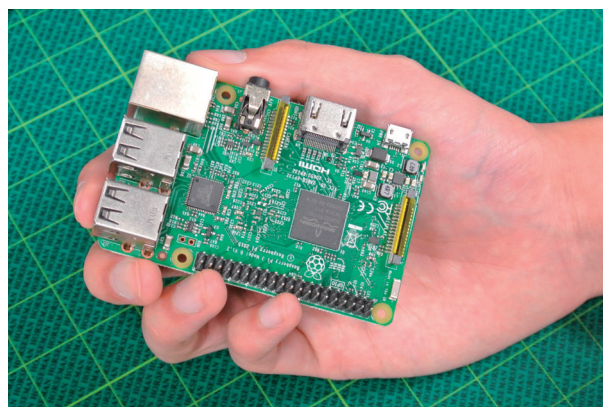


図 1: RaspberryPi3 モデル B

この RPi に, ネットワーク分散型制御を行うためのソフトウェア, EPICS(Experimental Physics and Industrial Control System [3]) を導入した。EPICS は国際共同開発されたオープンソフトウェアで, 素粒子・原子核実験, 加速器, 宇宙, 重力波実験等で使用されている。EPICS では, デバイスを直接制御するための計算機を IOC(Input Output Controller), 操作・監視を行うための計算機を OPI(Operator Interface) と呼ぶ。学生実験では, RPi を IOC として, 機器を制御した。制御した機器は, マルチメーター (IWATSU VOAC7523), デジタルオシロスコープ (Keysight InfiniiVision DSO-X 2002A), および Arduino[3](Arduino UNO) である。機器との通信は, Ethernet 通信, または USB-シリアル通信で行った。

ここで, Arduino は, AVR マイコン, デジタル・アナロ

グ信号入出力ポートを備えた安価な基板である（～2600円）。公式ホームページから提供される IDE（統合開発環境）を用いて、Arduino を制御するためのソースコード（スケッチファイル）を作成し、基板にアップロードできる。Arduino UNO は、10-bit の ADC を搭載しているため、Arduino を、センサーからのアナログ信号をデジタル変換させるための ADC として使用した。

Arduino の IDE には、様々なサンプルスケッチファイルが、あらかじめ用意されている。R Pi に Arduino IDE を導入し、サンプルスケッチファイルを用いて、LED 点灯や、温度センサーからの温度データ取得等を行い、Arduino による機器制御を行った。次に、Arduino IDE に用意されたシリアルモニタを使用し、R Pi からのシリアル通信により、読み込み・書き込み命令を行うプログラムを作成して、Arduino の制御を行った。

Arduino は適用例が多く、WEB 上でプログラム例を検索することも可能であり、制御の基礎を学習するツールとして手軽に使用することができる。スターターキット等も豊富にあり、学生が自ら、プログラミングと実習を進めることができる。一連の実習で、学生が Arduino の取扱いを習得した後、R Pi 上にある EPICS を用いて、シリアル通信により Arduino の制御を行うプログラムを作成した。

図 2 に、R Pi による温度センサーからのデータ取得のセットアップを示す。温度センサー（ナショナル セミコンダクター製 LM35DZ）で室温を測定し、センサーからのアナログ電圧信号を、マルチメーター、および Arduino によってそれぞれ AD 変換し、マルチメーターからは Ethernet 通信で、Arduino からはシリアル通信により、R Pi が取得する構成とした。

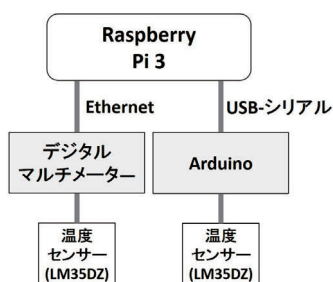


図 2: R Pi による温度センサーからのデータ取得のブロック図

Arduino によって得られた測定温度（測定精度 0.5 度）と、マルチメーターから同時に得られた測定温度（精度 0.001 度）を比較したところ、Arduino の測定精度 1 ビット分のずれがみられたものの、良い一致がみられた。Arduino からの出力データを校正すると、さらに良い一致が得られると思われる。2600 円の ADC だということを考慮すると、非常に満足のいく結果である。

また、R Pi と Arduino は、どちらもコンパクトなサイ

ズで、かつ低価格であるため、種々の講習・実習を行うのに適している。我々は、山本による R Pi・Arduino・ブレッドボード¹、という組み合わせを使用して、EPICS 講習会を行った。2017 年 2 月に大阪市立大学で実施した EPICS 講習会（11 名出席）では、上記の R Pi+Arduino を 5 セット使用して実習を行った。2017 年 5 月に国際ワークショップ [4] で実施した講習会では、講師用と参加者用に、R Pi+Arduino を 7 セット用意した。この講習会では、参加者が 30 名を超えたので、6 つのグループに分けて、各自のノート PC から R Pi にリモートログインして実習を行った。R Pi には無線 LAN が用意されているので、ネットワーク越しでの実習にも、非常に便利であった。

大学院生向けの教育として、デジタルオシロスコープからの波形取得実習も行っている。デジタルオシロスコープを、Ethernet 通信で制御して、リモートで信号読み取りレンジを変更し、波形データを取得する実習内容とした。ここで、オシロスコープの制御プログラムは、1) Python でプログラミングを行い socket モジュールを用いて制御を行うプログラムと、2) EPICS を用いて制御を行うプログラム、二種類を作成して、それぞれで制御を行った。機器制御を二つのアプローチで行う事で、より効果的に教育を行うことが出来た。「Raspberry Pi の”Pi” は、Python の略である [5]」とある通り、R Pi は Python を使用する環境が整っている。

3 大阪大学での例

大阪大学山中卓研究室では、図 3 のように、R Pi と計測器や回路を接続している。EPICS を介さず、直接 Linux コマンド、Python、C 言語を用いて計測制御を行っている。また、Arduino のようなマイコンを介さず、直接シリアル通信インターフェースを持つ IC と通信を行っており仲介機能を追加導入しない。

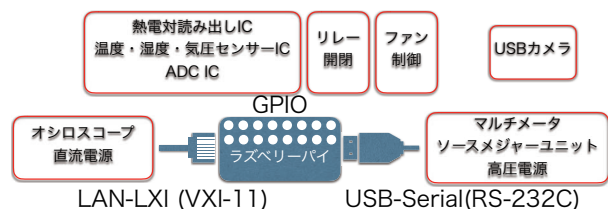


図 3: 大阪大学での R Pi と計測器、回路の接続

3.1 計測器との接続

表 1 に計測器の接続例を載せる。

¹4.1 節で、詳細を紹介している。

表 1: 接続した計測器の例

LAN-LXI(VXI11)		
オシロスコープ	TDS3000	Tektronix
オシロスコープ	DPO/MSO	Tektronix
直流電源	PMX-A	KIKUSUI
USB-Serial(RS-232C)		
マルチメータ	VOAC75xx	IWATSU
ソースメジャー	2400/2410	Keysight
高圧電源	NHS6xxx	iseg

表 2: SCPI コマンド例

*IDN?	機器個体識別問い合わせ	文字列返信
:READ?	データ読み出し	文字列返信

- LAN-LXI(LAN eXtensions for Instrumentation)
PC の LAN ポートと, LXI 機器を LAN ケーブルで接続する。PC, LXI 機器の IP アドレスを用い, TCP/IP をベースとする VXI-11 プロトコルで通信する。C 言語による Linux 用のライブラリが公開されており, これを一部変更し Mac OSX 用にも移植した。Tektronix のオシロスコープによる計測制御プログラムと共に公開した [6]。
- USB-Serial
PC の USB ポートと RS-232C(Recommended Standard 232c) 機器を, USB-Serial 変換ケーブルを介し接続する。Linux では USB-Serial という仕組みが標準装備されており, /dev/ttyUSB0 などのデバイスとして認識される。PC から特定のコマンド文字列を送信し (echo 文字列 > /dev/ttyUSB0), 必要であれば機器側からの文字列返信を受信し, 変数につめる (read -n 文字数 変数 < /dev/ttyUSB0)。特定のコマンド文字列は機器毎に定義されているが, ある程度共通化が図られており, SCPI (スキッピ, Standard Commands for Programmable Instruments) コマンドと呼ばれる (表 2)。

3.2 回路との接続

RPi で特に便利な点は, GPIO(General Purpose Input Output) ピンを装備し, 各 Pin でデジタル入出力が可能である。これを用いてリレー制御, ファン回転数制御を行う。熱電対読み出しや ADC 機能などをもつ IC との通信を取り上げる。

表 3: SPI 通信で用いる配線

SCLK	Serial Clock
MOSI	Master Out Slave In
MISO	Master In Slave Out
\overline{SS}	Slave Slect Bar
GND	Ground

3.2.1 リレー開閉

AC100V 機器を制御するのに, フォトカプラ入力のリリッドステートリレー OMRON G3CN-202P を用いた。3.3V(2.2mA) で開閉動作可能であり, GPIO3.3V から直接制御し, パトランプとブザーの作動に利用した。

3.2.2 ファン回転数制御

GPIO3.3V を直接トランジスタに供給することで, 5VTTL にロジックレベル変換し (図 4), ファンの回転数を Pulse Width Modulation(PWM) で制御した。5V も RPi のピンから供給し, 5mA 程度の電流消費である。

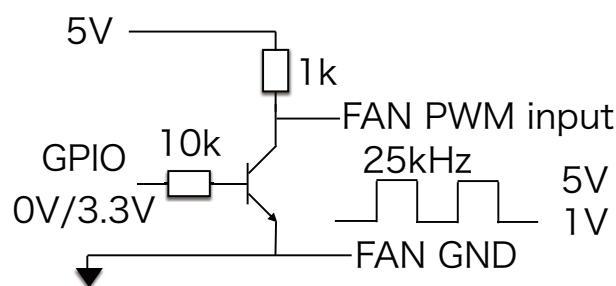


図 4: PWM 用 GPIO 接続回路 (抵抗値の単位は Ω)

3.2.3 シリアル通信

熱電対読み出し, 温度気圧湿度センサー, ADC 機能などをもつシリアル通信機能を備えた IC がある。RPi の GPIO ピンによりとこれらと通信し, 様々な計測ができる。今回は代表的なシリアル通信 SPI (Serial Peripheral Interface) を取り上げる。SPI では表 3 にあるように, グランドを除き, 4 本の配線で通信を行う。SCLK は一定周期である必要もなく, SCLK のエッジなどでの MOSI や MISO の信号レベルを用いて, IC で定める規約に従って通信を行う。複数の IC 間で, SCLK, MOSI, MISO は共通のバスとして利用できる。各 IC に対して, \overline{SS} ラインを設けて, これを Low にすることで, 通信先の IC を指定する。表 4 に SPI 通信を備えた IC の例を載せる。

表 4: SPI 通信機能搭載 IC の例

熱電対読み出し	MAX31855	Maxim
温度湿度気圧センサー	BME280	Switch Science
ADC	MCP3204	Microchip

3.3 RPi 計測制御基板

これまで紹介した機能を備えた計測制御基板を作り利用した。図 5 のように、10cm×20cm のマザー基板上の 40pin メスヘッダに RPi を挿す。SPI 通信可能な IC はドーター基板に実装し、マザー基板に挿して利用する。この用途に 10pin のフラットケーブルコネクタを 6 個用意した。フラットケーブル経由で、離れた場所にドーター基板を設置することもできる。この他に、GPIO 出力をリレー開閉に用いたり、PWM 制御によりファンの回転数制御を行う。

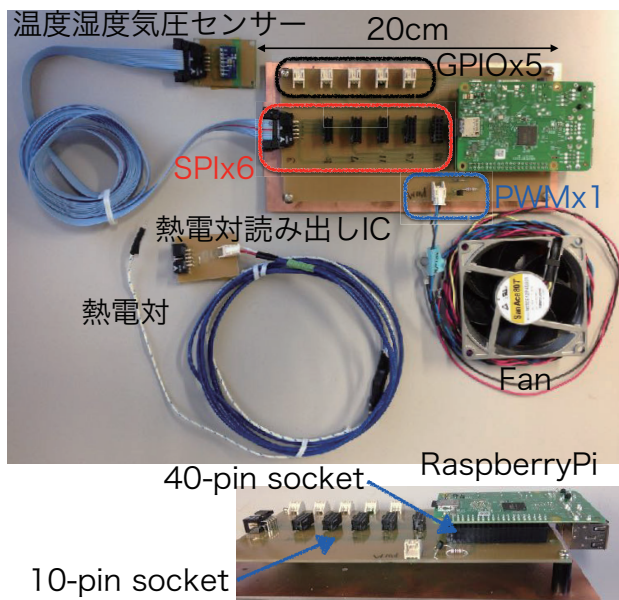


図 5: RPi 計測制御基板

以下応用例を示す。

3.3.1 チェンバガス供給システム監視

J-PARC KOTO 実験におけるガスワイヤチェンバのガス供給システムの監視である。各所の温度、可燃ガス濃度センサーの出力を SPI 通信可能な IC でモニタする。この他に、通常の USB カメラを用いバブラの監視を行う。異常の際は、パトランプ及びブザーでも警告する。KOTO 実験にて 2015 年より運用中である。

3.3.2 シリコンピクセルセンサー冷却システム

HL-LHC に向けたシリコンピクセルセンサーのテストビームにおける、センサー冷却システムのコントロール及びモニタである。センサー及びフロントエンド回路を断熱箱内に設置する。断熱箱内の冷凍機ヘッドに対し、回転数制御したファンによる内部空気循環を行い冷却を行う。冷却時結露がないよう窒素ガスフローを行い、温度湿度をモニタする。シリコンセンサーを -33°C に冷却して試験を行うことに成功した。

3.3.3 MPPC 制御システム

2018 年度に予定する KOTO 実験のカロリメータアップグレードにおいて、undoped CsI 結晶下流側の PMT に加え、上流側にも MPPC をとりつけ、シンチレーション光を読み出す。この MPPC の印加電圧の On/Off を PhotoMOS リレーで行い、暗電流出力と温度を SPI 通信機能付き IC でモニタする。こちらは別の専用基板を用いているが、紹介した RPi の機能をそのまま用いている。LXI 機能を備えた菊水電子の電源による MPPC やアンプの電圧印加も、RPi から行う。

4 J-PARC での取り組み

KEK の加速器施設が運用する加速器の多くが、EPICS をベースに開発されている。EPICS のコアとなる IOC ソフトウェアおよび通信プロトコルである CA(Channel Access) ライブラリは Posix 準拠の Unix で動作する。J-PARC センター/KEK の加速器制御グループでは、EPICS の教育用ツールとしての応用を当初の目的として、RPi や Arduino などの超小型コンピューターを EPICS で利用する環境の整備を行ってきた。加速器の制御への直接の応用はないが、気温、気圧などの環境データの取得などへの応用も検討されている。また、放射線作業中の放射線被曝量管理への応用を想定した開発なども行っている。

4.1 Arduino for EPICS training

EPICS は各種の Linux, MacOSX, Windows Subsystem for Linux などで動作する。特に Virtual Box などの仮想環境上の Linux でも動作する。このことから、EPICS の導入教育では受講者は各自の環境で EPICS ソフトウェアを動作させることができる。EPICS 教育ではこのような実習 (Hands On) が重要になる。実習の際には、実際に (LED やスイッチなどの単純な物で良いのだが) 装置を EPICS 経由で制御/監視することが有

効である。このような実習の際に簡単な入出力の装置として使えるように、Arduino をベースにしたセットを用意した (図 6) [7]。

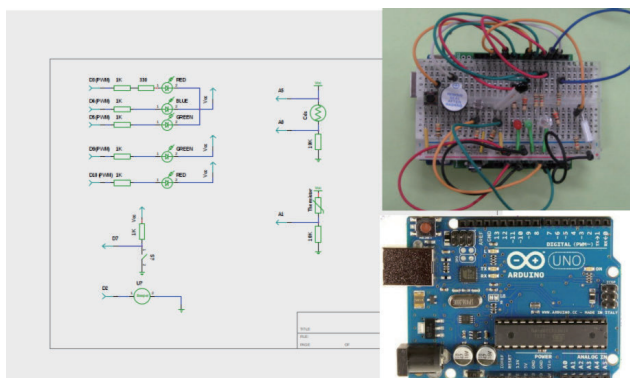


図 6: EPICS training のために用意された Arduino IO sample キット。ブレッドボード (右上) に LED, スイッチ, 温度センサー (サーミスタ), 照度センサー (CdS) などが配置 (左: 配線図) されている。これらの部品は、ブレッドボードの下の Arduino (右下) に接続されている。

Arduino は EPICS IOC となる PC と USB 接続され、EPICS IOC からはシリアル接続の装置として認識される。このシリアル接続経由で EPICS IOC から Arduino を制御するために EPICS/StreamDevice² を使った。簡単なシリアル通信上のプロトコルを定義し、これを実装する Arduino 側のプログラムと EPICS IOC 側のプロトコルファイルを開発した。

EPICS 教育ではこれらの必要な設定を提供した上で、EPICS DB の構築や、EPICS Sequencer プログラムの実習などを行う。

4.2 EPICS on RPi

RPi の CPU である ARM は以前より EPICS でのサポートに含まれており、Raspbian での EPICS 環境構築は、大きな問題なく実行できる。より簡単に Raspbian 環境に EPICS をインストールするために、Raspbian 用バイナリを含む EPICS base を tar.gz ファイルとして用意している。

RPi はユーザーが自由に使える GPIO をもつが、ADC 機能を持たない。EPICS 教育に向けて、Arduino などの ADC 機能をもつボードや、I2C/SPI などで接続可能な ADC などを用意する必要がある。Arduino の場合には、前節と同様に USB 接続で RPi 上の IOC からこの ADC 機能を使いこなせる。

²StreamDevice は EPICS で、シリアル、GPIO, VXI-11 などの接続をサポートするために汎用ライブラリである。

EPICS コミュニティの中でも RPi は評価が進んでおり、GPIO 入出力や、I2C/SPI 経由での装置制御をサポートするためのデバイスサポートが存在している [8]。

4.3 Processing 環境での EPICS 利用

Processing[9] は Java ベースのプログラム環境である。プログラミングを専門とする方でなくても画像処理などの様々な実時間処理を行うアプリケーションを作成できることを目的に開発されている。Processing 環境では、AR (Augmented Reality) マーカーの処理が簡単に利用できることから、センサーのモニター値を画像上のモニター位置に重ねて表示するプロトタイプ (図 7) を作成してみた。

CPU 負荷の分散を考慮すると、画像処理を担当する Processing 環境と、センサーからのデータを取得するプロセスに分離したい。Processing 環境で EPICS CA の入出力ができれば、このようなシステムは簡単に構築できる。EPICS CA ライブラリには互換性のある C/C++ 版と Java 版の二つがある。Processing は Java ベースであることから Java 版の EPICS CA ライブラリを Processing 環境に導入した。これにより、Processing 環境で EPICS IOC から取得したセンサーデータを、USB カメラから取得した画像上の AR マーカーの位置に表示するアプリケーションが簡単に構築できること確認している。

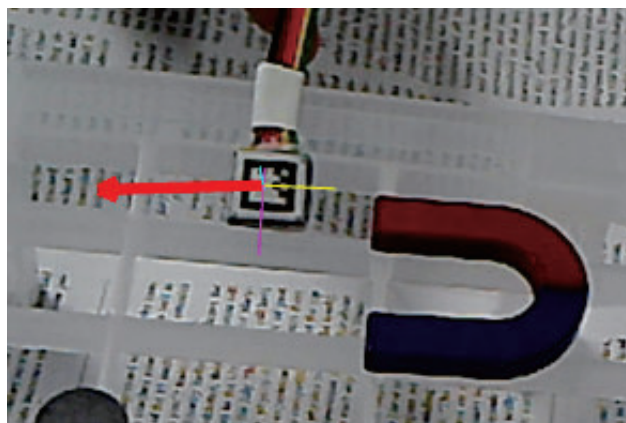


図 7: 磁気センサーの位置を AR マーカーを用いて検出。カメラ画像のその場所に、磁場をベクトルとして表示している。

5 KEK-PF/cERL での取り組み

5.1 EPICS 内蔵デバイスサーバー

小型カードサイズのシングルボードコンピューターとしては RPi が特に有名である一方、ほかにも多くの製

品が市場で入手可能となっている。KEK-PF/cEHL では、それらの中から“BeagleBone Black” (BBB) [10] を使用して RS232C 機器を遠隔から制御するための小型デバイスサーバーを開発した。これは実際に放射線のモニターや温度・湿度モニターなどに使用しており、数年にわたって安定して連続稼働している。

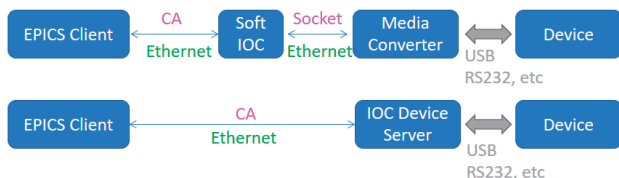


図 8: RS232C 機器を遠隔制御する際の典型的なブロック図。上:メディアコンバータを使用した場合, 下:EPICS 内蔵したデバイスサーバーでの構成。

USB 機器や RS232C 機器等のシリアル接続デバイスを Ethernet 経由で遠隔操作するとき、もっとも単純な方法は制御対象機器に Ethernet-RS232 変換器を接続し、TCP/IP Socket 通信等でコマンドをやりとりする方式であろう。メディアコンバータとして市販品も多くあり、性能や機能に応じて価格も様々な選択肢がある。制御の規模が小さい場合や、制御するホスト側が 1 つだけの場合はこれだけでも十分かもしれないが、加速器のような大規模装置、そして多くの開発者や運転員がかかわる環境では制御共通の「言葉」で通信をする必要があり、EPICS の場合は既に説明されている CA プロトコルである。図 8 上図に示すように、典型的にはサーバー計算機などにソフトウェアベースの入出力コントローラ (IOC) を置き、デバイスとの間を Socket 通信、EPICS ホストとの通信を CA プロトコルで行う構成がよく使用されている。

通常、RS232C 機器の通信速度は Ethernet 機器に比べて遅いためこのような方式でもネットワーク帯域に対して影響を与えるほどではなく、問題になることは無い。しかし、ネットワークを冗長な通信データが流れることは事実であり、今後デバイスの数が増えたり、高速通信を行うようになるとデータ通信量が増加してしまい問題になる可能性もある。そこで RS232C インターフェースを有した EPICS IOC を製作すれば遠隔制御に必要なデータのみを Ethernet 側に流すことが可能となる。最初に 1 つの BBB から 4 台のシリアル機器を制御するモデルを作成し、その後でより小型化を目指した 1 ポートモデルを製作した。図 9 に 4 ポートモデルと 1 ポートモデルの写真を示す。

RPi でもほぼ同じ機能をもつハードウェアを製作することは可能であるが、BBB は組み込みを考慮して“Cape”と呼ばれる拡張ボードの規格が定められていること、Digital I/O ピンの数が多いこと、USB や HDMI を

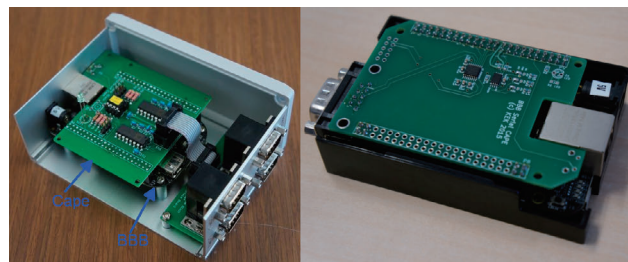


図 9: EPICS 内蔵デバイスサーバの写真。左: 4 ポートモデル, 右: 1 ポートモデル。

はじめ各種の I/O ポートがカードの短辺方向に集中して配置されていて組み込み用途に適していること、UART 4 ポートを同時に使用可能なデザインが最初から用意されていること、などの理由で我々は BBB を採用した。製品比較や導入の経緯など、詳しくは加速器学会での発表 [11] に記述している (ただし 2014 年時点での情報なので現在では少し状況が変わっている部分もある)。これらのハードウェア及び EPICS ソフトウェア情報はすべて後述する wiki に公開しているため、だれでも製作可能である。製造にあたっては Web 経由で CAD 図を送るとプリント基板を製作するサービスを利用した。基板サイズも小さいため安価かつ 1 週間程度の短納期で製造可能である。

5.2 BBB for EPICS training

EPICS 教育において、Hands-on 実習が重要であることは J-PARC の取り組みで取り上げた通りである。ここでは実習目的としての BBB について述べる。

BBB の大きな特徴として、「アナログ入力がある」ことが挙げられる。7 チャンネルという多入力数は魅力的である一方、入力電圧が 0 - 1.8 V と少し狭いことや分解能が 12 bit しかないこと、GND が共通であることなど、本格的な計測に使用するには少し見劣りするスペックとなっている。しかし、BBB 本体だけでアナログ入力が計測可能というのは特に初学者にとってとても魅力的であり、簡単に「ハードウェア」と「ソフトウェア」の相互作用を実感できるプラットフォームとして、特に教育用途に適しているといえる。最近では BBB とその派生版である BeagleBone Green (BBG) を使って EPICS 入門の講習に使用しており (図 10)、例えば

- ブレッドボードと BBB を使い LED 点滅
- EPICS から LED 制御
- Push Switch の割り込みで EPICS を動かす
- アナログ温度センサー IC を EPICS でモニター
- ノート PC に EPICS GUI を構築して制御

といった項目を基礎実習として行った。ここでは特にハードウェア割り込みで EPICS レコードをプロセスで

きることを実感してもらうことを重視している。同時に EPICS 最大のメリットである、ネットワーク分散制御が可能であることもここで体感できる。今後は実習者の習熟度に応じて I2C, SPI などを経したセンサーからの入出力を行ったり、ソフトウェア開発に興味がある場合は自分で EPICS Base/Module ソフトウェアを BBB 上に構築する演習やデバイスサポート製作実習など、要望に応じて各種のトピックスを取り上げていくことを予定している。また、実習に使用している教材や EPICS イメージの構築方法などは後述する wiki で公開している。

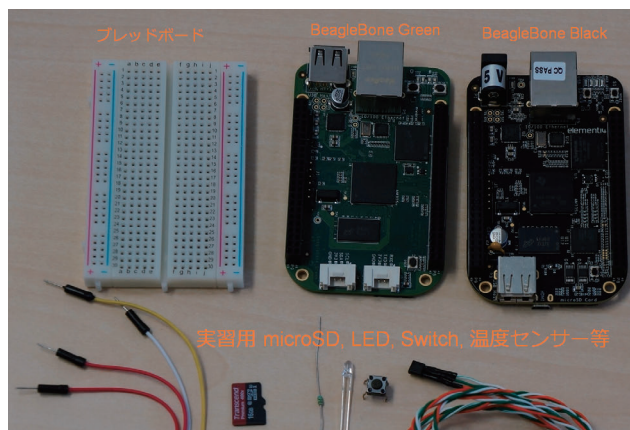


図 10: EPICS 実習用キット

5.3 日本語による EPICS 情報

従来より EPICS は国際協力として開発されてきたため、英語でのメーリングリスト [12] では活発に情報交換がなされてきた。しかし、特に EPICS 初心者や入門者にとってはいきなり英語 ML に質問を投稿することはかなり敷居が高いというのも事実であった。いきおい、周りの人や知人のツテで質問するだけになってしまっただけで情報の共有が進まないという状況が多く見られたように思う。そこで 2012 年には日本語での議論ができるメーリングリストとして“EPICS Users ML”を開設して、KEK はもちろん大学や研究機関、一般企業の方も含めて情報交換ができる環境となっている。ML で話題に上ったトピックスのほか、いろいろな開発アイテムやちょっとした Tips など、情報を共有するための wiki サイトも作成している (図 11, <http://cerldev.kek.jp/trac/EpicsUsersJP> [13])。最近では本稿で取り上げているデバイスサーバーをはじめ簡易な温度測定システムなどの製作情報も記載しているほか、EPICS 入門資料をまとめているので興味のある方はアクセスして頂きたい。ML などへのフィードバックを頂ければ幸いである。



図 11: EPICS 日本語情報 wiki サイト

6 EPICS と RPi について

6.1 EPICS 初心者の皆さんに向けて

EPICS で加速器制御システムを開発した KEK や J-PARC では、過去に EPICS について尋ねられたり、講習会を行ったりしてきた。EPICS の Document は大量にある。ただし英語の reference manual 的なものが多く、初心者は膨大な文書の中から何を最初に読むべきかわからない。その中では、過去に開催された講習会の資料 (例えば [14]) が適当と思われるが、初心者の独習には敷居が高いかもしれない。年に 2 回ある EPICS collaboration meeting [15] の講習会に参加できれば申し分ないが、国内開催はなかなか回ってこない。

EPICS は様々なツールの集合体である。レゴに例えればいろいろな形のブロックである。制御や DAQ を EPICS で開発することは、そのブロックで車を作るかお城を作るか、という話に相当する。ツールには複数の選択肢があり、例えば GUI editor では、medm/edm, CSS-BOY, Qt-base などお絵かき系から、SAD [16] や Python-base など script 系もある。自分の目的にはどれを選ぶべきか理解が足りないまま開発に着手すると、EPICS-base でもバランスを欠いた変なものが出る。ガイドとなる情報 (特に日本語) が無いのが現時点の問題である。J-PARC 建設初期 (2004 年頃)、部隊は JAEA 職員 (加速器の経験が無い人が主力) と KEK-PS 職員が多かったが、いずれも EPICS の経験は無かった。当時 KEKB には EPICS 経験者が多数いたが、J-PARC にはあまり異動しなかった。そこで、J-PARC 関係者を念頭に EPICS 教育用資料「miniEPICS」の整備を進めた [17] が、私 (上窪田) が J-PARC 建設で忙殺され、担当者も異動し、残念ながら完遂しなかった。しかしこの時の資料は、2012 年 12 月 (インド) や 2017 年 5 月 (京大原子炉) の EPICS 講習会で再構成され利用された [4]。

6.2 最近の EPICS 事情と RPi

EPICS は、国内では KEK, J-PARC, RIBF, KURRI などの加速器制御に導入されている。世界では多数の加速器で導入されているが、新規の大型計画でも良く採用される (例えば ESS や FRIB や CSNS など; CERN は独自路線)。最近では、加速器以外への EPICS 導入が目を見張る。ITER (核融合) が EPICS 導入を決め、関連する QST-青森 (IFMIF) や QST-那珂へ EPICS が波及している。また、LIGO (重力波) の最上位部分 (Guardian) が EPICS で構築されており [18], KAGRA でも EPICS が導入されている。QST や KAGRA の EPICS 状況は、2017 年 5 月の EPICS meeting に報告がある [4]。

国内で加速器以外を含め EPICS が広まっているが、支える人材は少ない。短期的には外国の EPICS 経験者を入れるしかなさそう。長期的には、KEK や J-PARC などの EPICS 経験者が講習会を開催し、国内で EPICS 体験者を増やす努力をすべきであろう。

さて、本稿でも紹介があるように、RPi に EPICS base を実装すること自体は難しくはない。研究・教育用に、RPi は手軽で低コストな EPICS system になる。また、世にあふれる IoT device を RPi で EPICS 化すれば、EPICS の様々な汎用ツール (GUI, Archiver, ..) が利用できる。多数の IoT device (EPICS 化した RPi) を DAQ したり連携させたりするのは、EPICS の得意技である。RPi と IoT device で EPICS 講習会の教材を作り、講習会を開催し、EPICS 経験者を増やしていく。今後の活動は、この方向に収斂していきそうである。

7 まとめ

本稿では、RPi や BBB といった、小型シングルボードコンピュータを用いた測定器データ収集システムの実例を紹介した。この例では、Python, C, あるいは EPICS を用いた機器制御を行っている。最近の市販小型シングルボードコンピュータは、低価格で非常にコンパクトでありながら、教育や研究目的で使用可能な機能や性能を備えている。今後も様々な用途に使用できることが期待される。

参考文献

- [1] Raspberry Pi <https://www.raspberrypi.org/>
- [2] Arduino <https://www.arduino.cc/>
- [3] EPICS <https://epics.anl.gov/>
- [4] Spring 2017 EPICS Collaboration Meeting <http://www.rri.kyoto-u.ac.jp/EPICS/>

- [5] E. Upton *et al.*, "Raspberry Pi ユーザーガイド", インプレスジャパン, 2013 年.
- [6] <http://osksn2.hep.sci.osaka-u.ac.jp/~nanjo/dokuwiki/doku.php?id=scopeman:scopeman>
- [7] "Arduino as a Tool for EPICS training", Noboru Yamamamoto, https://epics.anl.gov/meetings/2012-10/program/1024-A1_Arduino_as_tool_for_EPICS_training.pdf, EPICS collaboration meeting 2012.
- [8] Florian Feldbauer, epics-devgpio, <https://github.com/ffeldbauer/epics-devgpio/blob/master/README.md>, (2016)
- [9] Processing Home Page, <https://processing.org/>; "Processing: A Programming Handbook for Visual Designers and Artists (Second Edition)", Casey Reas and Ben Fry, The MIT Press(2014).
- [10] BeagleBoard.org <https://beagleboard.org/black>
- [11] 帯名 崇, 路川 徹也, "EPICS 内蔵小型デバイスサーバの開発", 加速器学会年会 (2014) MOOL08 https://www.pasj.jp/web-publish/pasj2014/proceedings/PDF/MOOL/MOOL08_oral.pdf
- [12] EPICS Tech-Talk <https://epics.anl.gov/tech-talk/>
- [13] EPICS Users JP (日本語 EPICS 情報) <http://cerldev.kek.jp/trac/EpicsUsersJP>
- [14] <https://epics.anl.gov/docs/USPAS2014.php>
- [15] <https://epics.anl.gov/meetings.php>
- [16] SAD Home Page <http://acc-physics.kek.jp/SAD/>
- [17] 上窪田紀彦, 他 "小規模系 EPICS パッケージ (miniEPICS) の開発", 加速器学会 (2005) pp.211-213.
- [18] "Status of the Advanced LIGO Project", talk at ICALEPCS 2015 http://accelconf.web.cern.ch/AccelConf/ICALEPCS2015/talks/moa3o03_talk.pdf