

第31回 日本ロボット学会 学術講演会

2013年9月4日(水)～6日(金)

THE 31st
ANNUAL CONFERENCE
OF THE ROBOTICS SOCIETY OF JAPAN

講演概要集



主催：一般社団法人 日本ロボット学会

RaspberryPi用拡張入出力ボード：PiRT-Unit

－ RT ミドルウェア教育用途への適用 －

○安藤慶昭 (産総研), 関山守 (産総研), 神徳徹雄 (産総研),
深澤篤史 (ウィン電子工業), 片見剛人 (ウィン電子工業)

1. はじめに

近年、コンピュータ等の技術の発達により、以前は高度な知識が要求されたり個人レベルでは困難であったことが、PIC/AVR等のマイコン、Arduinoなどのマイコンボード等の出現によりアイデアとソフト・ハードに関する多少の知識があれば誰でも実現できるようになりつつある。最近では、安価な組み込み Linux ボード RaspberryPi がその価格とサイズ、GPIO等の拡張性により注目を集めており、愛好家による様々な作品が製作されたり、これを組み込んだ小型ロボット [1] 等が発表されるなど応用が広がっている。

本稿では、RaspberryPiを題材として、プログラミングとセンサ・アクチュエータ等ハードウェアの2つの領域を同時に学習するための拡張ボード PiRT-Unit を製作し、ソフトウェアからハードウェアにまたがる広範な知識とスキルが求められるロボット工学をはじめとする電気・機械・情報融合分野、高エネルギー物理・先端計測分野などへの教育へ応用することを提案する。

PiRT-Unit は RaspberryPi の GPIO 等の入出力を拡張し、AD, DA, シリアルポート, ZigBee¹等の入出力ポートを設け、Linux 上から容易に扱えるようにした拡張入出力ボードである。教材として利用するため、いくつかのセンサ・ZigBee モジュール等とセットにし、RT ミドルウェアを利用したロボットプログラミングを学習する題材を作成、これを、ROBOMEC2013 の RT ミドルウェア講習会において実際に受講者に使用してもらったのでその結果と考察を示す。

2. RaspberryPi

Raspberry Pi とは、英国のラズベリーパイ財団²が開発した ARM プロセッサを搭載したシングルボードコンピュータである。RaspberryPi の仕様を表 1 に示す。

Raspberry Pi は組み込みボードサイズにも関わらず、ARM 用の通常の (μ CLinux 等組み込み用ではない) Linux や FreeBSD が動作し、ボード上でのセルフコンパイルも可能である。組み込みボード開発で通常必要なファームウェアの書き換え作業や、プログラムのアップロード作業も必要なく [2]、開発者にとって利用しやすい組み込みボードである。ボード自体も \$30 程度と安価であることに加えて、ストレージデバイスとして近年大変安価かつ大容量化している SD カードを利用できるため、通常の Linux と同じ感覚で気軽に開発が可能である。

¹近距離無線通信規格の一つ (IEEE802.15.4)

²<http://www.raspberrypi.org/>

表 1 RaspberryPi 仕様

RaspberryPi (Model B) 仕様	
価格	\$35
SoC	Broadcom BCM2835
CPU	700 MHz / ARM1176JZF-S
GPU	Broadcom VideoCore IV
SDRAM	512MB (GPU 共有)、
USB 2.0	2 ポート
映像出力	コンポジット RCA (PAL & NTSC), HDMI (rev 1.n3 & 1.4)
音声出力	3.5 mm ジャック, HDMI
ストレージ	SD/MMC/SDIO スロット
ネットワーク	10/100 Mbps (RJ45)
I/O	GPIO×8, UART, I ² C, SPI +3.3V, +5V, GND
電源	700 mA (3.5 W)
サイズ	85.60 mm × 53.98 mm
OS	Debian, Fedora, Arch Linux, FreeBSD

また、GPIO, I²C³, SPI⁴, UART 等基本的な IO が提供されており、外部の様々なデバイスとも接続できるため、ロボット制御やセンサによる計測など、いろいろな応用が考えられる。

3. PiRT-Unit

RaspberryPi には基本的 IO が提供されていることは上述したが、実際にセンサやアクチュエータなどを接続する場合には、AD/DA コンバータ等を I²C や SPI 経由で接続する必要がある。UART も RS232C と接続するためにはレベル変換 IC を経由しなければならない、電子回路等の初心者には敷居が高い。こうした外部との入出力デバイス等を容易に接続するために、AD/DA、シリアルポートなどを提供する RaspberryPi 用拡張ボード: PiRT-Unit を設計・製作した。PiRT-Unit の外観および入出力配置図を図 1 に示す。

3.1 基本設計

PiRT-Unit は小型のネットワーク接続型 IO デバイス small RT-Unit [3] の考え方を RaspberryPi へ適用した、RaspberryPi 用 IO 拡張基板ボードである。PiRT-Unit の諸元を表 2 に示す。

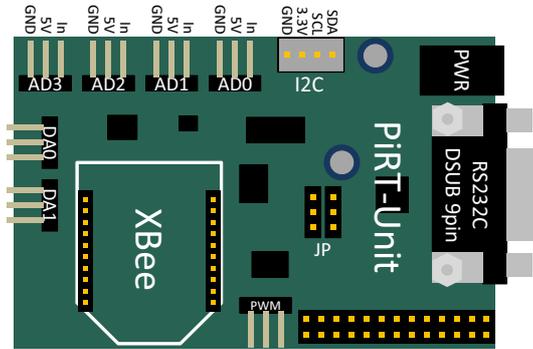
以前製作した small RT-Unit ではマイクロプロセッサとして PIC または dsPIC を使用していたため、別

³Inter-Integrated Circuit (I-squared-C: アイ・スクエア・シー)、フィリップス社が提唱した 2 線式のシリアルバス規格。

⁴Serial Peripheral Interface、モトローラ社が提唱した 4 線式のシリアルバス規格。



(a) PiRT-Unit外観



(b) PiRT-Unit端子配置図

図 1 PiRT-Unit:入出力配置図および外観

途 PC 上のプロキシ RTC を必要とする RTC-Lite[4] としてしか RTC を動作させることができなかつた。一方 PiRT-Unit は CPU ボードとして RaspberryPi を使用するため、通常の RTC が利用できるだけでなく、ボード上で RTC の開発も行うことができる。入出力に関して、AD(4ch)/DA(2ch) のみならず、RS232C/ZigBee モジュール、PWM、I²C および RaspberryPi へも同時供給可能な電源コネクタが用意されている。

表 2 PiRT-Unit 仕様
Raspberry Pi 拡張 IO ボード

AD コンバータ	10bit, 4ch, 200S/s
DA コンバータ	12bit, 2ch
PWM 出力	1ch, RC サーボモータ用
RS232C	D-SUB 9pin コネクタ Xbee とジャンパにて切り替え
Xbee	Xbee 接続コネクタ
電源入力	5V DC 入力 Raspberry Pi に電源供給可能 Raspberry Pi からの電源供給でも動作

3.2 ADC/DAC と Phidget センサ

PiRT-Unit には 4ch の AD コンバータ (ADC) および 2ch の DA コンバータ (DAC) が搭載されている。ADC 端子は、5V, GND および、両者の分圧値がアナログ入力ピンへ接続されている。図 2 に示すように半固定抵抗等の受動回路も接続可能である。

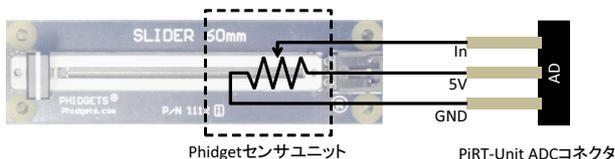


図 2 ADC とセンサの接続

AD コンバータ端子のピンアサインは Phidget センサ群と同一のピンアサインとなっている。Phidget とは Phidgets Inc.⁵ から発売されている、PC に接続可能な

⁵<http://www.phidgets.com/>

IO 拡張ボードとセンサ群・アクチュエータ群等の製品の総称である。電子工学を学習する人やメディアアート、プロトタイピングなどに利用されている (図 3)。

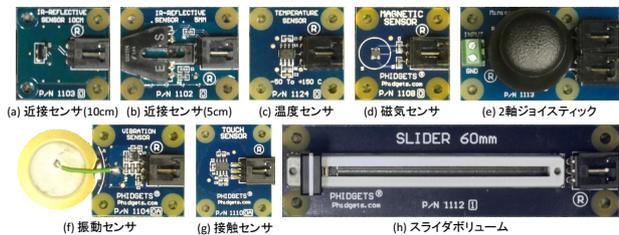


図 3 Phidget センサ群

電子回路の初心者や、ソフトウェアからの外部 IO デバイスのプログラミングを学習したい人、入出力デバイスを利用しメディアアート等の作品を作成したい人にとっては、センサの選定、基板の作成・配線などは専門知識・時間などの面からも敷居が高い。Phidget の様な既存のセンサモジュールを利用することで、様々な種類のセンサを容易に試すことができ、プログラミング学習にかかる時間を長くとることができる。

ADC や DAC からのデータの読み出し、書き込みは C 言語等コンパイル言語以外にも Python 等スクリプト言語からも利用できるため、プログラミング初心者にとっても利用しやすい (図 4)。

```
import spidev
spi = spidev.SpiDev()
spi.open(0, 0)
sned_ch = [0x00, 0x08, 0x10, 0x18]
r = self.spi.xfer2([sned_ch[0], 0, 0, 0])
voltage = ((r[2] << 6) & 0x300) |
          ((r[2] << 6) & 0xc0) |
          ((r[3] >> 2) & 0x3f)
```

図 4 Python による ADC へのアクセス例

3.3 シリアルポートと XBee

PiRT-Unit にはシリアルポート (RS232C) および ZigBee モジュールの一種である XBee を利用することができる。両者は RaspberryPi の同一の UART に接続されており、基板上のジャンパピンにより排他的に選択できるようになっている。

デフォルトではシリアルポートは Linux のシリアルコンソールとして設定されており、ターミナルソフトを接続することでログイン、操作を行うことができる。ペアリングした XBee モジュールを PiRT-Unit および PC の両者に接続することで、コンソールを無線化することも可能である。さらに、シリアルコンソールを無効にすることで、単なるシリアルポートとして利用し、例えば多数の XBee モジュールを制御するコーディネータとして設定するといった利用も考えられる。

3.4 その他

このほか、ラジコンサーボを直接制御可能な PWM コネクタおよび、センサやメモリなどマイコン周辺チップが多数採用しているシリアルバス規格の I²C 用のコネクタを 1ch 搭載している。

さらに、PiRT-Unit には RaspberryPi 側への電源供給可能な電源入力コネクタが搭載されている。基本的に PiRT-Unit は RaspberryPi からの電源供給のみで動作する。RaspberryPi は最大 700mA の電流消費量であるが、USB microB コネクタ用 AC アダプタでは通常最大 1A 程度の容量しか供給できない。したがって、PiRT-Unit 上の負荷回路に供給可能な電力容量は 300mA 程度の余裕しかない。このため、一般的な DC ジャック (2.1mm 標準 DC ジャック, 内径 2.1mm 外径 5.5mm) で大容量の AD アダプタによって両者に電力供給可能な構造とした。

4. 教材としての活用

4.1 学習キット

PiRT-Unit を利用し、Linux による IO プログラミングと RT コンポーネントの作成方法を学ぶための学習キット (図 5) を作成した。内容は以下の通りである。

- RaspberryPi + PiRT-Unit
- Phidget Sensor Kit #1 または #2
- USB シリアル変換ケーブル
- XBee モジュール ×2 USB エクスプローラ
- USB 無線 LAN ドングル
- AC アダプタ (5V 2.3A), USB microB ケーブル

Phidget センサキットおよび開発に必要な種々の機材を同梱した。RaspberryPi の Linux ディストリビューションは標準の Wheezy を採用した。名前による IP アドレス検索と ssh によるログインが利用できるよう mDNS (マルチキャスト DNS の一種) の Linux 実装である avahi と sshd をあらかじめ設定している。さらに、USB 無線 LAN についても設定済みであるが、アクセスポイント等の設定はいったん有線 LAN で設定の上利用する必要がある。

このほか、OpenRTM-aist (C++版, Python 版) や依存ライブラリ、SPI・I²C などへアクセスするためのコマンド、ライブラリ、Python モジュール等を予めイ



図 5 RT ミドルウェア PiRT-Unit 学習キット

ンストールしており、起動後すぐに IO プログラミングを始められる環境となっている。

実際の開発は、ssh によるリモートログインで行うか、RS232C または XBee を利用したシリアルコンソール経由で行うかを選択できるようになっており、別途 HDMI モニタやキーボード・マウスを用意する必要はない。

4.2 講習会での利用

Phidget の 2 軸ジョイスティックセンサを利用し、移動ロボットを操作する例を作成した。Phidget Ministick センサ (図 6 (a)) は X-Y 各軸の変位を 2ch のアナログ値で出力するセンサである。PiRT-Unit の ADC0 および ADC1 に接続し (図 6 (b)), Linux からこれらに対応する電圧値 (0.0~5.0V, ニュートラル時約 2.5V) を読み込みジョイスティックとして利用することができる。

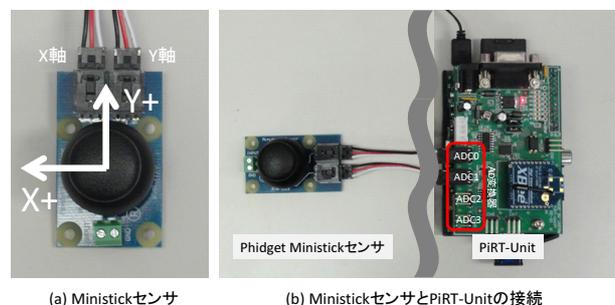


図 6 Ministick センサによるジョイスティック

ジョイスティックの操作対象として、市販の移動ロボット (Yujin Robotics 社, Kobuki) を利用した。読み取った X-Y の値から、移動ロボットの 2 次元速度ベクトルや車輪の角速度を出力する RT コンポーネントを作成し、Kobuki を操作する RT コンポーネントを作成する作業を学習題材と設定した。

この課題では、

- ジョイスティックのキャリブレーション (アナログセンサの扱い方)
- ADC からのセンサデータの取得と X-Y 座標への変換

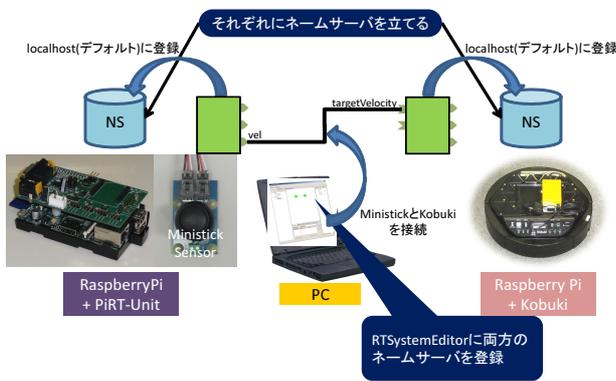


図7 Kobuki と PiRT-Unit を利用したシステム

- X-Y 座標から速度への変換 (および移動ロボットの運動学)
- OutPort を 1 つ持つ RT コンポーネントの作り方を学習する。

ジョイスティックの X-Y の電圧値 (V_x, V_y) から 2 次元速度ベクトル (v_x, v_y, v_a) への変換は、変換係数を K_x, K_a として

$$\begin{aligned} v_x &= K_v V_y & [\text{m/s}], \\ v_y &= 0 & [\text{m/s}], \\ v_a &= -K_a V_x & [\text{rad/s}], \end{aligned}$$

で表すことができる。また、2 輪独立駆動式移動ロボットの車輪角速度 (w_l, w_r) への変換は係数を K として、

$$\begin{aligned} \theta &= \arctan\left(\frac{V_y}{-V_x}\right), \\ w_l &= K \cos\left(\theta - \frac{\pi}{4}\right) & [\text{rad/s}], \\ w_r &= K \sin\left(\theta - \frac{\pi}{4}\right) & [\text{rad/s}], \end{aligned}$$

のように表すことができる。車輪角速度 (v_l, v_r) から運動学を用いて 2 次元速度ベクトル (v_x, v_y, v_a) への変換を導出する課題等も考えられる。

4.3 ROBOMECS2013 チュートリアル

2013 年 5 月 22 日に機械学会ロボメカ部門講演会 ROBOMECS2013 において行われた RT ミドルウェアチュートリアルにおいて、4 人一組のグループで上記の課題を解決する実習形式で講習会を行った。

問題点として

- Linux のコマンドラインでの作業に慣れていない
- X-Y から速度への変換係数の設定に苦労していた
- 初期化時のキャリブレーションが不十分でニュートラル時も移動ロボットが低速で動いていた

等があげられる。一方で Python によるプログラミングは初心者でもわかりやすく、コンパイルも不要なのでトライ&エラーが行いやすい、X-Y 座標から 2 次元速度ベクトルへの変換式の導出は問題なく行っていた、などの傾向が見られた。



図8 講習会での PiRT-Unit の利用 (ROBOMECS2013 RTM チュートリアル)

5. おわりに

本稿では、RaspberryPi 用 IO 拡張基板 PiRT-Unit を設計・製作し、センサ等外部デバイスと Linux 上でのプログラミング学習に利用することを提案した。

RaspberryPi の GPIO を利用し、AD, DA, I²C, PWM, RS232C/XBee 等が利用できる基板 PiRT-Unit を設計製作した。センサとして市販の Phidget センサ群を利用し、学習キットを製作、RT ミドルウェア講習会において移動ロボット Kobuki とともに利用した。

単にセンサデータを読み取るだけでなく、そのデータを利用し実際にロボットを操作するという内容は受講者の興味を高め、理解を促進するのに役立った。ただし、基本的な Linux の操作方法を理解していない点などについては今後基本的なチュートリアルを追加するなどして改善する予定である。また、ロボットを動かす際に必要な雑多な知識 (パラメータの設定方法) などは RT ミドルウェアを活用して理解を助ける手法を考案する。

参考文献

- [1] RAPIRO, <http://www.rapiro.com/>
- [2] 安藤 慶昭, "組込み機器用 RT コンポーネント開発環境 ATDE for OpenRTM-aist", 計測自動制御学会 システムインテグレーション部門 講演会 2012 (SI2012), pp.543-547, 2012.12
- [3] 安藤 慶昭, 小川秀明, "RT ミドルウェアのメディアアートへの応用-パーソナルなコミュニケーションメディア: SmallConnection-", 日本機械学会ロボティクス・メカトロニクス講演会 2005, p.1P2-S-038, 2005.06
- [4] Noriaki Ando, et al., "RTC-Lite: Lightweight RT-Component for Distributed Embedded Systems", SICE Journal of Control, Measurement, and System Integration (SICE JCMSI), Vol. 2, No.6, 2009.11