

マイコン  
活用シリーズ

ARM

Wi-Fi/Bluetooth/ZigBee無線用

# Raspberry Pi プログラム全集

今すぐ  
IoT  
製作

Complete collection of Raspberry Pi software programs for Wi-Fi/Bluetooth/ZigBee

国野 亘 著

インターネットと電子回路を  
ワイヤレスで直結!



ご購入はこちら。  
<http://shop.cqpub.co.jp/hanbai/books/42/42231.htm>

Web制御ロボットや  
無人モバイル作りに

CQ出版社



付属CD-ROM

全50!

動作確認済みの  
サンプルプログラム  
を収録

見本

## 第1節 Raspberry Pi と Linux について

Raspberry Pi(ラズベリー・パイ)は、英国 Raspberry Pi Foundation(ラズベリー・パイ財団)が開発した安価なマイコン・ボード・コンピュータです。クレジット・カードとほぼ同じサイズの基板にも関わらず、USB キーボードと USB マウス、そしてテレビもしくは PC 用モニター(ディスプレイ)等に接続することで、学習用パソコンとして使用することができます。

小型の学習用パソコンといっても、インターネット閲覧やビデオ再生、ワープロ、表計算、ゲームといったパソコン並みの能力をもっています(図 1-1)。

2015 年 2 月に発売された Raspberry Pi 2 Model B には、スマートフォンなどで使用されている ARM Cortex-A シリーズのアプリケーション・プロセッサ Cortex-A7 Quad Core 900MHz が搭載されました。さらに Raspberry Pi 3 では 64 ビットの Cortex-A53 が採用され、処理能力が上がりました。

一般的なパソコンでは、Microsoft Windows や Mac OS X といった OS が動作しており、その上でアプリケーション・ソフトが動作します。Raspberry Pi には、複数の Linux ベースのディストリビューションがリリースされており、それらの中からインストールする OS を選択し、その上でアプリケーション・ソ

フトを動作させることができます。

本書では、Raspberry Pi でよく使われている Raspbian と呼ばれる Linux ベースの OS を使用します。Raspbian は、Raspberry Pi ユーザーの有志達が Linux ベースの Debian を Raspberry Pi のハードウェア向けにカスタマイズした OS です。もちろん、無料で利用することができます。Raspberry Pi Foundation が公式にサポートを行っており、Raspberry Pi 標準 OS と呼んでも良いでしょう。

さて、Linux ベースの OS と聞いても馴染みのない人もいると思います。Linux は、スマートフォンの Android のカーネル部<sup>1</sup>に採用されているなど、とても身近に存在する OS です。古くはスマートフォンの基となった PDA(写真 1-2)でも使われてきました。これらから、モバイル機器の IT 化にもっとも寄与した OS と言えるでしょう。

また、クラウド・サーバやスーパーコンピュータなどにも用いられています。かつてミニコンやワークステーションといったサーバ機器には、UNIX が使われていました。その後、UNIX 技術を多くの人が利用できるように開発された Linux が普及し、現在では、Linux が IT インフラを支えるサーバ用の OS として

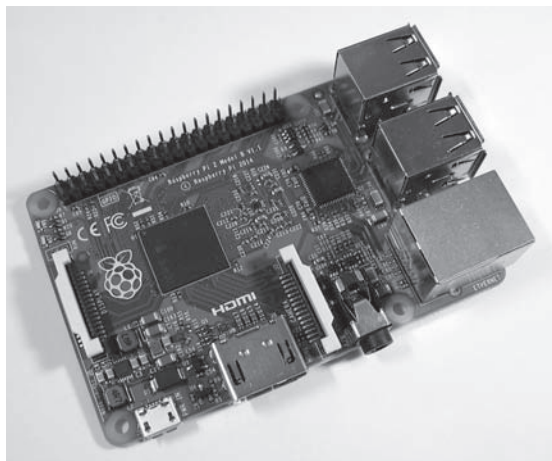


写真 1-1  
Raspberry Pi 2 Model B

<sup>1</sup> カーネル：OS の中核となるハードウェア・リソースを管理するソフトウェア部。

## 第1節 覚えておかなければならないLinuxコマンド① ls

Raspbianは、Raspberry Pi上で動作するLinuxをベースとしたOSです。LXTerminalからLinuxコマンド(bashコマンド)を入力することで、Raspbian内のLinuxカーネルにアクセスすることができます。本節以降、このLinuxコマンドについて説明します。

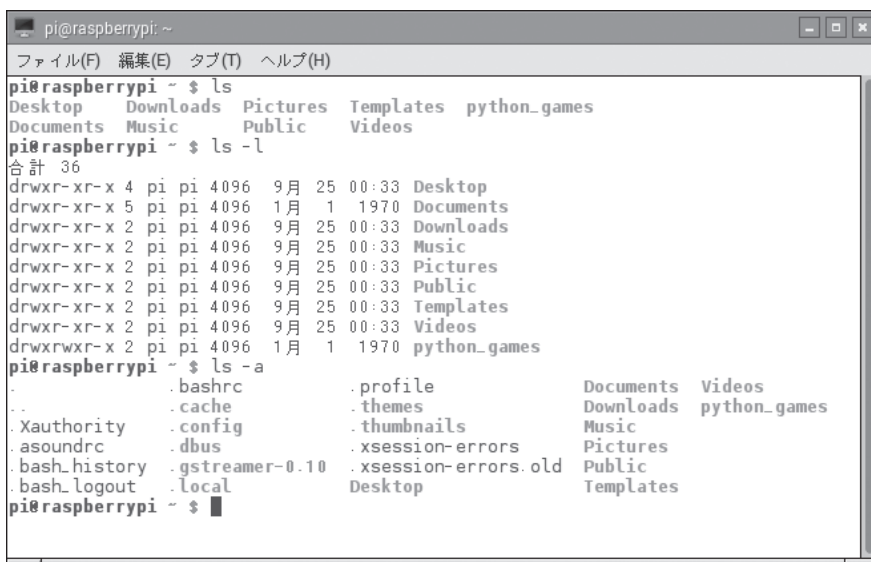
多くのLinuxの専門書では、すべての作業をLinuxコマンドで行えるほどの詳細な説明が書かれています。これはCLIと初期のGUI<sup>10</sup>とを比べたときに、CLIのほうがOS上の操作を効率的に行うことができたからです。しかし、現在はグラフィックならではの表現力によって、操作を判断するまでの時間や操作ミスの防止が可能であり、開発者のGUI環境への移行

も進んでいます。ここではGUI操作と併用することを前提として、必要最小限度のLinuxコマンドを学習します。

フォルダ内のファイル等を確認するには「ls」(List Segment)コマンドを使用します。図1-13(p.22)のアイコンをクリックしてLXTerminalを起動し、「L」キーと「S」キーで小文字の「ls」を入力し、「Enter」キーを押下すると、図2-1のようにフォルダ内のファイルやサブフォルダ<sup>11</sup>が表示されます。

```
$ ls ↵
```

「ls」に続けて「スペース」キー、「-(マイナス)」キー、「L」キーを順に押下し、「ls -l」と表示されるのを確認



```
pi@raspberrypi ~ $ ls
Desktop  Downloads  Pictures    Templates  python_games
Documents Music      Public     Videos

pi@raspberrypi ~ $ ls -l
合計 36
drwxr-xr-x 4 pi pi 4096  9月 25 00:33 Desktop
drwxr-xr-x 5 pi pi 4096  1月  1 1970 Documents
drwxr-xr-x 2 pi pi 4096  9月 25 00:33 Downloads
drwxr-xr-x 2 pi pi 4096  9月 25 00:33 Music
drwxr-xr-x 2 pi pi 4096  9月 25 00:33 Pictures
drwxr-xr-x 2 pi pi 4096  9月 25 00:33 Public
drwxr-xr-x 2 pi pi 4096  9月 25 00:33 Templates
drwxr-xr-x 2 pi pi 4096  9月 25 00:33 Videos
drwxrwxr-x 2 pi pi 4096  1月  1 1970 python_games
pi@raspberrypi ~ $ ls -a
.          .bashrc          .profile         Documents      Videos
..         .cache           .themes         Downloads     python_games
.Xauthority .config          .thumbnails     Music
.asoundrc  .dbus            .xsession-errors Pictures
.bash_history .gstreamer-0.10 .xsession-errors.old Public
.bash_logout .local           Desktop         Templates
pi@raspberrypi ~ $
```

図2-1 Linuxコマンド「ls」の実行結果の一例

表2-1 Linuxコマンド「ls」のおもな機能

コマンド	機能
ls	フォルダ内のファイルやサブフォルダを表示する
ls -l	フォルダ内のファイルやサブフォルダを一覧表示する
ls -a	フォルダ内を隠しファイルを含めて表示する
ls -la	隠しファイルを含めて一覧表示する(「ls -l -a」でも可)

<sup>10</sup> GUI: グラフィカル・ユーザ・インターフェース=グラフィックとマウス等によるUI.

<sup>11</sup> サブフォルダ: フォルダ内にあるフォルダ.

見本

## 第1節 コマンドUIの定番プログラム「Hello, World!」を表示しよう

パソコンのプログラム演習において、一番初めに学ぶ定番のプログラムが「Hello, World!」です。本節では、プログラムの作成方法やコンパイルの方法、実行方法について説明します。

まずは、練習用プログラムをダウンロードしてください。キーボードに慣れていない人は本書に書かれているプログラムを自分で入力してみても良いですが、ダウンロードしたものを基にして、処理の流れをつかむことも重要です。

ダウンロードするには、LXTerminal を起動し、下記のコマンドを入力します。大文字と小文字も間違えないようにしてください。付属のCR-ROMや、筆者のサポート・ページ<sup>14</sup>からダウンロードすることもできますが、git コマンドによるダウンロードを奨めます。

```
$ git clone https://github.com/
  bokunimowakaru/RaspberryPi.git ↵
```

### 練習用プログラムのダウンロード (GitHub)

```
https://github.com/bokunimowakaru/
RaspberryPi.git
```

上記の git コマンドを使ってダウンロードすると、「RaspberryPi」フォルダが作成されます。また、その中の「practice」フォルダ内に練習用のプログラムが格納されます。拡張子に「.c」が付与されているファイルがC言語のプログラムです。

それでは、練習用のプログラム 3-1「practice01.c」

を、テキスト・エディタ「Leaf Pad」で開いてみましょう。File Manager を使う場合は、「practice」フォルダ内の「practice01.c」を右クリックし、「アプリケーションで開く」を選択し、図 2-4 (P.38) の画面で「Text Editor」を選択します。

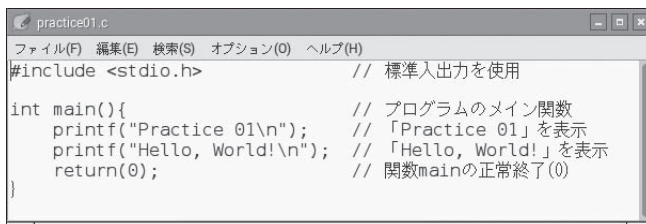
ダウンロードしたプログラムを Leaf Pad で開いたときの画面の一例を、図 3-1 に示します。このプログラムを編集することもできますが、まずはコンパイル<sup>15</sup>方法から説明します。

ダウンロード時に開いた LXTerminal に以下のコマンドを入力して、practice フォルダへ移動します。補完機能を利用する場合、「C」キー、「D」キー、「スペース」キーを順に押して小文字の「cd」まで入力し、「Shift」キーを押しながら「R」キーを押して大文字の「R」を入力、「Tab」キーを押して「RaspberryPi」に補完します。この段階では「Enter」を押さずに、「P」キー、「R」キーを押して小文字の「pr」と入力してから「Tab」キーを押して「practice」に補完し、「Enter」を押します。

```
$ cd R [Tab]
→ $ cd RaspberryPi /
→ $ cd RaspberryPi /pr [Tab]
→ $ cd RaspberryPi /practice / ↵
```

「practice」フォルダに移動したら、念のために「ls」コマンドを使ってプログラム「practice01.c」などが入っていることを確認します。そして、下記のコマンドを実行します。

```
$ gcc practice01.c ↵
```



```
#include <stdio.h> // 標準入出力を使用

int main(){ // プログラムのメイン関数
    printf("Practice 01\n"); // 「Practice 01」を表示
    printf("Hello, World!\n"); // 「Hello, World!」を表示
    return(0); // 関数mainの正常終了(0)
}
```

図 3-1  
練習用プログラム practice01.c

<sup>14</sup> 筆者のサポート・ページ：<http://www.geocities.jp/bokunimowakaru/cq/raspi/>.

<sup>15</sup> コンパイル：プログラムを実行可能な形式に変換すること。

見本

## 第1節 プロトコル・スタック搭載ワイヤレス通信モジュール

ワイヤレス通信の実験やワイヤレス通信を応用したアプリケーションを作る場合、高周波回路の設計や通信プロトコル・スタック<sup>19</sup>の実装が必要です。ワイヤレス通信用ICの中には、これらの機能をワンチップ化してソフトウェアとともにICに搭載しているものがあります。

それでも、アンテナとICを接続する回路には高周波回路設計が必要で、電波を送信する場合は、後述の技適を受ける必要もあります。また、通信プロト

コル・スタックが実装されていたとしても、そのAPIが使いやすくなければ、簡単にアプリケーションを開発することはできません。

そこで本書では、簡単にワイヤレス通信の基礎実験や応用・活用できるように、これら必要なすべての要素が含まれた通信モジュール(図4-1)を使用します。また、これらの活用に必要な技術内容と豊富なサンプル・プログラムを紹介します。

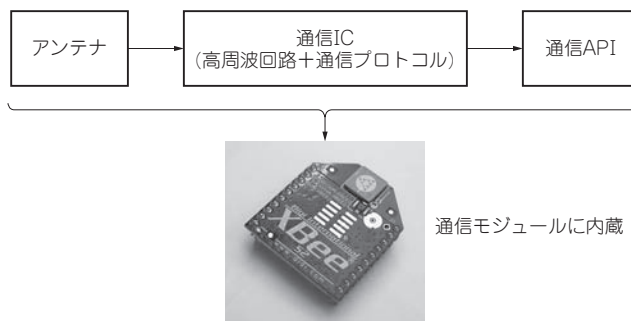


図 4-1  
プロトコル搭載 XBee モジュールの例

## 第2節 技適や認証の取得済みモジュールでしか送信してはならない

日本国内で電波を送信する場合は、少なくとも、技適<sup>20</sup>または工事設計認証などを受け、電波法令で定められている技術基準に適合した機器を使用する必要があります。これらの機器には、郵便局を表す〒のシンボルが入った技適マーク(図4-2)が表示されています。

ワイヤレス通信モジュールのうち、技適や認証を受けていない製品は、日本国内で電波を送信することができません。また、通信モジュールのアンテナを指定外のものに交換した場合や、モジュール内の回路を改変した場合は、技適を受け直す必要があります。例え

ば、XBee モジュールの中にはアンテナが交換できるタイプ(RPSMA タイプやU.FL タイプ)がありますが、指定外のアンテナを使用する場合は、技適の再受検が必要です。

また、2012年8月以前は、はんだ付けを前提としたモジュールの技適や認証が認められていなかったため、基板との接続用コネクタのない通信モジュールの認証が得られませんでした。

本書で使用する通信モジュールは、通信モジュール単体で認証を得ているので、技適の手間や費用をかけずにすぐの実験を行うことができます(表4-1)。

<sup>19</sup> プロトコル・スタック：複数の階層に分かれた通信プロトコルを実装したソフトウェア。

<sup>20</sup> 技適：技術基準適合証明。

見本

## 第1節 ZigBeeの歴史とその特長を知っておこう

ZigBeeは、ZigBeeアライアンスが定めた無線通信方式の規格です。1998年にIntel社やIBM社などが設立したHomeRFワーキング・グループによって策定されたHomeRF Lite規格が基となっています。当時、Ericsson社もBluetooth SIGを設立してBluetoothの規格化を進めており、さらに無線LANについてもCCK方式と呼ばれる22Mbpsの規格化が進められていた時代です。規格化競争の中、HomeRFとBluetooth、無線LANの3方式の比較が盛んでした。

このような規格のデファクト化の競争の中で、HomeRFから派生し、ZigBeeの原型であるHomeRF Liteの検討が始まりました。ZigBee方式では、パソコンをターゲットとした無線LANや、携帯電話をターゲットとしたBluetoothとの競争を避け、通信速度を控える一方、表5-1に示すようなIoT<sup>21</sup>デバイスとしての特長を目標とすることになりました。

一戸建ての住宅内くらいをカバーする通信可能距離や、65535台もの最大接続可能端末数、乾電池で2年間は動作する超省電力、さらにチップ単価\$2といった特長から、家庭内のすべてのモノをワイヤレス接続するようなアプリケーションを想定しました。家電、充電電池や乾電池で動作する小型機器だけでなく、電気を使わなかったような物品など幅広い機器によるネットワーク構築が考えられ、またZigBee用のインターネット・ゲートウェイを経由し、クラウド上で機器の情報を共有したり、機器を制御したりすることが可能になりました。

実用的な応用例としては、家電機器に組み込んで

ホーム・オートメーション・システムを実現したり、センサや警報器などによるホーム・セキュリティ・システムを構築したりすることが得意な規格です。

ZigBee規格は、2004年12月にZigBee 2004(Ver. 1.0)仕様が策定され、2006年12月にZigBee 2006、そして2007年10月に通信の信頼性や効率性、秘匿性などを向上させたZigBee PRO 2007が策定されました。さらにエネルギー・ハーベスト(環境発電)とよばれる自然エネルギーや、ドアノブの回転による発電などの極めて小さな消費電力で動作するGreen Power対応のZigBee PRO 2012や、インターネット標準の規格化団体IETFが策定した6LoWPANを用いたZigBee IPも規格化されています。

ZigBeeが注目され始めたのは、ホーム・オートメーション機能を応用した家庭内の機器のネットワーク技術の実用化が始まったからです。また、2008年6月には、ZigBee PRO 2007上に、ZigBee Smart Energyプロファイルと呼ばれるアプリケーションが定義され、電力メータを中心にしたZigBeeネットワークも展開されました。

一方、前章に記載したとおり、Bluetooth SIGが規格化したBLEが、ZigBeeと似たような用途を展開しようとしています。従来のBluetoothとの互換性はなく、現実としてはあまり使われていないものの、デュアル・モードとして通常のBluetooth用ICに搭載され、搭載品の普及が進みつつあります。かつてのHomeRFの時代から、再びZigBeeの脅威となりました。

表5-1 ZigBee方式の概要

仕様	特長
通信可能距離	約40m(Digi International社 XBee ZBの屋内通信距離)
最大端末数	65535台
低消費電力	乾電池2本で最大2年間の動作(ZigBee SIG目標値)
低コスト	LSI単価で\$2(ZigBee SIG目標値)

<sup>21</sup> IoT(Internet of Things)：インターネットで情報共有したり制御することが可能な機器。

## 第1節 市販のXBee USB エクスプローラの機能比較

XBee ZB モジュールをパソコンや Raspberry Pi で使用するには、図 6-1 のように XBee USB エクスプローラ(写真 6-1)、または Digi International 社の XBIB-U-DEV ボード(写真 6-2)を使用して、パソコンの USB 端子に接続します。ここでは、市販の XBee USB エクスプローラの機能について説明します。なお、XBee ZB モジュールのファームウェアを書き換えるために、XBee ZB モジュールをパソコンに接続する必要があります。したがって、かならず1台以上の XBee USB エクスプローラが必要です。

市販の XBee USB エクスプローラの比較表を、表 6-1 に示します。それぞれ機能に違いがあるので、欲しい機能に合わせて購入すると良いでしょう。親機に使用するのであれば LED や SW は不要のように思われるかもしれませんが、開発時のデバッグ用と考えるとなるべく多くの機能がついているほうが良いでしょう。

以下に、表 6-1 中の各機能について説明します。

表中の項目「RSSI LED」は、他の XBee ZB からのパケットを受信したことを示す LED です。受信強度が高いほど明るく点灯します。デバッグ時の動作確認などに役立ちます。

項目「アソシエート LED」の LED は、XBee ZB のネットワークのジョイン(参加)状態や動作状態を示します。また、同じ ZigBee ネットワーク内にある XBee ZB 機器のコミッションング・ボタンが押されたときに高速に点滅します。うまく動かないときやデバッグ時、運用時に便利なので、アソシエート LED のついているものをお奨めします。

「リセット SW(ボタン)」は、XBee ZB モジュールの異常時などの復旧に使用するボタンです。電源 OFF と電源 ON による再起動でも対応できます。しかし、後述の XCTU という設定ツールから End Device の設定を変更するときなどに頻繁に使用します。End Device を使った実験には、必須の機能と考えても良いでしょう。

「コミッションング SW(ボタン)」は、XBee ZB ネットワークへの接続開始時にスリープ中の XBee ZB モジュールを起動したり、新しい XBee ZB 子機の ZigBee ネットワークのジョイン許可やネットワーク情報の初期化を行ったりする際に使用します。ZigBee ネットワークそのものの実験や、複数の ZigBee ネットワークを切り換えて使用する場合などに必要な機能です。

「FTDI Chip」は、USB シリアル変換用 IC が推奨 IC であるかどうかを示しています。XBee USB エクスプローラのメイン機能は、XBee ZB モジュールのシリアル信号をパソコン用の USB 信号に変換する機能です。このシリアル USB 変換用の IC には、FTDI 製の IC を推奨します。他社の IC の中には XBee ZB モジュールとの相性の良くないものもあり、機能に制約が生じたり XCTU からの設定にエラーが発生しやすくなったりする場合がありますからです。とはいえ、FTDI 製の IC を使えば、かならず安定して動作するとまでは言えません。他の IC に比べて安心できるといった程度です。

「ブート・ロード書き換え」は、XBee のファームウェアの書き換えを失敗したときなどに XBee ZB モジュールのソフトウェア修復に必要な機能です。XBee ZB モジュールが全く起動できなくなり、ファーム

ウェアの書き換えを失敗したときなどに XBee ZB モジュールのソフトウェア修復に必要な機能です。XBee ZB モジュールが全く起動できなくなり、ファーム

表 6-1 市販 XBee USB エクスプローラの比較表

XBee USB エクスプローラ	参考価格	RSSI LED	アソシ エート LED
秋月電子通商 AE-XBEE-USB	1,280 円	○	○
SWITCH SCIENCE XBee USB アダプタ	1,890 円	○	×
Strawberry Linux XBee エクスプローラ USB	2,205 円	○	×
SparkFun WRL-08687 XBee Explorer USB	2,500 円	○	×
Seeed Studio UartSBee V4	\$19.50	○	○
CQ 出版社 XBee 書込基板 XU-1	10,500 円	×	○

見本

## 第1節 サンプル1 XBeeのLEDを点滅させる

### SAMPLE 1

親機 XBee の RSSI 表示用 LED を点滅させる

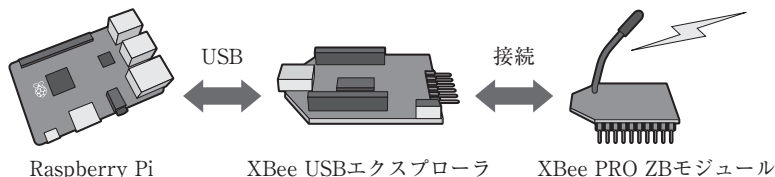
練習用サンプル

通信方式：XBee ZB

開発環境：Raspberry Pi

Raspberry Pi に接続した親機 XBee の LED を点滅させるサンプルです。市販されている多くの XBee 用 USB エクスプローラに搭載されている RSSI 表示用 LED を点滅させてみます(ワイヤレス通信は行わない)。

親機



ファームウェア：ZIGBEE COORDINATOR API		Coordinator	API モード
電源：USB 5V → 3.3V	シリアル：Raspberry Pi	スリープ(9)：-	RSSI(6)：LED
DIO1(19)：-	DIO2(18)：-	DIO3(17)：-	Commissioning(20)：(SW)
DIO4(11)：-	DIO11(7)：-	DIO12(4)：-	Associate(15)：(LED)
その他：Raspberry Pi に接続した親機 XBee ZB のみ(子機 XBee なし)の構成です。			

必要なハードウェア

- Raspberry Pi 2 Model B(本体, AC アダプタ, 周辺機器など) 1 式
- 各社 XBee USB エクスプローラ 1 個
- Digi International 社 XBee (PRO) ZB モジュール 1 個
- USB ケーブルなど

このサンプル1は、多くの市販 USB エクスプローラに実装されている RSSI 表示用の LED を点滅させるプログラムです。

RSSI は電波の受信強度を示す指標です。この LED の明るさで、およその電波の強度を示す目的で実装されています。このため、通常は制御する対象の LED ではありません。

ここでは、Raspberry Pi と XBee ZB モジュールとの間の UART シリアル通信が適切に動作しているかどうかを確認するために、RSSI 用の LED を使用します。Raspberry Pi のハードウェアや OS、XBee USB エクスプローラ、XBee ZB モジュールのそれぞれに問題がなければ、RSSI 表示の LED が正しく点滅します。

まず、親機用の XBee PRO ZB モジュールのファームウェアを「ZIGBEE COORDINATOR API」に変更します。PRO でない XBee ZB モジュールでもかまいません。すでに、第6章第5節で行っていた場合は、そのまま使います。また、XBee 管理用ライブラリのダ

ウンロードも必要です。こちらも、第6章第7節でダウンロードしていれば、再実行する必要はありません。

Raspberry Pi 用のサンプル・プログラムはダウンロードした「xbeeCoord」フォルダ内の「cqpub\_pi」フォルダに入っています。Leaf Pad でサンプル・プログラム1「example01\_rssi.c」を開いて確認してみましょう。

ここからは、コンパイルと実行方法について説明します。

Raspberry Pi の Raspbian で LXTerminal を開き、以下のコマンドを入力してフォルダ(ディレクトリ)を移動します。

```
$ cd ~/xbeeCoord/cqpub_pi
```

「example01\_rssi.c」をコンパイルするには以下のように入力します。

```
$ gcc example01_rssi.c
```

エラーが表示されなければコンパイルが成功し、実行ファイル「a.out」が作成されます。実行するには下記を入力します(「./」は「このフォルダ」ある)を意味

見本



## 第1節 サンプル 21 XBee Wall Router で照度を測定する

### SAMPLE 21

Digi International 製 XBee Wall Router で照度と温度を測定する

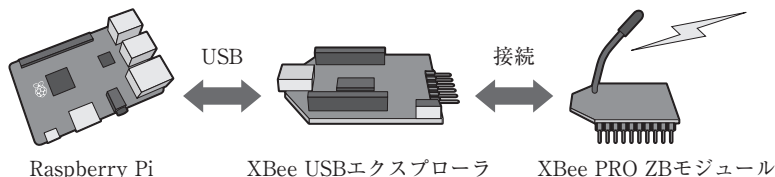
実験用サンプル

通信方式：XBee ZB

開発環境：Raspberry Pi

Digi International 社製の XBee Wall Router もしくは XBee Sensor の照度と温度を Raspberry Pi に接続した親機 XBee ZB モジュールから読み取る実験用サンプルです。

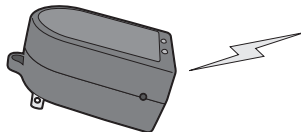
親機



ファームウェア：ZIGBEE COORDINATOR API	Coordinator	API モード
電源：USB 5V → 3.3V	シリアル：Raspberry Pi	スリープ(9)： -
DIO1(19)： -	DIO2(18)： -	DIO3(17)： -
DIO4(11)： -	DIO11(7)： -	DIO12(4)： -
		API モード
		RSSI(6)：(LED)
		Commissioning(20)：(SW)
		Associate(15)：(LED)

その他：XBee ZB モジュールでも動作します(ただし通信可能範囲は狭くなる)。

子機



ファームウェア：ZIGBEE ROUTER AT	Router	Transparent モード
電源：AC コンセント	シリアル： -	スリープ(9)： -
AD1(19)：照度センサ	AD2(18)：温度センサ	DIO3(17)： -
DIO4(11)： -	DIO11(7)： -	DIO12(4)： -
		API モード
		RSSI(6)： -
		Commissioning(20)：SW
		Associate(15)：LED

その他：照度センサ、温度センサの値は目安です。大きな誤差が生じます。

必要なハードウェア

- Raspberry Pi 2 Model B(本体、ACアダプタ、周辺機器など) 1式
- 各社 XBee USB エクスプローラ 1個
- Digi International 社 XBee PRO ZB モジュール 1個
- Digi International 社 XBee Wall Router または XBee Sensor 1台
- USB ケーブルなど

サンプル・プログラム 21 は、Digi International 社製の XBee Wall Router(ウォールルータ)もしくは、XBee Sensor の照度センサと温度センサの測定値を、Raspberry Pi 側の親機から取得する実験用サンプル・プログラムです。

XBee Wall Router は、コンセントに差し込んで XBee ZB の中継を行うための製品です。国内ではストロベリーリナックス社がレンジ・エクステンダーという名称で販売しています。

XBee Wall Router には、XBee PRO ZB モジュー

ルと、照度センサ、温度センサ、そして電源回路が内蔵されています。ファームウェアは、ZIGBEE ROUTER AT、デバイス識別子 ATDD の応答値は 00 03 00 08 です(後述)。また、内部ではアナログ入力 AD1 ポートに照度センサ、AD2 ポートに温度センサが接続されています。

まず、XBee Wall Router の動作確認をしてみましょう。これまでどおり、親機 Raspberry Pi には、ファームウェア ZIGBEE COORDINATOR API が書かれた XBee PRO ZB モジュールを接続します。そして、

見本

## 第1節 XBee Wi-Fi モジュールの特長

XBee Wi-Fi モジュールは、XBee PRO ZB モジュールとほぼ同じ大きさ、類似したインターフェースと使い勝手のまま、無線 LAN 規格である IEEE 802.11 に対応したワイヤレス通信モジュールです。

XBee ZB の場合は、親機となる XBee PRO ZB モジュールが必要であったため、最低2個の XBee ZB モジュールが必要でした。一方、ここで説明する XBee Wi-Fi モジュールは、1個から始めることができます。XBee Wi-Fi 単体でインターネットの標準プロトコルを搭載しているからです(図 9-1)。ただし、少なくとも Raspberry Pi と同じネットワークに接続された無線 LAN アクセス・ポイントが必要です。

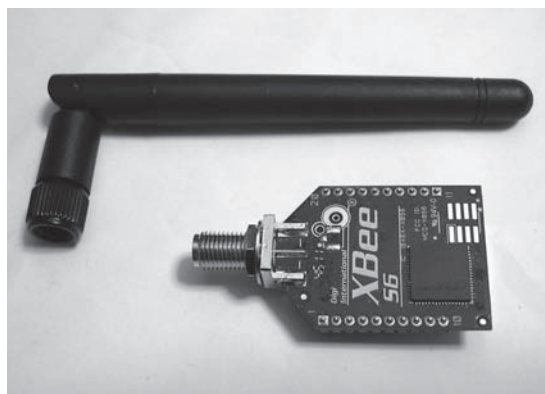


写真 9-1 XBee Wi-Fi モジュール (RPSMA アンテナタイプ) の一例

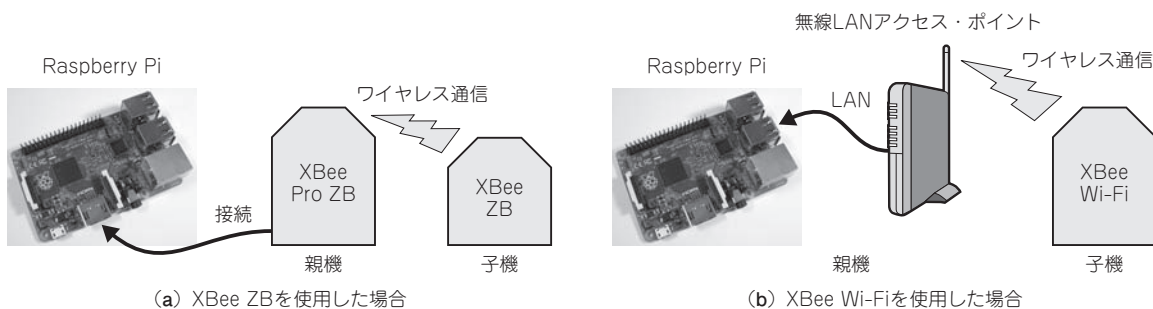


図 9-1 XBee ZB と XBee Wi-Fi との子機へのワイヤレス通信方法の違い

## 第2節 XBee Wi-Fi モジュールの無線 LAN 設定方法

XBee Wi-Fi の無線 LAN 設定を行うには、パソコンと XCTU が必要です。パソコンと XBee Wi-Fi との接続には、市販の XBee USB エクスプローラを使用します。ただし、XBee Wi-Fi の消費電力は XBee ZB に比べて大きいので、XBee USB エクスプローラの中には XBee Wi-Fi に対応していないものがあります。数回に一度、電源が入らないなど不安定な動作に陥るような場合は、電源供給が不足している可能性が高いです。不安定な場合は、XBee USB エクスプローラを XBee Wi-Fi 対応のものに変更してください。AC アダプタが使用可能な XBee USB エクスプローラ

の場合は、AC アダプタを接続してください。なお、不安定な状態で XBee Wi-Fi のファームウェアの書き換えると、故障の原因となります。

XCTU を起動し、XBee Wi-Fi が接続されたシリアル COM ポートを選択してから「Modem Configuration」タブに移り、「Read」ボタンを押すと、図 9-2 のような画面が表示されます。この画面で「Active Scan」を選択し、「Scan」をクリックすると無線 LAN アクセス・ポイントのスキャンが実行され、図 9-3 のようなスキャン結果が表示されます。

この中からアクセスした無線 LAN アクセス・ポ

見本

# 第1節 サンプル31 XBee Wi-FiのLEDを制御する①リモートATコマンド

## SAMPLE 31

XBee Wi-FiのLEDを制御する①リモートATコマンド

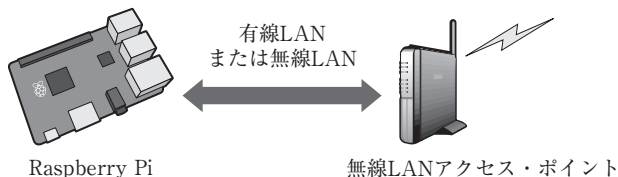
実験用サンプル

通信方式：XBee Wi-Fi

開発環境：Raspberry Pi

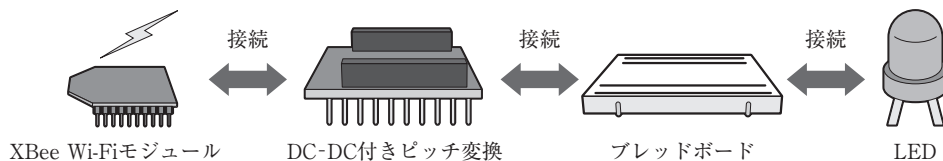
Raspberry Pi から子機 XBee Wi-Fi モジュールの GPIO(DIO) ポートをリモート制御するサンプルです。

親機



その他：Raspberry Pi と無線 LAN アクセス・ポイントを有線または無線で接続。

子機



ファームウェア：XBEE WI-FI

MA=1(Static IP)

電源：乾電池 2本 3V

シリアル：

-

スリープ(9)：

-

DIO10(6)：LED(RSSI)

DIO1(19)：(タクト・スイッチ)

DIO2(18)：

-

DIO3(17)：

-

DIO0(20)：

-

DIO4(11)：LED

DIO11(7)：

-

DIO12(4)：

-

Associate(15)：

-

その他：最新の Digi International 社の開発ボード XBIB-U-DEV (TH/SMT Hybrid)でも動作します。

必要なハードウェア

- Raspberry Pi 2 Model B(本体, ACアダプタ, 周辺機器など) 1式
  - 無線LANアクセス・ポイント 1台
  - Digi International社 XBee Wi-Fiモジュール 1個
  - DC-DC電源回路付きXBeeピッチ変換基板(MB-X) 1式
  - ブレッドボード 1個
  - 高輝度LED 1個, 抵抗 1kΩ 1個, セラミック・コンデンサ 0.1μF 1個, (タクト・スイッチ 1個 ※)
  - 2.54mmピッチピン・ヘッダ(11ピン) 2個
  - 単3×2直列電池ボックス 1個, 単3電池 2個, LANケーブル 1本, ブレッドボード・ワイヤ適量, USBケーブルなど
- ※ 本サンプルではタクト・スイッチをしません。

ここでは、子機 XBeeWi-Fi モジュールの GPIO (DIO) ポートに接続された LED を制御するサンプルを紹介します。ハードウェアは、ブレッドボード上に LED とタクト・スイッチ、ストロベリーリナックス製の DC-DC 電源回路付き XBee ピッチ変換基板「モバイルパワー XBee 変換モジュール MB-X」を実装して製作します(写真 10-1)。

XBee Wi-Fi にはコミショニング・ボタンがありません。あらかじめ第9章第2節にしたがって無線 LAN の設定を行い、XBee Wi-Fi に設定された IP ア

ドレスをプログラムに記述することで機器を特定します。したがって、使用する XBee Wi-Fi モジュールの IP アドレスを控えておく必要があります。

子機は、ブレッドボード上に DC-DC 電源回路付き XBee ピッチ変換基板を実装し、その上に XBee Wi-Fi モジュールを実装して製作します。しかし、先にモジュールを実装してしまうと他の部品を実装しにくいので、写真 10-2 のように先に周辺回路からブレッドボード上に実装します。

XBee ピッチ変換基板のピン(ピッチ変換基板

見本

## 第1節 入手しやすいBluetooth モジュール RN-42XVP

ここで使用する Bluetooth モジュール RN-42XVP は、入手しやすく、また国内の電波法に関する認証を取得済みなので、よく使われる定番品です。

もともと、通信方式としての Bluetooth はスマートフォンなどに標準搭載されており、価格や生産台数に強みがあり、アプリケーションも充実しています。しかし、その一方で、モジュール製品の供給先が大手企業に集中してしまい、少量での入手性が難しいという課題がありました。

筆者は2013年5月に Bluetooth モジュールを選定するための調査を行いました。機能の比較においては、対応プロファイル数の少ない RN-42XVP の優位性は低かったものの、国内の技適や認証の取得済みの Bluetooth モジュールの中ではもっとも安価だったことや、少量でも入手が可能であったこと、XBee と互換性のあるコネクタを装備していることなどから、本 Bluetooth モジュールを選定しました。しかし、当時の入手手段としては、米 Microchip Technology 社から直接購入するくらいしかなく、送料や輸送期間などを考えると手軽とは言えませんでした。

その後、秋月電子通商が RN-42XVP の取り扱いを

開始し、また同社から互換品の AE-RN-42-XB(参考価格 2,000 円)が発売されるなど、今では Bluetooth モジュールの定番品と呼ばれるくらいになりました。

対応プロファイル数が少ないとはいえ、キーボードやマウス、ゲーム用コントローラなどで用いられている HID(ヒューマン・インターフェース・デバイス)プロファイルを搭載しているのも特徴の一つです。本書でも HID プロファイルを利用し、Arduino から Raspberry Pi のキーボード操作(カーソル操作)を行うサンプルを紹介します。



写真 11-1 Bluetooth モジュール RN-42XVP

## 第2節 Raspberry Pi に接続する Bluetooth USB アダプタ

ここでは Raspberry Pi に Bluetooth USB アダプタを接続し、RN-42XVP との間で Bluetooth 通信を行います。最新の Raspberry Pi 3 Model B を使用することも可能ですが、執筆時点では動作確認ができていません。動作が不安定な場合は、Raspberry Pi 3 の Wi-Fi を切って、有線 LAN で使用してください。Bluetooth と Wi-Fi は同じ周波数を使うので、電波干渉が発生し、通信が安定しなくなる場合があります。ここでは PCI (プラネックスコミュニケーションズ) 製 BT-Micro4 (写真 11-2) を使用します。必要な機材を表 11-1 に示します。

なお、この製品に関わらず CSR(Cambridge Silicon

Radio)社のチップが使われているものであれば、動作すると思います。ただし、Bluetooth 2.0 以前の古い



写真 11-2 PCI 製 Bluetooth USB アダプタ BT-Micro4

見本

## 第1節 サンプル36 IchigoJamをRaspberry Piから制御する

### SAMPLE 36

IchigoJamをRaspberry Piから制御する

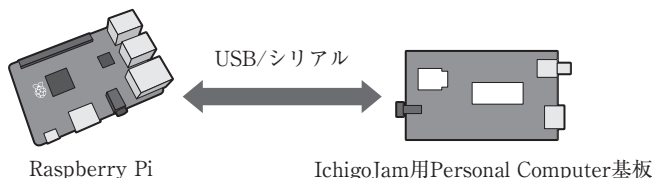
実験用サンプル

通信方式：UART シリアル

開発環境：Raspberry Pi

Raspberry PiからIchigoJamをリモート制御するサンプルです。

親機



その他：参考文献(6)用の別売パーツ・セットに含まれるパーツとプリント基板を組み立てて使用。

必要なハードウェア

- Raspberry Pi 2 Model B(本体, ACアダプタ, 周辺機器など) 1式
- IchigoJam用Personal Computer基板[参考文献(6)] 1式
- ブレッドボードジャンパ線(メス⇄オス)3本, またはUSBケーブル(USBシリアル変換IC搭載時)

ここではRaspberry Piに, jig.jp社のIchigoJamもしくは, 参考文献(6)用のIchigoJam用Personal Computer基板(CQ出版社)を接続し, Raspberry PiからIchigoJam BASICを利用する方法について説明します。

IchigoJamは株式会社jig.jpによって開発されたブ

ログラミングの学習教材用マイコン・ボードです。テレビとキーボードを接続することで, BASICによるプログラミングが可能な学習用パソコンとして使用することができます。パソコンと同等機能を備えているRaspberry Piに対し, IchigoJamはBASICのプログラミングに限定して設計されました。

表 12-1 Raspberry Pi と IchigoJam の比較

項目	Raspberry Pi 2 Model B	IchigoJam U
CPU	ARM Cortex-A7 Quad Core 900MHz	ARM Cortex-M0 48 MHz
Flash	micro SD	32KB
RAM	1GB	4KB
OS	Raspbian(Linuxベース)等	IchigoJam BASIC
表示	フルHD出力	テキスト文字
ストレージ	micro SD(別売)/USBメモリ(別売)	内蔵4KB/外付EEPROM(別売)
カメラ	専用品(別売)	-
HDMI端子	○	-
LAN端子	○	-
USB端子	○	-
GPIO端子	○	○
アナログ入力	-	○
キーボード	汎用USBキーボード対応	PS/2仕様キーボードのみ対応
マウス	汎用USBマウス対応	-
ACアダプタ	汎用Micro USB ※1A以上が必要	汎用Micro USB電源が使用可能
市販マイコン	-	市販マイコンにファームの書き込み可能
消費電力(実力)	約500mA	約20mA 乾電池で駆動可能

見本

## 第1節 BLE(Bluetooth Low Energy)について

ここでは各社から販売されている BLE タグ(写真 13-1)を用いて BLE タグのビーコンを受信する実験を行います。BLE は Bluetooth 4.0 でサポートされた Bluetooth Low Energy と呼ばれるプロトコルです。Bluetooth Smart の名称で普及を進める動きが高まりつつあります。

執筆時点では、Bluetooth と言えば従来のクラシック Bluetooth を示します。Bluetooth Low Energy や Bluetooth Smart にも Bluetooth の名称が付きますが、クラシック Bluetooth との通信を行うことはできません。また、今のところ BLE を使ったアプリケーションも普及しているとは言い難い段階です。

しかし、すでに多くのスマートフォンや Bluetooth USB モジュールに Bluetooth 4.0 対応の IC が実装されています。これらの IC には、従来のクラシック Bluetooth と BLE の両方の通信機能がデュアル・モードで搭載されています。もちろん、本書で使用するブ

ラネックスコミュニケーションズ製の Bluetooth USB アダプタ BT-Micro 4 にも BLE 機能が搭載されています。BLE そのものの普及はこれからですが、すでに BLE レディ機器が普及している点に強みがあり、今後、急速に普及が進む可能性があります。

なお、Bluetooth Smart 機器は、BLE プロトコルのみをサポートしたシングル・モード品です。デュアル・モードではないため、従来のクラシック Bluetooth との通信は行えません。

それでは、BLE タグの動作確認を行ってまいしょう。Raspberry Pi 側の構成は、これまでのクラシック Bluetooth の実験のときと同じです。Bluetooth USB アダプタ BT-Micro 4 を USB ポートに接続します。また、第 11 章第 3 節でインストールした Bluetooth 用プロトコル・スタック BlueZ のツール hcitool を使用します。

接続先の機器は BLE タグです。この間のワイヤレス通信の方式が、従来のクラシック Bluetooth 方式とは異なる BLE 方式になります(図 13-1)。

機器の探索方法は、従来のクラシック Bluetooth とは異なります。BLE 機器を探索する場合は、hcitool の `lescan`(LE スキャン)命令を使用します。LXTerminal から以下のコマンドを入力してください。

```
$ sudo hcitool lescan --pa --du
```

電源の入った BLE タグの MAC アドレスが図 13-2 のように表示されれば、正しく動作していることがわかります(3 個の BLE タグを使用した例)。ただし、電源を入れてから時間が経つと送信頻度が低下します。また、LBT-VRU01 の場合は約 5 分後に自動的に電源が切れて、ビーコンを送信しなくなります。



写真 13-1 各社から販売されている BLE タグ

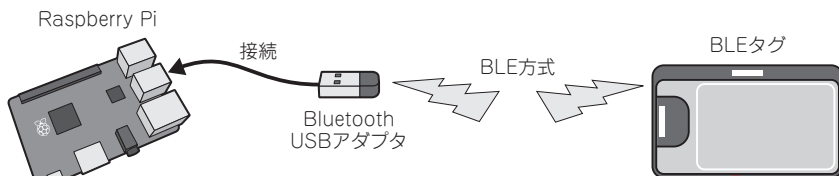


図 13-1 BLE タグを使った実験

見本

## 第1節 Bluetooth モジュール RN-42XVP のローカル接続コマンド・モード

XBee ZB モジュールでは、AT コマンドを使ってモジュールを制御することができました。Bluetooth モジュール RN-42XVP にも、似たようなコマンド・モードがあります。本章では、このコマンド・モードについて解説します。

はじめに本節では、Raspberry Pi に RN-42XVP をローカル接続した場合のコマンド実行方法について説明します(非ワイヤレス通信)。

RN-42XVP を XBee USB エクスプローラに接続し、その XBee USB エクスプローラを Raspberry Pi に接続します。そして、シリアル端末 cu を使って、RN-42XVP を接続したシリアル・ポートを開きます。USB0 を開くには以下のように入力します。

```
$ cu-h-s-115200-l/dev/ttyUSB0 ↵
```

「Connect」が表示されたら、ローカル接続の完了です。次に、コマンド・モードへ入る手続きを行います。「\$ (ドルマーク)」を3回、「\$\$\$」と続けて入力してください。このときに「Enter」キーを押さないように注意します。Bluetooth モジュール RN-42XVP がコマンド・モードに入ると「CMD」の応答が得られ、RN-42XVP 上の LED が高速に点滅します。

```
(入力) 「$」「$」「$」 (Enter 不要)
```

…コマンド・モードへ

```
(応答) CMD
```

次節では、リモート接続した状態でコマンド・モードに入りますが、Bluetooth モジュール RN-42XVP が起動してから 60 秒が経過すると、リモート接続によるコマンド・モードへの移行ができなくなる場合があります。

RN-42 のバージョンが、Ver 6.15 04/26/2013 よりも古い場合は、この60秒の制限を解除しておきましょう。コマンド・モードに移行した状態で以下の、ST, 255 を実行します。

```
(入力) 「s」「T」「,」「2」「5」「5」 ↵
```

…モード移行制限の解除

```
(応答) AOK
```

コマンド・モードを抜けるには、「-(マイナス)」を3回「---」と入力して「Enter」を押します。「END」の応答が得られ、LED の点滅速度も元の速さに戻ります。また、「~」(チルダ)と「.」(ピリオド)を入力すると、cu 端末が終了します。

```
(入力) 「-」「-」「-」 ↵
```

```
(応答) END
```

```
(入力) 「~」「.」
```

ワイヤレスを経由した実験ではトラブルも多く、とくに RN-42XVP が正常に動作することを確認する際や、コマンドのテストを行うときなどにローカル・コマンドを使用することがあります。

例えば、Bluetooth 接続ができなくなってしまう場合や、ヒューマン・インターフェース・デバイス (HID) プロファイルに変わってしまうこともあります。そのような場合は、ローカル接続した状態で、以下の手順で設定を工場出荷状態に初期化します。

```
$ cu-h-s-115200-l/dev/ttyUSB0 ↵
```

```
(入力) 「$」「$」「$」 (Enter 不要)
```

…コマンド・モードへ

```
(応答) CMD
```

```
(入力) 「s」「F」「,」「1」 ↵
```

…工場出荷状態の設定に戻す

```
(応答) AOK
```

```
(入力) 「R」「,」「1」 ↵
```

…再起動(再起動後に設定完了)

```
(応答) Reboot!
```

```
(入力) 「~」「.」 …シリアル端末 cu の終了
```

なお、ローカル接続コマンド・モードとリモート接続コマンド・モードを同時に実行することはできません。LED の点滅速度が低速に戻っていることを、確認してから、次節に進んでください。高速に点滅したままの場合は、シリアル端末 cu から「-」「-」「-」↵を入力して、コマンド・モードを解除してください。

# 見本

# 第1節 サンプル 46 Bluetooth モジュール RN-42XVP の LED 制御

## SAMPLE 46

Bluetooth モジュールのペアリングと LED の制御

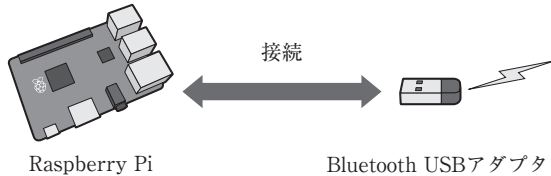
実験用サンプル

通信方式：Bluetooth

開発環境：Raspberry Pi

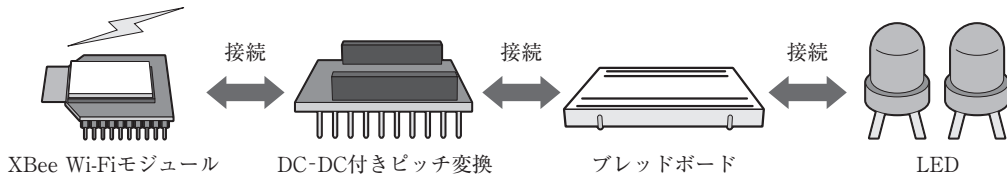
親機となる Raspberry Pi から子機 Bluetooth モジュールの GPIO を制御して GPIO ポート 9 と 10 に接続した LED を点滅します。リモートで使用するための各種設定とペアリングも行います。

親機



その他：Raspberry Pi に Bluetooth USB アダプタ (Bluetooth 4.0 対応) を接続。

子機



ファームウェア：RN-42 Firmware 6.15 以降	Slave	SPP プロファイル
電源：乾電池 2 本 3V	シリアル： -	GPIO 11(9)： -
ADC 1(19)： -	GPIO 7(18)：(タクト・スイッチ)	GPIO 3(17)：(タクト・スイッチ)
-	GPIO 9(7)：LED	GPIO 10(4)：LED
その他：XBee ZB とはピンの役割などに相違があります。		Associate (15)：LED (RN42)

### 必要なハードウェア

- Raspberry Pi 2 Model B (本体, AC アダプタ, 周辺機器など) 1 式
- Bluetooth USB アダプタ Planex BT-Micro4 1 個
- Microchip 社 Bluetooth モジュール RN-42XVP 1 個
- DC-DC 電源回路付き XBee ピッチ変換基板 (MB-X) 1 式
- ブレッドボード 1 個
- 高輝度 LED 2 個, 抵抗 1kΩ 4 個, セラミック・コンデンサ 0.1μF 1 個, (タクト・スイッチ 2 個), 2.54mm ピッチピン・ヘッダ (11 ピン) 2 個
- 単 3×2 直列電池ボックス 1 個, 単 3 電池 2 個, ブレッドボード・ワイヤ適量, USB ケーブルなど

Bluetooth によるワイヤレス通信を使用して、親機から子機に GPIO 制御信号を送信し、LED の点灯・消灯をリモート制御するサンプルについて説明します。

親機の Raspberry Pi 側の構成は、これまでのクラシック Bluetooth や BLE と同様です。本章のサンプル・プログラム 46 とサンプル・プログラム 47 で使用する子機は、ブレッドボードを使用して製作します。表 15-1 に製作する子機に搭載する Bluetooth モジュール RN-42XVP の GPIO ポートの使用方法を示します。表中の「XBee ピン」の番号は、Bluetooth モ

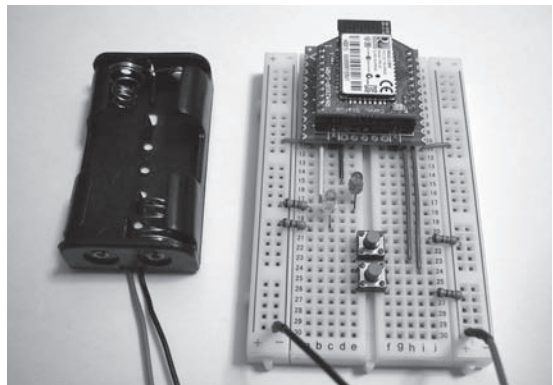


写真 15-1 Bluetooth 搭載スイッチ & LED 子機の製作例

見本



## 第1節 インターネット上のデータを curl や wget で取得する

curl は、インターネット上のデータを取得するためのコマンドです。ここではインターネットからファイルを取得する方法について説明します。まずは、Raspberry Pi の Raspbian を使用し、LXTerminal から以下のように curl コマンドを実行してみてください。

```
$ curl -s www.geocities.jp/  
bokunimowakaru/cq/pi.txt ↵
```

図 16-1 のように、指定した URL のテキスト・ファイルをインターネットから取得し、情報が表示されます。この情報は本書のサポート・ページの更新情報です。更新頻度は、数カ月に 1 回程度か、場合によってはもっと長期間になるかもしれません。

オプションの「-s」は、取得時の進捗状況やエラーなどを表示させないためのオプションです。その次の URL は取得するファイルのインターネット・アドレスです。ここでは「http://」を省略しましたが、明示することでプロトコルを指定することも可能です。もちろん、ウェブ・ページなどの HTML ファイルを取得することも可能です。

今度は、時刻を取得してみましょう。下記を実行すると、(独立行政法人)情報通信研究機構が配信する時刻情報を取得することができます。

```
$ curl -s ntp-a1.nict.go.jp/  
cgi-bin/time ↵
```

ただし、こういったサービスは、配信する団体等の都合で終了する可能性もあります。何のデータも得られなかった場合や、HTML で書かれたエラーが表示された場合は、ご容赦ください。またサービスによっては、販売や運用目的で利用する場合にサービス提供者との契約やサービス利用料の支払いが必要になります(以降で紹介するサービスも同様)。

次に、類似のコマンド wget を使ってみましょう。以下のように入力してみてください。

```
$ wget -q -O /dev/  
stdin www.geocities.jp/  
bokunimowakaru/cq/pi.txt ↵
```

この wget コマンドの第 1 引き数の「-q」は、curl コマンドの「-s」に相当し、取得時の状況表示やエラー表示を無効にします。第 2 引き数の「-O/dev/stdin」は、標準入力に取得内容を出力するためのオプションです。このオプションを省略すると、現在のフォルダにファイルとして保存されます。

このように、curl コマンドを使っても wget コマンドを使っても同じようなことを実現することができます。どちらかと言えば、curl コマンドはインターネット上のデータにアクセスする際に使われ、wget コマンドはインターネット上のコンテンツのダウンロードに使われることが多いです。本書では curl コマンドを用いることにします。

```
pi@raspberrypi ~ $ curl -s www.geocities.jp/bokunimowakaru/cq/pi.txt ↵  
<title> これはテキストによるメッセージ配信の実験です。</title>  
<descr> 以下、お知らせ、近況、関連 URL、更新日の情報です。</descr>  
<info> 今のところ[お知らせ]はございません。</info>  
<state> 現在のボクの[近況]は本書の執筆中です。</state>  
<url> http://www.geocities.jp/bokunimowakaru/cq/15/</url>  
<date> 2016/01/11</date>  
pi@raspberrypi ~ $ curl -s ntp-a1.nict.go.jp/cgi-bin/time ↵  
Mon Jan 11 19:46:00 2016 JST  
pi@raspberrypi ~ $
```

図 16-1 インターネットから情報を取得したときのようす

見本

ISBN978-4-7898-4223-5

C3055 ¥3200E

**CQ出版社**

定価：本体3,200円（税別）



9784789842235



1923055032002

Wi-Fi/Bluetooth/ZigBee無線用  
**Raspberry Pi**  
プログラム全集

Complete collection of Raspberry Pi software programs for Wi-Fi/Bluetooth/ZigBee

見本