



第3章

しちかから！何はともあれ
入出力機能GPIOを攻略しよう

レッスン3

ラズパイ自由自在！ 二つの飛び道具 PythonとWiringPi

砂川 寛行 Hiroyuki Sunagawa

ラズベリー・パイのプログラミングには、Python と言う言語がおすすめです。C言語より記述が短いのでプログラミングに不慣れでも使いこなせます。さらに、ハードウェア制御用ライブラリ(WiringPi)も使うとプログラムを作る手間はもっと減ります。

Python×WiringPiで 目の前がパッと開ける

① Python言語なら短く読みやすく書ける

Pythonプログラムでラズベリー・パイのGPIOを操作してみます。

Pythonはコード数が少なくて済み、可読性が高いスクリプト言語の一種です。ラズベリー・パイの当初の目的は教育用途であり、ラズベリー・パイのパイは、Pythonに由来しているそうです。Raspbianをインストールしたラズベリー・パイには、Pythonが標準でインストールされています。

Pythonはネットワーク関連の操作も容易です。メールもツイッター投稿もすぐできます。ただし、インデントの位置は重要で、そろえないとエラーになります。

② Python用ハードウェア制御ライブラリ WiringPi

GPIOを操作するためにライブラリ WiringPi を使用します。

LinuxでGPIOを制御するには、いろいろな手順が必要ですが、WiringPiはそれらを肩代わりしてくれます。お手軽マイコン・ボード Arduino を使うのと同じイメージで入出力、PWM、シリアル(UART, SPI, I²C)を扱えます。

WiringPiではGPIOの番号(呼び名)が変わります。コネクタの配置順に番号が付与されている従来の呼び名も設定可能ですが、機能に制約が発生することもあります。

WiringPiの動かし方

■ 準備

ライブラリ WiringPi を準備します。Python は Raspbian に標準搭載されているので準備は不要です。

手順1 ソフトウェア管理プログラム git のインストール

```
sudo apt-get install git-core
```

手順2 git で WiringPi を入手

git は、インターネットで公開されている各種プログラムやドキュメントなどを取得するためのツールです。git を使用して wiringPi のコードを入手します。

```
git clone git://git.drogon.net/wiringPi
```

手順3 WiringPi をコンパイル

入手した WiringPi をビルド(コンパイル)します。

BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
2	8	3.3v		1	2			5v		
3	9	SDA_1	ALTO	1	3	4		0v		
4	7	SCL_1	ALTO	1	5	6		0v		
17	0	GPIO_0	IN	1	11	12	0	IN	GPIO_1	1
27	2	GPIO_2	OUT	0	13	14			GPIO_5	5
22	3	GPIO_3	IN	0	15	16	0	IN	GPIO_4	4
10	12	MISO	ALTO	0	17	18	0	IN	GPIO_6	6
9	13	MOSI	ALTO	0	21	22	0	IN	GPIO_8	8
11	14	SCLK	ALTO	0	23	24	1	OUT	GPIO_9	9
0	30	SDA_0	IN	1	27	28	1	IN	GPIO_10	10
5	21	GPIO_21	IN	1	29	30			GPIO_11	11
6	22	GPIO_22	IN	1	31	32	0	IN	GPIO_12	12
13	23	GPIO_23	IN	0	35	36			GPIO_13	13
19	24	GPIO_24	IN	0	37	38	1	OUT	GPIO_14	14
26	25	GPIO_25	IN	0	37	38	0	IN	GPIO_15	15
		0v			39	40	0	IN	GPIO_16	16

図1 gpio readall を実行した結果。コネクタの全GPIOの状態を一度に確認できる