

ラズベリーパイでセンサーを扱おう

ラズベリーパイというデバイスを使ってLEDやセンサなど外部デバイスとの入出力を試みよう。ラズベリーパイでドリトルを使用するためには、ラズベリーパイと専用版ドリトルが必要である。

ラズベリーパイとは

ラズベリーパイは、イギリスのラズベリーパイ財団が開発したシングルボードコンピュータである。CPUはARMを搭載しておりOSはLinuxや専用版Windowsなどが動作する。そのため、キーボードやマウス、ディスプレイを接続することで、一般的なコンピュータと同じように使うことができる。またRaspberry 2 Model Bでは、40の入出力ピンを備えておりLEDの点灯や、スイッチの入力、センサーでの計測などを行うことができる。¹⁾



ドリトルのインストール

今回は、OSとしてRaspbianを導入したRaspberry 2 Model Bという製品での動作を確認している。また、以下の作業はOSのインストールが済んでいる状態で、すべてコンソール上で行う。²⁾

1. Raspberry piをアップデートする

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

2.以下のコマンドを実行するとドリトルがインストールされ、センサーを利用するためのI2Cの設定が行われる

```
$ wget http://dolittle.eplang.jp/data/dist/dolittle238rp.deb
$ sudo dpkg -i dolittle238rp.deb
```

3.I2Cの設定を有効化するためにラズベリーパイを再起動する

ドリトルを起動する

ドリトルをインストールすると画面左上のメニューに登録される。ドリトルは、Menu→プログラミング→Dolittleから起動することができる。

センサーなどのパーツを接続する

センサーやLEDなどのパーツは、すべてGPIOポートに接続する。³⁾ポート毎の機能は以下の表の通りである。パーツの接続にはジャンプワイヤーやブレッドボードを使うと便利である。⁴⁾

ポートの機能一覧

機能	番号	番号	機能
3.3V	1	2	5V
GPIO2(I2C□SDA)	3	4	5V
GPIO3(I2C□SCL)	5	6	GND
GPIO4(GPCLK0)	7	8	GPIO14(UART□TXD)
GND	9	10	GPIO15(UART□RXD)
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10(SPI□MOSI)	19	20	GND
GPIO9(SPI□MISO)	21	22	GPIO25
GPIO11(SPI□SCLK)	23	24	GPIO8(SPI□CE0)
GND	25	26	GPIO7(SPI□CE1)
ID□SD	27	28	ID□SC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

なお、使用できるセンサーは現在すべてI2Cで通信を行うのでGPIO2ピンとGPIO3ピンを占有する。この2つのピンはGPIOとしての機能も持っているためLEDやタクトスイッチなどでこれらのピンを使わないようにする必要がある。またI2Cのセンサーを複数同時に使用する場合はラズベリーパイのI2Cのピンを分岐させてそれぞれのセンサーへと繋がれば良い。

ラズベリーパイのプログラムの基本

ラズベリーパイを用いたドリトルのプログラムは以下の様な形になる。先頭の行では、ラズベリーパイを使うプログラムを作成することを示している。以後、ラズベリーパイと接続されたセンサー類を指す「ラズパイ」というオブジェクトが利用可能となる。

```
システム□"raspberrry" 使う。
ポート4 = ラズパイ！4 入力ポート。
ポート4！プルアップ。
...
```

LEDを使う

LEDはGPIOの機能を持ったポートに接続する。⁵⁾

「出力ポート」オブジェクトは、GPIOポート番号を引数に取って初期化を行い、以下の命令を持っている。

出力ポートオブジェクトの命令一覧

命令	機能
書く	1か0を引数に取り、オン・オフを制御する
読む	現在のGPIOポートの値を取得する
後始末	ポートを解放する

次の配線図とプログラムは、GPIO4ポートに接続したLEDを2秒毎に10回点滅させるものである。



```
システム[] "raspberrry"[]使う。
```

```
ポート4 = ラズパイ！ 4 出力ポート。
```

```
[]
```

```
    ポート4！1 書く。
    ラズパイ！2 待つ。
    ポート4！0 書く。
    ラズパイ！2 待つ。
```

```
」！ 10回 繰り返す。
ポート4！後始末。
```

スイッチを使う

GPIOポートはデジタル入力として用いることもできる。「入力ポート」オブジェクトはGPIOポート番号を引数に取って初期化を行い、以下の命令を持っている。

入力ポートオブジェクトの命令一覧

命令	機能
プルアップ	Raspberry pi内部のプルアップ抵抗を有効にする
プルダウン	Raspberry pi内部のプルアップ抵抗を無効にする
読む	現在のGPIOポートの値を取得する
後始末	ポートを解放する

次の配線図とプログラムは、GPIO4ポートに接続したタクトスイッチから値を取得し、画面上に表示させるものである。⁶⁾

```
システム[] "raspberrry"[]使う。
```

```
ポート4 = ラズパイ！ 4 入力ポート。
```

```
ポート4！プルアップ。
```

```
値 = フィールド！作る。
```

```
時計 = タイマー！作る 1秒 間隔 60秒 時間。
```

```
時計！「
```

```
    値！（ポート4！読む） 書く。
```

```
」実行。
```

```
ポート4！後始末。
```



温湿度センサーを使う

温度と湿度を計測できるセンサーとしてTexasInstruments社のHDC1000に対応している。このセンサーはI2Cで通信を行っている。

温湿度センサーを扱うためにはまず温湿度センサーオブジェクトを作る必要がある。そして、温湿度センサーオブジェクトには「温度？」命令と「湿度？」命令が存在し、それぞれ温度と湿度を読み出すことができる。

次のプログラムは、センサーから1秒毎に10回温度と湿度を読み取って画面上に表示するプログラムである。

```
システム"raspberrry"を使う。
温湿度センサー = ラズパイHDC1000
温度 = ラベル！作る。
湿度 = ラベル！作る 次の行。
□
  温度！（温湿度センサー！温度？） 書く。
  湿度！（温湿度センサー！湿度？） 書く。
  ラズパイ！1 待つ。
」！10回 繰り返す。
```

照度センサーを使う

照度を計測できるセンサーとしてTAOS社のTSL2561に対応している。このセンサーはI2Cで通信を行っている。

照度センサーを扱うためにはまず照度センサーオブジェクトを作る必要がある。照度センサーオブジェクトは「照度？」命令を持っており、照度を読み出すことができる。

次のプログラムは、センサーから1秒毎に10回照度を読み取って画面上に表示するプログラムである。⁷⁾

```
システム"raspberrry" 使う。
照度センサー = ラズパイTSL2561
照度 = ラベル！作る。
□
  照度！（照度センサー！照度？） 書く。
  ラズパイ！1 待つ。
」！10回 繰り返す。
```

気圧センサーを使う

気圧と気温を計測できるセンサーとしてSTMicroelectronics社のLPS25Hに対応している。次のプログラムはI2Cで接続された気圧センサーから値を読み取り、画面上に表示するプログラムである。⁸⁾

気圧センサーを扱うためにはまず気圧センサーオブジェクトを作る必要がある。気圧センサーオブジェクトは「気圧？」命令と「気温？」命令を持っており、それぞれ気圧と気温を読み出すことができる。

```
システム"raspberrry"を使う。
気圧センサー = ラズパイLPS25H
気圧 = ラベル！作る。
気温 = ラベル！作る 次の行。
```

```

□
  温度！（気圧センサー！気圧？） 書く。
  湿度！（気圧センサー！気温？） 書く。
  ラズパイ！1 待つ。
」！10回 繰り返す。

```

モータードライバを使う

モータードライバは、モーターを動作させるために用いるICである。このライブラリでは、東芝のTA7291に対応している。モーターを動作させるには、モータードライバとモーターに加え、電池ボックス等外部の電源が必要になる。

モータードライバを扱う際にはまず、モータードライバのオブジェクトを作る必要がある。そして、モータードライバの制御ピンに繋いだピン番号2つを引数にとって「作る」命令を実行する。「作る」命令を実行すると、モータードライバが初期化される。モータードライバオブジェクトは以下の命令を持っている。機能は以下の表のとおりである。

モータードライバオブジェクトの命令一覧

命令	機能
作る	モータードライバオブジェクトを初期化する
後始末	モータードライバオブジェクトが占有しているGPIOピンを開放する
正転	モーターを正転させる
逆転	モーターを逆転させる
止める	モーターを停止させる
ブレーキ	モーターにブレーキをかけて停止させる

次のプログラムはGPIO14□GPIO15ピンに接続したモータードライバを用いて、モーターを回転させるプログラムである。⁹⁾

```

システム□ "raspberrypi" □使う。
モータードライバ=ラズパイ□TA7291□
モータードライバ！14ポート 15ポート 作る。
モータードライバ！正転。
ラズパイ！3 待つ。
モータードライバ！止める。
ラズパイ！1 待つ。
モータードライバ！逆転。
ラズパイ！3 待つ。
モータードライバ！ブレーキ。
ラズパイ！1 待つ。
モータードライバ！後始末。

```

1)

ラズベリーパイの仕様については（<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>）のサイトに詳しく掲載されている。

2)

Raspbianのインストールには、（<http://breadboard.jp/noobs-20150603>）のサイトを参考にさせていただきたい。また、日本語入力には追加のソフトウェアのインストールが必要である。こちらについては（<http://ryus.co.jp/blog/raspberrypi2-4/>）のサイトを参考にさせていただきたい。

3)

GPIO:General Purpose Input/Output 汎用入出力

4)

ラズベリーパイのピンをブレッドボード上へ引き出す延長ケーブルのような製品も存在する。<https://www.switch-science.com/catalog/2074/>

5)

GPIOが出力できる電圧は3.3V、電流は8mAである。そのため、定格が上限を超えるLEDを直接点灯させることはできない。またLEDに合わせて適当な容量の抵抗を接続することが望ましい。

6)

入力は内部でプルアップしているため、ボタンが押されている時値は0になる。

7)

このセンサーは仕様により、急に明るさが変わると値が0になる

8)

このセンサーは4ピンをVCCに繋ぐかGNDに繋ぐかによってセンサーのアドレスが変わる。ここではGNDに繋いだ場合の0x5cを前提としている。

9)

モータードライバオブジェクトの「後始末」命令が実行されると、ポートが開放されるため、ピンの電圧は不定となる。そのため、止まっているモータが突然動き出す可能性がある。

From:

<https://dolittle.eplang.jp/> - プログラミング言語「ドリトル」

Permanent link:

https://dolittle.eplang.jp/ch_raspiLast update: **2018/02/09 11:07**