### Raspberry Pi 3 Model BによるIoTの実践(2019/09/18改)

松井博和, 松崎大起 第一回Raspberry Pi 3 Model Bの紹介と連載予定 (2017/01/10) 第二回LinuxOSのインストール(2017/03/16) 注(2019/09/18) 松井博和,島中美羽 第三回Raspbianの設定(前半) (2017/08/22) 松井博和 補回 Raspbianの設定(後半)(2017/09/18) 注(2017/11/24) 松井博和 第四回画像処理のためのOpenCVの導入 (2017/09/19)(2019/09/18改) 注(2017/11/24) 松井博和 第五回GPIOを用いるLED点滅 (2017/10/19)(2019/09/18改)(トイサーボ制御割愛) 注(2019/09/18)松井博和 第六回ネットワークの設定 (2017/11/20) 松井博和 第七回<u>リモートログイン(前回続編)(</u>2018/11/25) 松井博和 第八回アナログ入力の使用 (spiデバイスとの通信例) (2018/12/17) 松井博和 補回 RasPi3Bでの音声の入出力 (未定) 松井博和 来回 Mathmaticaの使用(未定) 松井博和 来回 Fpga拡張カード(未定) 松井博和 松井博和, 松崎大起 来回 音声認識(CMU Sphinx)(未定)

この連載の一部は、公益財団法人三重県産業支援センターが毎年3月にその実施を決定し10月から翌年2月まで無料で実施する 高度製造技術基礎講座の一部として演習付きで実施する.本レポートの内容の詳細は、その講座でも解説する.

> 三重大 機械工 メカトロニクス研 松井博和 mailto:hmatsui{at}robot.mach.mie-u.ac.jp Phone: (059)231-9802

# Raspberry Pi 3 Model BによるIoTの実践(第一回)

## 三重大学 機械工 松井博和, 松崎大起 http://www.robot.mach.mie-u.ac.jp/~hmatsui/RasPi3B/

Keywords: 超小型パソコン, IoT, 画像処理, ロボット制御

## 1. Raspberry Pi 3 Model Bの紹介

**Fig. 1** にロゴを示す Raspberry Pi は,英国のラズベリーパイ財団製の名刺大の超小型コンピュータのシリーズであり,発売約4年で累計1000万台売れている.その



Fig. 1 Raspberry Pi Logo

中でも Fig. 2 に示す最新の Raspberry Pi 3 Model B(以降 RasPi3B) は,日本で税込み 5000 円弱で購入でき,無線 LAN や bluetooth をもつ IoT(Internet of Things) の基として適するものである.RasPi3B は下記特徴をもつ.



Fig. 2 Raspberry Pi 3 Model B

- (1) 名刺大コンピュータ HDMI ポートにディスプレイ, USB ポートに, あるいは, bluetooth でキーボード, マウスを接続すれば,低スペックではあるが,一般のパソコンと同様に使える.インターネットへの接続として,有線のみでなく無線 LAN もあり, IoT(Internet of Things) の機器として適している.
- (2) ハードウェアへの接続性 Fig. 2 の本体の奥側に並ぶ 2 列のピン配列は, GPIO (General-Purpose Input/Output) というデジタル入出力ピンである.このピンを用いることで,センサからの入力,LEDやモータへの出力を直接制御できる.
- (3)OS 使用例の多くは Linux 系の Debian であるが, Windows 10 IoT Core も使える. Debian には, さま

ざまなフリーソフトウェアが使え, CやPythonなど さまざまな言語のプログラミング環境も整っている.

- (4) Mathematica 数値演算シミュレータの Mathematica の製造元の Wolfram が, Raspberry Pi 用に無料 ライセンスを提供している.これにより,一般のパソコン上で動かすより,約 10 倍ほど処理が遅いが制限なく Mathematica を使える.
- (5) Arduino との比較 Arduino は、アナログ入力端子をもつマイコンであり、ハードウェア制御には向いている.しかし、インターネット接続には別にシールドと呼ばれる拡張機器が必要であり、ソフト開発環境は別のパソコンに用意する必要がある.

これら上記の特徴により,RasPi3Bの USB ポートに WebCam を接続することで,対象に変化があるときのみに録画する知的遠隔カメラや,RasPi3Bの GPIO ポートにモータ制御指令を出させることによるモータ制御,また,これらを組み合わせることによる遠隔操作移動ロボットなどをつくることができる.

### 2. 連載記事の予定

Raspberry Pi 3 Model B の実際の使用例を Linux 系 OS をベースに 2017 年 2 月から 9 月まで月一の計 8 回連続連載する予定である.RasPi3B の情報は,インターネットや本でも多く公開されているが,当連載記事としては,より具体的な実践の手順を中心に解説する予定である.(1)RasPi3B の紹介と連載予定(今回),(2)LinuxOS のインストール,(3)OpenCV による画像処理の初歩,(4)GPIO による LED 点灯とモータ制御,(5) ネットワークの設定,(6)Mathmatica のインストールと使用,(7) 音声認識 (CMU Sphinx),(8)Fpga 拡張カード.ただし,これらの予定は変更する可能性がある.

### 3. 関連演習の予定

この連載の一部は、公益財団法人三重県産業支援センターが毎年3月にその実施を決定し10月から翌年2月まで無料で実施する(仮称)高度製造技術基礎講座の一部として演習付きで実施する予定である。随時更新する上記 URL でも確認して下さい。

# Raspberry Pi 3 Model BによるIoTの実践(第二回)

## 三重大学 機械工 松井博和 , 島中美羽 http://www.robot.mach.mie-u.ac.jp/~hmatsui/RasPi3B/

Keywords: IoT , Linux , Raspbian

#### 1. RasPi3B用OSのインストール

Raspberry Pi 3 Model B(以下 RasPi3B) を起動するための Operating System(OS) には、制限付きの MS 系の Windows 10 IoT Core を無料で用いることもできるが、非力な PC でも快適に動作する Linux 系 OS の Debian を RasPi 用に調整した Raspbian が一般的には用いら、書籍 やネット上での情報も多い.ここでは、2017/02/16 にリリースされた Raspbian Jessie with Pixel を用いる.Jessie は、Debian の最新ディストリビューションであり、Pixel は、Pi Improved Xwindows Environment、Lightweight の略で、Xwindow という GUI が軽くなるように改良したものである.インストール手順は、OS のダウンロードと microSD への書き込みである.

### 2. LinuxOSの用意

OS データは、(参 1) にブラウザでアクセスをすると、Fig. 1 の表示が出て、左側のボタンをクリックすると、 $1.5 \mathrm{GByte}$  ある  $\mathrm{zip}$  圧縮データがダウンロードできる(容量が大きいので注意)、それを  $\mathrm{unzip}$  して、2017-03-02-raspbian-jessie という OS データを得る.



Fig. 1 OS のダウンロード画面

### 3. MS上でのLinuxOSの書き込み

OS データを Windows 上で microSD(8GByte 以上) へ書き込む.ここでは, Fig. 2の Transcend の32Gの TS32GUSDHC10E を Baffaloのカードリーダ(BSCR24EU2BK) に差し, Windows10 PC に USB 接続で接続し, Win32 Disk Imager というソフトを用いて書き込む. Win32 Disk Imager は,(参 2) にアクセスし, win32diskimager-1.0.0-install.exe (Date: 2017-03-08, Size: 12.3 MB) をダウンロードしてインストールする.



Fig. 2 使用した microSD とカードリーダ

## 4. RasPi3Bの起動

Fig. 3 のように , RasPi3B に先ほどの microSD を差し , USB キーボードとマウスを , HDMI ケーブルでモニタを接続すると , 名刺大のディスクトップ PC として使用できる . Fig. 4 は , 電源をつないで OS を起動させた



**Fig. 3** RasPi3B へのモニタ, キーボード, マウスの接続

直後の画面である.

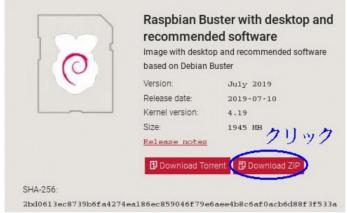


Fig. 4 Raspbian の起動後の画面

参1 https://www.raspberrypi.org/downloads/raspbian/

参2 https://osdn.net/projects/sfnet\_win32diskimager/

注2019/09/18現在では、第二回の記事の通りに実行すると、ホームページの表示が下記のものを含めて最新のDebianパッケージであるBusterの3つのパッケージサイズのものがダウンロードできるように変更されている.



記事と同じものをダウンロードするには,下記URLにアクセスし,「2017-07-05-raspbian-jessie.zip」をダウンロードして下さい.

http://ftp.jaist.ac.jp/pub/raspberrypi/raspbian/images/raspbian-2017-07-05

Raspberry Pi model 3B+を使用するには、第二回の記事のOSより新しいバージョンが必要で、OpenCVを使うことを考えると、Debian Busterよりは古いパッケージであるDebian Stretchの中で最新である「2019-04-08-raspbian-stretch.zip」を下記URLにアクセスし、ダウンロードして下さい。(フォルダ名とファイル名の日付が異なるので注意) http://ftp.jaist.ac.jp/pub/raspberrypi/raspbian/images/raspbian-2019-04-09

第二回の記事よりも新しいパッケージを用いると、起動直後にsetupのために2,3の質問を答えてと英語で尋ねてくるものがある.ここでは、「Cancel」と「Next」の内、「Cancel」を選択する.

三重大 機械工 メカトロニクス研 松井博和 mailto:hmatsui{at}robot.mach.mie-u.ac.jp Phone: (059)231-9802

## Raspberry Pi 3 Model BによるIoTの実践(第三回)

### 三重大学 機械工 松井博和

http://www.robot.mach.mie-u.ac.jp/~hmatsui/RasPi3B/

Keywords: 日本語化, wi-fi 設定

## 1. Raspbian の日本語化

Raspbian Jessie with Pixel のインストール後の調整として,日本語化,無線 LAN 接続を取り上げる.ここでは,日本語化として,(A) 表示メッセージの日本語化,(B) 日本時刻,(C) 日本語キーボードの対応,(D)wi-fi の地域設定,(E) 日本語入力を設定する.(A,B,C,D) の作業は,(E) は,(D,C) には,(D,C) でも,実行できるが,(E) は,(D,C) には,(D,C) の には,(D,C) の記定のために,(D,C) の起動画面

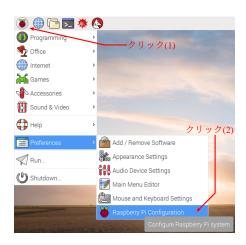


Fig. 1 メニュー

の左上のメニューバーの中,一番左の RasPi のロゴをク リック(1)する. そこから出てくるメニューの中の"Preferences"から, さらにサブメニューの"Raspberry Pi Configuration"をクリック(2)する. 出てきた"Raspberry Pi Configuration"ウィンドウの"Localisation"のタグをクリ ック (3) すと, Fig. 2 の左上図のようになる. "Set Locale", "Set Timezone", "Set Keyboard", "Set WiFi Country"をクリック (4a)(4b)(4c)(4d) して, それぞれ 4 つの ウィンドウを出し設定する. 具体的には, (4a) では,右 上図のように, "ja(Japanese)", "JP(Japan)", "UTF-8"を 選択し,"OK"をクリックする.(4b)では,左中図のよう に ,"Asia","Tokyo"を選択し ,"OK"をクリックする .(4c) では,右下図のように,"Japan","Japanese(PC98xx...)" を選択し,"OK"をクリックする.(4d)では,左下図のよ うに,"JP Japan"を選択し,"OK"をクリックする.こ れらを設定した後,クリック(5)をする.英語で再起動 するか聞かれるので,"OK"をクリックし,日本語化を有 効にする.

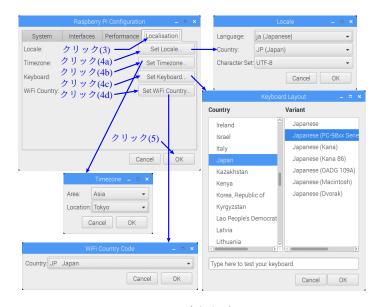


Fig. 2 日本語設定

### 2. 無線 LAN 接続

無線 LAN 接続について, Fig. 3 の左図に示すように, 起動画面の右上のメニューバーの 2 つの  $\times$  印の記号をクリック (6) する. すると, 現在接続可能な無線 LAN のアクセスポイント (AP) を表示する. 図の例では, 2 つしか AP を表示していないが, 場所によっては 20 個近く表示される. AP の 1 つである"MieC"をクリック (7) すると, 右下図が表示され, その AP の接続キー (SSID) を聞いてくる. 空白に SSID を入力し, "OK"をクリックすると, 右上図のように, 2 つの  $\times$  印の記号が, 通信可能であることを示す表示に変更される. この AP への接続は, 再起



Fig. 3 無線 LAN の接続

動しても,有効のままである.

#### 3. 日本語入力

ibus-anthy をインストールし,前章と同様に"Preferences"から設定可能であるが,補回で説明する.

## Raspberry Pi 3 Model BによるIoTの実践(補回)

### 三重大学 機械工 松井博和

http://www.robot.mach.mie-u.ac.jp/~hmatsui/RasPi3B/

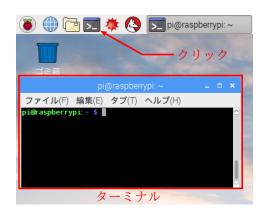
Keywords: 日本語入力,パッケージインストール, OS アップデート

### 1. 日本語入力設定

第三回の (E) 日本語入力を可能にするには , (1)LAN の接続をした上で , (2) 日本語入力パッケージのインストールをし , (3) 日本語入力設定をする . この補回では , (1) の説明は第三回でしたので , (2)(3) の説明をする .

#### 2. パッケージインストールについて

パッケージは,ターミナルを用いてインストールする.ターミナルは,画面左上にあるメニューバーの左から4番目のアイコンをクリックすると出てくる.Fig.1にアイコンをクリックしてターミナルが出てきた状態を示す.



**Fig. 1** ターミナル

このターミナルに「sudo apt-get install ibus-anthy」と打ち「Enter キー」を何回か打てば、日本語入力用パッケージ ibus-anthy と関連パッケージがインストールされる.ただし、どんなパッケージもインストールする前に、OSのアップデートをする方が良い.具体的には、ターミナルに「sudo apt-get update」と打ち「Enter キー」を打ち「sudo apt-get upgrade」と打ち「Enter キー」を何回か打つ.update は、パッケージのデータベースのアップデートであり、upgrade は、パッケージのデータベースと本機にインストールされているバージョンを比較して、新しいものがあれば自動入れ替えする.入れ替え数が多い場合など、入れ替えの是非を確認してくる.

#### 3. 日本語入力設定について

インストールした日本語を入力できるようにするためには,左上のメニューバーから,Fig.2のように,入力メソッドを選択する.前回の日本語化により,メニューが日本語されているのが分かる.1つ目の「インプット



Fig. 2 入力設定

メソッド設定」というタイトルのウィンドウが出るので,「OK(O)」を選択し,2つ目でも「OK(O)」を選択する.3つ目の  $\mathbf{Fig.}$  3 に示すウィンドウでは, $\mathbf{Fig.}$   $\mathbf{SE}$  に示すウィンドウでは, $\mathbf{Fig.}$  を選択してから「OK(O)」を選択する. $\mathbf{SE}$  であると日本語入力ができるようになる.

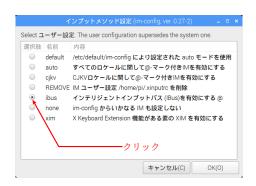


Fig. 3 入力メソッド選択

最後に  ${\bf Fig.~4}$  のように , 右上のメニューバーの「 ${\it US}$ 」をクリック (1) し , 日本語をクリック (2) すれば「 ${\it A}$  ち」となり日本語入力できる.日本語と英語の入力の入れ替えは ,  ${\it Ctrl}$  キーと  ${\it j}$  キーを同時に押すことできる.



Fig. 4 言語モード変換

注補回の記事に記述不足があります. 最後の段落の「US」のクリックの前に再起動(システム)が必要です.

三重大 機械工 メカトロニクス研 松井博和  $\underline{mailto:hmatsui\{at\}robot.mach.mie-u.ac.jp}$ 

Phone: (059)231-9802

## Raspberry Pi 3 Model BによるIoTの実践(第四回)

### 三重大学 機械工 松井博和

http://www.robot.mach.mie-u.ac.jp/~hmatsui/RasPi3B/

Keywords: 画像処理, C言語, OpenCV

#### 1. 画像処理

カメラを用いる画像処理は,工場ラインから自動運転まで,さまざまなところで用いられる古くて新しい技術である.カメラセンサーは,他のセンサーに比べて,価格が安く,センサー密度が高い特徴がある.本報告では,Raspberry Pi 3 Model B に,専用カメラモジュールでなく,Fig. 1(a) のように,一般的な UVC(Usb Video Class) の USB カメラを取り付け画像を取り込み画像表示をする.画像処理ライブラリとして OpenCV を用い,プログラミング言語には C 言語を用いる.

## 2. OpenCVとは

OpenCV は,フリーライセンスの画像処理のプログラムセット(ライブラリ)で,c/c++言語で使えるだけでなく,Java や最近流行りの python で使用することができる.ここでは,RasPi3B 上で動く RaspbianOS 上でプログラミングをし使用するが,OpenCV は,Linux 系 OS 上のみならず MS Windows や Mac OS X などのさまざまな OS に対して無料配布されている.そのため,Raspbian 上で作成したプログラムコードを改変することなしに,MS Windows 上で使用することもできる.ただし,c/c++ の場合は,再コンパイルは必要である.

## 3. OpenCVのインストール

補回で記述した方法で OpenCV 用パッケージ libopency-dev をインストールする.ターミナル上で, OS のアップグレードしてからインストールする. 具体的には,「sudo apt-get update」+「Enter キー」,「sudo apt-get upgrade」+「Enter キー」数回,「sudo apt-get install libopency-dev」+「Enter キー」数回する.これで100以上の関連パッケージもインストールされる.





(a)カメラ付きラズパイ

(b)実行時の画面表示

Fig. 1 RasPi3B の外観とその実行画面

# 4. OpenCV のサンプルコード

カメラから画像を取り込み画面に表示する c 言語プログラムを sample.c という名前で作成する.具体的には,ターミナル上で「leafpad sample.c」+「Enter キー」と入力し,下記プログラムを入力し保存する.ここでは,プログラムの説明を行間に次の行のコメント文として挿入する.画像処理する場合,Imageに対する画像処理プログラム」の箇所に画像処理プログラムを挿入する.

```
#include <cv.h>
#include <highgui.h>
main()
 unsigned char key; //キー入力の文字格納用変数
                  //カメラ情報用変数
 CvCapture *Cam;
 IplImage *Image;
                  //画像情報用変数
//O 番目のカメラ情報をメモリに展開しポインタを Cam に格納
 Cam = cvCreateCameraCapture(0);
//Video という名前のウィンドウを自動サイズで用意
 cvNamedWindow("Video", CV_WINDOW_AUTOSIZE);
 do{ // do-while文
//Cam カメラから画像情報を 1 フレームとる
   Image = cvQueryFrame(Cam);
//「Image に対する画像処理プログラム」
//Video ウィンドウに Image の画像データを表示
   cvShowImage("Video", Image);
//10msec の間のキー入力を key に代入
   key = (unsigned char)cvWaitKey(10);
 }while(key != 'q'); // 'q' が押されるまで実行
//Cam のメモリ上のデータを開放
 cvReleaseCapture(&Cam);
//Video ウィンドウの崩壊
 cvDestroyWindow("Video");
```

### 5. コンパイルと実行

ここでは,先ほど作成した sample.c の RaspbianOS 上でのコンパイルの仕方を示す.OpenCV のコンパイルは,C 言語で記述しても,c++ コンパイラを用いる必要がある.ターミナル上で,下記のコマンドを改行なしで入力すると,sample.c がコンパイルされて,sample という実行ファイルができる.

```
g++ 'pkg-config --cflags opencv' 'pkg-config
--libs opencv' -o sample sample.c
```

上記中の記号 'は , バッククオートでシングルクオート'ではない . OpenCV のソースコードは , 他の OS で共通に使えても , コンパイルの仕方は必ずしも共通ではない . 実行ファイル sample をターミナル上で下記のように実行すると , Fig. 1(b) の図中右側のウィンドウのようにカメラからの画像を画面に表示する .

./sample

<mark>注</mark>第四回の記事に記述の分かりにくい部分があります. g++のコンパイルオプションにおいて使用されている記号「`」は、バックオートという記号です.シングルクオート「'」とは異なります.

三重大 機械工 メカトロニクス研 松井博和 mailto:hmatsui{at}robot.mach.mie-u.ac.jp

Phone: (059)231-9802

# Raspberry Pi 3 Model BによるIoTの実践(第五回)

## 三重大学 機械工 松井博和

http://www.robot.mach.mie-u.ac.jp/~hmatsui/RasPi3B/

Keywords: デジタル入出力, LED 点滅

#### 1. RasPi3BのGPIOへの出力

RasPi3B の特徴の一つは,カード上に 2 列に並ぶ 40 本のピンヘッダの中に,多くの汎用目的入出力 (GPIO:General Purpose Input/Output) ピンがあることである.これらの GPIO ピンは,それぞれ独立に,1 あるいは 0 を H:3.3[V] と L:0.0[V] の 2 種類の電圧として出力でき,逆に,ピンの電圧が 3.3V 付近の H か,0.0 付近の L かを,1 か 0 として入力できる.Fig. 1 に,40 本

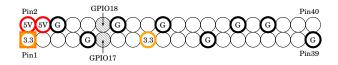


Fig. 1 RasPi3B のピンヘッダの配置

のピンヘッダのピン配置を示す.ピン番号は,一番左下のピンが 1 番ピンであり,その上が 2 番ピン,その右下が 3 番ピンとジグザグに 40 番までつけられている.図中太丸で囲まれる 12 個のピンは,電源端子を表し,2 ピンと 4 ピンにそれぞれある"5V"の記号が 5V 電源端子であることを表し,1 ピンと 17 ピンにそれぞれある"3.3"が 3.3V 電源端子を,図中 8 個ある"6"が,Ground,すなわち 6V 電源端子を表す.図中残りの細丸で囲まれる 11 ピンの GPIO17(と 12 ピンの GPIO18) だけマークして表している.

### 2. GPIO ライブラリのインストール

ここでは, RasPi3BのGPIOをC言語を用いて制御する. そのために, RasPi3BのRaspbianに wiringPi ライブラリ(http://wiringpi.com/参照)をインストールする. 具体的には, 前報告の通りインターネットを接続し,ターミナルを開いて下記の手順を実行する.

```
git clone git://git.drogon.net/wiringPi
cd wiringPi
git pull origin
./build
```

上記のコマンドの実行は,次章のプログラムのコンパイルと異なり,一度だけすれば良い.

## 3. 制御プログラム (C言語)

ここでは , Raspberry PI 3 model B の GPIO17 ピンの出力をどのように , C 言語を用いて制御するかを例を用い説明する . 実際に , GPIO をセットアップし , GPIO17 のピンを出力ピン設定にして , GPIO17 の出力電圧を High一秒 , Low 一秒 , High 一秒 , Low にして終了するプログラムを下記する .

```
/***** GPIO control for Raspberry PI ******/
#include <stdio.h>
                   //標準入出力用ヘッダ
#include <stdlib.h>
                   //標準リブ用ヘッダ
                   //UNIX 標準ヘッダ
#include <unistd.h>
#include <wiringPi.h> //GPIO 制御用ヘッダ
#define GPI017 17
//#define GPI018 18
void main()
{
 if(wiringPiSetupGpio() == -1){
   fprintf(stderr, "GPIO Setup error\n"); exit(1);
 pinMode(GPI017, OUTPUT); //GPI017を出力ピンに
 digitalWrite(GPI017, 1);//GPI017をHighに
 usleep(1000000); // 1000000 µ sec 休憩
 digitalWrite(GPIO17, 0);//GPIO17をLowに
 usleep(1000000);
 digitalWrite(GPI017, 1);
 usleep(1000000); // 1000000 µ sec 休憩
 digitalWrite(GPI017, 0);
//pinMode(GPI018, INPUT); //GPI018 を入力ピンに
//printf("GPI018=%d\n", digitalRead(GPI018));
上記のプログラムをターミナル上で「leafpad gpio.c」を
実行してファイル gpio.c を作成する.ファイル gpio.c を
下記のコマンドにより , コンパイルしコマンド gpio を作
成する.その後,コマンドgpio を./gpio で実行する.
cc -o gpio gpio.c -lwiringPi
./gpio
```

### 4. 実行

Fig. 2 に , ./gpio を実行しているときの様子を示す . 発光している緑の LED は , 内部に適切な抵抗がある LED で , 5V の電圧を LED の両足にそのまま加えても焼けない . そのため , LED の両足を RasPi3B の GPIO の 2 列の図中手前側の Pin9 のグランドと Pin11 の GPIO17 に それぞれケーブルのみで直接つなげる . LED には向きがあり , 全く点灯しないときは LED の両足を入れ替える .



Fig. 2 LED の点滅制御

注第五回の記事において、wiringPiは、gitを用いてインストールするように書かれていますが、2019年9月18日現在ではできません。その代わり、wiringPiは、下記のようにaptを用いてインストールできます。apt-get install wiringpi

三重大 機械工 メカトロニクス研 松井博和 mailto:hmatsui{at}robot.mach.mie-u.ac.jp Phone: (059)231-9802

# Raspberry Pi 3 Model BによるIoTの実践(第六回)

### 三重大学 機械工 松井博和

http://www.robot.mach.mie-u.ac.jp/~hmatsui/RasPi3B/

Keywords: インターネットアドレス (IP アドレス), 固定 IP アドレス, DHCP

### 1. 固定 IP と動的 IP

RaspberryPi を離れたところに設置し,リモートで制御しようとするとき,無線,有線を問わずに,IPアドレスを基に通信するのが一般的である.端末のIPアドレスは,一般的には,固定IPでなく,DHCP(Dynamic Host Configuration Protocol)を用いる動的 IPが用いられる.動的 IPは,接続する都度,そのネットワーク適した IPと設定をサーバからもらうため,ユーザはIP設定を気にしなくても良い利点がある.その一方,端末でなく接続先のサーバとして用いる場合,都度変動する動的 IPではなく,固定 IPを用いる.

### 2. IP アドレス

IP アドレス (Internet Protocol address) は、イン ターネットアドレスとも呼ばれるインターネットにつ ながるコンピュータがもつ実名である.マシン名(例: Lily.robot.mach.mie-u.ac.jp) は,通称であり,通信する ときには必ず IP アドレスに変換する.この変換をドメイ ン (例: robot.mach.mie-u.ac.jp) ごとに管理し, それをつ ないで,インターネット全体のマシン名と IP を両方向で 変換するのがドメインネームサーバ:DNS である. IP ア ドレスは基本的にはユニーク(世界に同じIPアドレスを もつものがない状態)に割り当てられる.多く用いられ る IP アドレスのバージョンには , v4 と v6 の 2 つがある . ここでは,使用例が特に多い IPv4 を説明する. IPv4 ア ドレスは,32bitsで表現されるインターネット通信用ア ドレスであり,通常8bitsごとの4つに分けられ,それ ぞれの 8bits を 10 進数で表現する. すなわち, IPv4 は,  $2^{32}$  つまり約 40 億個の PC に番号を割り当てることがで きる.しかし,割り当ての不均衡により現在IPv4のアド レスは枯渇してきている. そこで, 下記のグローバル IP アドレスには割り当てられていない範囲の IP アドレスを ユニークでなくても良いプライベート(ローカル)アドレ スとして用いる.この IP アドレスでローカルネットワー クを組み,その内の一台だけをグローバルネットワーク と接続することでグローバル IP の数を節約する.このた め, グローバル IP アドレスは世界に1つしかないが,同 じローカル IP アドレスのマシンは世界中に複数ある.

具体的なネットワークの例として , 三重大学機械エメカ トロニクス研究室 AI グループのコンピュータネットワー

ク Fig. 1 を示す . 図中 IP アドレスが並んで示されている のは, 一つのハードウェアが複数の IP アドレスをもつこ とを示している.並んで示される IP 間の矢印は通信がそ の方向へ流れることを示す. グループ内の 192.168.1.xxx の IP アドレスをもつ PC は,ディスクトップ PC であ る. これらの PC は,ネットマスクを 255.255.255.0 と してあり, 192.168.1.yyy の IP をもつ PC とは直接通信 する.しかし, 192.168.1.xxx の IP アドレスをもつ PC には, 192.168.1.yyy 以外の IP と通信する場合は, IP ア ドレス 192.168.1.4 の「Pine」をゲートウエイとするよ うに記述してある.具体的には「Lilv」が「Klein」と通 信するとき、「Klein」の IP アドレスは 192.168.11.110 で あり 192.168.1.yyy でないので, Pine に通信を転送する. 「Pine」には,192.168.11.zzzへの通信は,192.168.1.45の 無線 LAN ルータへ転送することが記述してあり,無線 LAN ルータである ApMieC は , その通信を 192.168.11.1 経由で 192.168.11.110 の「Klein」へ転送する.

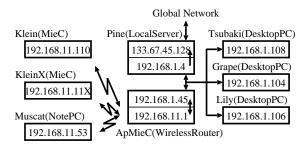


Fig. 1 AI グループのコンピュータネットワーク

#### 3. IP アドレスの固定の仕方

ここでは, RasPi3B 用の OS「Raspbian Jessie with Pixel」において固定 IP 化の設定を記述する. そのために, /etc/dhcpcd.conf を編集する. 具体的には, 前報告と同様にターミナルを開いて sudo leafpad /etc/dhcpcd.conf として, 以下のものをファイルの最後に追加する.

interface wlan0
static ip\_address=192.168.11.81/24
static routers=192.168.11.1
static domain\_name\_servers=192.168.11.1

上記では、IP アドレスを 192.168.11.81 と設定し、ネットマスクを上位 24bits すなわち、255.255.255.0 とした.また、routers で、ゲートウェイを無線ルータの 192.168.11.1 と設定した.ドメインネームサーバ (DNS) を、192.168.11.1 と設定した.一般に、DNS は無線 LAN ルータが中継している.DNS の設定がなくても、マシン 名でなく、直接 IP アドレスを用いて通信できる.

## Raspberry Pi 3 Model BによるIoTの実践(第七回)

### 三重大学 機械工 松井博和

http://www.robot.mach.mie-u.ac.jp/~hmatsui/RasPi3B/

Keywords: リモートログイン, IP アドレス

### 1. リモートログイン

リモートログインとは、現在使用しているマシンからインターネットなどの通信経路を用いて別のマシンにログインすることである.RasPiで多く用いられる Raspbian などの Unix 系 OS では、一般的に暗号化に対応した sshコマンドを用いる.このとき,ログインする側を ssh クライアントと呼び,ログインされる側を ssh サーバと呼ぶ.リモートログインをするときに,Raspbian では,初期設定(デフォルト)において,クライアント側は,そのままで良いが,サーバ側は設定変更が必要である.本稿は,第六回 (本タイトル下の URL 参照) の続編である.

### 2. ssh サーバ設定

ログインされる側である  $\sinh$  サーバ設定のためには,左上のメニューバーから, $\mathbf{Fig.}$  1 のように,「クリック (1)(2)(3)」の場所を順番にクリックし, $\mathbf{Fig.}$  2 の「 $\mathbf{Raspberry}$   $\mathbf{Pi}$  の設定」のウィンドウを開かせる.図中の四つあ



Fig. 1 「Raspberry Pi の設定」の開き方

るタグの中の「インターフェイス」をクリック (4) し,SSH の「有効」側のラジオボタンをクリック (5) し,OK」をクリック (6) する.以上で,ssh サーバの設定が完了する.補足: $Fig.\ 2$  の中にカメラの項目があり「無効」であるが,これは RasPi 専用のカメラの使用を有効にするものであり,USB カメラの使用には無関係である.また,RasPi 用の A/D,D/A 変換などのガジェットを接続するシリアル通信の一種である SPI の設定もここで有効にする.



Fig. 2 インターフェイスの設定

### 3. ssh コマンドの実行

Fig. 1 に示す「クリック (A)」の場所をクリックしターミナルを開く . IP:192.168.11.171 のマシンにリモートログインするときには , ターミナル上で下記を実行する . ssh -X 192.168.11.171

すると,初めて 192.168.11.171 のマシンにリモートログインするときには英語で接続を続けるかを (yes/no) で聞かれ,yes で答えると,サーバのもつ fingerprint をクライアント側に登録する.二回目以降は聞いてこない.パスワードを入力すると,そのターミナルでは,それ以降192.168.11.171 のマシンでの実行となる.そこでの LED 点滅やカメラ画像表示などの実行はリモートになる.



Fig. 3 2 つの RasPi3B を用いるカメラ動画表示の例

 ${f Fig.~3(a)}$  は,サーバ側が USB カメラと電源ケーブルの接続,クライアント側がモニタ,キーボード,マウスと電源ケーブルの接続である.クライアント側のターミナルから無線 LAN を介して  ${
m ssh}$  でサーバにログインし,カメラ動画表示を実行するだけで, ${f Fig.~3(b)}$  のように,クライアント側にサーバ側の動画表示ができる.

### 4. IP アドレスの調べ方

Linux マシンの IP アドレスは , 固定 IP でも動的 IP でも , インターネットに接続していれば , ターミナル上で  $\Gamma$  hostname - $\Gamma$  , を入力し , 調べられる .

# Raspberry Pi 3 Model BによるIoTの実践(第八回)

### 三重大学 機械工 松井博和

http://www.robot.mach.mie-u.ac.jp/~hmatsui/RasPi3B/

Keywords: シリアル通信 (SPI), A/D 変換器 (MCP3002)

## 1. A/D 変換のための SPI 通信

SPI 通信は,i2c 通信とともによく使われるシリアル通信の規格の一つである.送信するバイト数と同バイト数受信する.Raspberry Pi Model 3B では,40 ピンのGPIO(General-Purpose Input/Output)の一部でSPI通信ができる.RasPiでSPI通信をするには,本報告前の第七回の「ssh サーバ設定」と同様にspiを設定し,第五回の「GPIO ライブラリのインストール」をする.RasPi3BのGPIOとA/D変換チップのMCP3002との接続図をFig. 1,写真をFig. 2 に載せる.

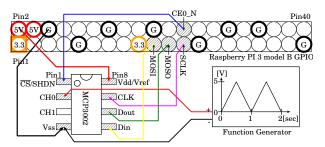


Fig. 1 RasPi3BとMCP3002との接続図



Fig. 2 RasPi3B と MCP3002 との接続写真

ここでは,A/D 変換チップの CH0 に与える入力電圧 として,ファンクションジェネレータから最低 0V から最高 5V で 0.5 秒上昇,0.5 秒下降の 1 秒周期のランプ入力を用いる.

## 2. SPI 通信する C プログラム

本プログラムは, $\operatorname{SPI}$  通信で  $\operatorname{2bytes}$  データを  $\operatorname{RasPi3B}$  から送り, $\operatorname{2bytes}$  データをガジェットから受け取るプログラムである. $\operatorname{A/D}$  変換チップの  $\operatorname{MCP3002}$  は, $\operatorname{0x68}$  と  $\operatorname{0x00}$  の  $\operatorname{2}$  バイトを送ると, $\operatorname{CH0}$  の電圧を  $\operatorname{10bits}$  で量子化し,それを  $\operatorname{2bits}$  と  $\operatorname{8bits}$  の  $\operatorname{2}$  バイトデータで返す.

```
#include <wiringPiSPI.h>
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#define SsPort 8 //GPI08
#define SpiChannel 0 //SPI チャンネル
#define BuffSize 2
#define SpiSpeed 10000000 //通信速度(100kHz)
void main()
 unsigned char spiBuff[BuffSize];//送受信用バッファ
 //SPI チャンネル初期化
 if((wiringPiSPISetup (SpiChannel, SpiSpeed)) < 0){</pre>
   fprintf(stderr, "SPISetup error\n");exit(1);
 if(wiringPiSetupGpio() == -1){//GPIO 初期化
   fprintf(stderr, "SetupGpio error\n");exit(1);
 pinMode(SsPort, OUTPUT);
                        //SSpin を出力に設定
 digitalWrite(SsPort, 1);
                        //SS 信号初期化
 fprintf(stderr, "Start SPI\n");
 for(i=0; i<200; i++ ){//2 秒間計 200 点のサンプリング
   spiBuff[0] = 0x68; spiBuff[1] = 0x00; //CH0 用
   //SPI 通信実行
   digitalWrite(SsPort, 0);
                         //通信開始
   //BuffSize のデータを送受信する.
   wiringPiSPIDataRW(SpiChannel, spiBuff, BuffSize);
   digitalWrite(SsPort, 1); //通信終了
   printf("%4.2f %d\n",0.01*i,spiBuff[0]*256+spiBuff[1] );
   usleep(10*1000);// 10msec インターバル
 fprintf(stderr,"End SPI...\n");
上記プログラムをターミナル上の下記コマンドにてコン
パイルし, 実行ファイル spi を作り, 実行する.
$ cc -o spi spi.c -lwiringPi
$ ./spi
Start SPI
0.00 234
0.01 213
197 行の省略
1.99 211
```

上記のデータをプロットするとファンジェネの出力波形とほぼ同型の  ${f Fig.~3}$  のグラフを得, ${f A/D}$  変換がうまく動作していることが分かる.

End SPI...

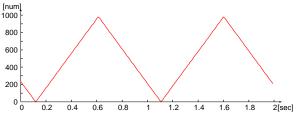


Fig. 3 実行出力のグラフ化