



loTのシステムはマイコンのI/O(入出力端子)にセンサやスイッチ,モータなどをつないだ複数の機器で構成 されます.本章では、Pythonを用いてI/Oを遠隔制御するシステムの機器構成について説明します.

IoT 向けインターネット時代に

インターネットは、1995年に商用の運用が始まり ました.当初、パソコンやワークステーションなどが 接続され、2000年に入ると、常時インターネット接 続や、モバイル・インターネット接続といったサービ スが登場し、携帯電話やPDA、テレビなどの情報機 器がインターネットに接続されるようになりました. 限られたPC(パソコン)でインターネットが利用され ていた時代から、スマートフォンなど1人1台の情報 機器で利用されるようになりました.

IoT元年と言われる2017年からは、情報機器に限らず、さまざまな機器がインターネットに接続されるよ

うになり、今後、さらに多くのモノがインターネット に接続されるようになりつつあります(図1-1). 将来 的には、人によるインターネット利用よりも、モノに よるインターネット利用のほうが上回ると予想されて います.

2 IoT の機能が満載 ラズベリー・パイ

IoTのシステムに使うハードウェアには、少なくと も通信機能とマイコンなどによる演算機能が必要です。 汎用性や開発効率を考慮すると、IP(インターネット・ プロトコル)に対応したネットワーク機能を内蔵し、 Linuxが動作するハードウェアが便利です。そこで、 本書では、教育向けコンピュータとして登場したラズ



当初はPC(パソコン)やワークステーションが接続されていたが、技術進歩とともにさまざまな機器が接続されるようにまたで、ためで

章 ラズベリー・パイの使い方

本章では、Linuxが動作するラズベリー・パイをIoTシステムのサーバやIoT機器として利用します。本章では、本書の解説で使用するラズベリー・パイの仕様や周辺機器について説明します。すでにラズベリー・パイ を使いこなしている方も一読して仕様や周辺機器を確認してください。

ラズベリー・パイをはじめるのに必要なもの ①ラズベリー・パイ本体

第

表2-1に、おもなラズベリー・パイの仕様を示しま す.本体は、ラズベリー・パイ3以降のModel Bシリ ーズが良いでしょう.廉価版のラズベリー・パイ ZeroやZero Wは、消費電力が低いことや、サイズが 小さいことが特長です.しかし、本体以外に準備しな くてはならないものが多く、プロセッサの処理能力も



開発用としては貧弱です.最初の開発はラズベリー・ パイ3以降のBシリーズで行い,2台目以降の追加時 にラズベリー・パイZeroW等を検討すれば良いでし ょう.

なお, 筆者はおもにラズベリー・パイ 3+で開発お よび動作確認を行いました. サンプル・プログラムは, ラズベリー・パイ 3, 3+, 4で動作します. Piカメラ については, ラズベリー・パイ Zero Wでも動作確認 済みです.



写真2-1 IoTのシステムに必要な機能が満載のラズベリー・パイ 4 Model B(左)とラズベリー・パイ 3 Model B+(右) IoTサーバやIoT機器を簡単に製作することができ、汎用的なIoTシステムの実験が行える

表2-1	ラズベリー・	パイシリー	-ズのおもな仕様	€比較
------	--------	-------	----------	-----

推奨	モデル名	発売時期	発売価格	無線 LAN	Ethernet	プロセッサ	DMIPS	RAM	USB	HDMI	消費電流** (平均~最大)
0	ラズベリー・パイ 4 Model B	2019年 6月	\$35~\$75	0) Gigabit	64bit ARM Cortex-A72	7,080	1GB∼ 8GB	4 ports	Micro 2 ports	570mA~3A OFF時:22mA [USB Type C]
0	ラズベリー・パイ 3 Model B+	2018年 3月	\$35	0) Gigabit	64bit ARM Cortex-A53	3,200	1GB	4 ports	標準	420mA~2.5A OFF時:100mA
0	ラズベリー・パイ 3 Model B	2016年 2月	\$35	0	○ 10/100	64bit ARM Cortex-A53	2,763	1GB	4 ports	標準	300mA~2.5A OFF時:80mA
\bigtriangleup	ラズベリー・パイ 2 Model B	2015年 2月	\$35	×	○ 10/10	32bit ARM Cortex-A7	1,710	1GB	4 ports	標準	250mA~1.8A OFF時:70mA
\bigtriangleup	ラズベリー・パイ Zero W	2017年 2月	\$10	0	×	32bit ARM 1176	1,250	512MB	Micro 1 port	Mini	130mA~0.3A OFF時:40mA
Ι	ラズベリー・パイ Zero	2015年 11月	\$5	×	×	32bit ARM 1176	1,250	512MB	Micro 1 port	Mini	80mA~0.2A OFF時:20mA

※平均消費電流とOFF時消費電流は筆者による実測値(接続機器なし・GUIなし)

7

N

N

0)



Pythonのプログラムをラズベリー・パイ上で作成する方法について説明します. Python 言語は他の言語に 比べて少ないルールでプログラムが書けます. また, 1 つのプログラムを他用途のプログラムに簡単に応用す ることもできます. 少ないルールを使いこなしてプログラムを作成するコツを解説します.

1) 学習用プログラムをダウンロードする

本書でおもに使用するハードウェアは、(前章でセ ットアップした)ラズベリー・パイです. IoTシステ ムなる機器を想定すると、Linux上で学習/開発した ほうが、活用範囲が広がるからです. ラズベリー・パ イがない場合は、Microsoft Windowsが動作するPC にCygwinをインストールして学習/開発することも できます.

はじめに、ラズベリー・パイのLXTerminal上で下 記のコマンドを入力し、本書で解説する学習用プログ ラムをダウンロードしてください(図3-1). なお、前 章でダウンロード済みであれば再実行する必要はあり ません.

cd 쉬 git clone https://github.com/bokunimowakaru/iot 쉬

一般的に、プログラムを開発するにはIDE(統合開

発環境)を使用します.まずは試しに、ラズベリー・ パイ上にあらかじめインストールされている Thonny Python IDEを、Raspbianの[メニュー]→[プログラ ミング]→[Thonny Python IDE]の順に選択して起動 してみてください.起動後、[Load]アイコンをクリ ックし、ダウンロードしたサンプル・プログラムiot/ learning/example01_print.pyを選択して開きます、プ 3

0

pi@raspberrvpi:~\$ cd							
pi@raspberrypi:~ \$ git clone							
https://github.com/bokunimowakaru/iot							
Cloning into 'iot'							
remote: Enumerating objects: xxxx, done.							
remote: Counting objects:							
100% (xxxx/xxxx), done.							
remote: Compressing objects:							
100% (xxxx/xxxx), done.							
~~ 詳細表示内容を省略 ~~							
pi@raspberrypi:~ \$							

図3-1 LXTerminal から git clone コマンドを使い,サンプ ル・プログラムを GitHub からダウンロードする



図 3-2 ラズベリー・パイで Thonny Python IDE を起動し、プログラムを実行したときのようす プログラムを作成する際にIDE(統合開発環境)を使用すると便利

第4章 ラズベリー・パイ用 Pythonプログラム ~GPIO制御編~

この章ではラズベリー・パイでデータを送受信をするPythonプログラムを解説します. Wi-FiのUDP通信やTCP通信を使ったスイッチON/OFF情報の送信,温度値送信,リモートLチカ,チャイム制御,UDP通信の受信,TCP通信の受信方法について,サンプル・プログラムを使って解説します.次の章では,ESPマイコンと組み合わせたシステムのPythonプログラムを解説します.

プログラムの解説前に、ハードウェアの準備をします、ブレッドボードで「GPIO実験ボード」を作り、ラズベリー・パイのGIPO端子に接続します。

この章で解説するPythonプログラムは、このラズベリー・パイとGPIO実験ボードの組み合わせで動作します.



Pythonのプログラムで、ラズベリー・パイのI/O ポートを制御するプログラムを解説します、プログラ ム解説の前に、ラズベリー・パイのI/Oポートを制御 しやすいように、ブレッドボードを使ってスイッチ、 圧電スピーカ、フルカラー LED を配線します、本書 では、これを「GPIO実験ボード」と呼びます(写真 4-1)、

ラズベリー・パイのGPIO(General-purpose input/ output; 汎用入出力) 端子は, 汎用的に利用可能なディ ジタル入出力端子です. この端子は3.3VのCMOSデ ィジタル入出力仕様で入力時は0VでLow, 3.3Vで High が得られ, 出力時はLowで約0V, Highで約 3.3Vを出力します. 出力時の最大電流は16mAです. 高効率タイプのLEDであれば, 電流を絞ることで直 接接続して発光させることができます. 複数の端子か ら出力するときは, 出力の合計電流を50mA以内にし



写真4-1 ラズベリー・パイに接続した GPIO 実験ボード ラズベリー・パイの GPIO 端子にフルカラー LED, 圧電スピーカ, タクト・スイッチを接続している

ます.

図4-1のラズベリー・パイの40本の端子には、17 本のGPIO端子と、I²C、UART、SPI専用インターフ ェース端子、電源用端子が備わっています。GPIO端 子が17本で足りないときは、他の端子を使用するこ



写真4-2 GPIO実験ボードの配線 ブレッドボードの縦のラインは内部で導通 している. タクト・スイッチを挿す向きに 注意



loT機器が送信するデータをラズベリー・パイで受信,表示,保存する宅内サーバ用のPythonプログラムを解 説します.本章共通で使用する送信機は、章の後半で概説するESP32マイコンで製作します.



Python プログラム⑨ ラズベリー・パイで温度を UDP 受信

ESP32マイコン(子機)がWi-Fi送信した温度値を受信して表示するラズベリー・パイのサーバ(親機)用 Pythonプログラムを作成します(図5-1).

データ送信側 ESP32 マイコン(子機)

温度の値をWi-Fi送信する送信機を本稿ではIoT温 度計と呼ぶことにします. ESP32マイコン開発ボード T-Koalaの電池端子にアルカリ乾電池3本を直列に接 続して製作しました. ESP32マイコンのボードなら他 のものも使用できます. ただし端子番号は違うことが あるので気をつけてください.

送信用 ESP32 マイコンとプログラムの詳細はこの 章の後半 p.74 ~ を参照してください.

ESP32マイコンにインストール後, PCやスマート フォンでアクセスして, ブラウザ画面の[Wi-Fi 設定] から AP+STAモード(またはSTAモード)でLANに 接続すると送信した温度値を同じLANに接続したラ ズベリー・パイで受信できるようになります. 初期設 定では, 温度値を30秒ごとに受信します. example21_rx_temp.py(**リスト5-1**)は、ラズベリ ー・パイで動作するPythonのプログラムです、UDP を受信するプログラムp.92**リスト4-7**のexample16_ udp_logger.pyに、IoT温度計のデータであることを 確認する関数check_dev_nameと、UDPデータの中 から値を抽出する関数get_valを追加しました、プロ グラム example21_rx_temp.pyの処理のようすを図 **5-2**に示します.

● 受信側ラズベリー・パイ用 Python プログラム

プログラムの処理の流れを解説します.

- ① IoT温度計のデバイス名があらかじめ設定したもの と一致するかどうかを確認する関数check_dev_ nameを定義します.定義だけなので,起動時は実 行されず,後の手順⑧から呼び出されたときに実 行します.
- ② 関数check_dev_nameの引き数sが、表示可能な文字列かつ7文字、かつ先頭4文字がtemp、かつ6文字目が「_」のとき、IoT温度計からのデータであると判断し、手順③を実行します。条件を変更すれば、他のデータを受信することもできるようになります(例:example21 rx_temp_m.py).
- ③処理②のデバイス名がIoT温度計と1致もた。



本章では、Bluetooth LEのビーコン(アドバタイジング)送信に対応したIoT センサを用い、センサ値データを ラズベリー・パイ内蔵のBluetoothモジュールで受信して表示するプログラムや、CSV形式での保存方法、クラ ウド・サービスに転送してグラフ化表示する方法について紹介します(図6-1).

先にBLEの送信機を準備し、その後で受信側のラズベリー・パイのPythonプログラムを説明します。



送信機の準備 1 ESP32 マイコンで BLE 送信

BLE送信はESP32マイコン

Espressif Systems 製 ESP32-DevKit C, TTGO 製 T-Koala, CQ出版社製 IoT Express など ESP32 マイコ ンを搭載するマイコン・ボードに、プログラム(iottemp-ble-esp32)を書き込むことで、温度センサ値を BLEで送信できます(写真6-1).

ESP32マイコンにプログラムを書き込むには、ESP32 マイコン開発ボードをラズベリー・パイに接続し、フ オルダiot/iot-temp-ble-esp32/target/内のiot-temp-bleesp32.shを実行します.

※内蔵温度センサの測定精度は良くないので実験用として使 用. できれば外付けセンサの利用がベター.



写真6-1 ESP32マイコン内蔵の温度センサを使いBluetooth LEで温度値を送信する

あらかじめ, プログラムiot_exp_temp_bleを書 T-Koala

第一章

MicroPythonプログラム (micro:bitで試す)

MicroPythonは、Pythonをベースにしつつ、ARM Cortex-Mシリーズ等のマイコン向けに開発されたプログラミング言語です。Pythonの主要な機能に加え、マイコンのハードウェアを制御する機能が搭載されており、IoT 機器向け言語として注目されています。

本章では、BBC micro:bitV1またはV2を使ってMicroPythonによるワイヤレス通信を試します.



● micro:bitでMicroPythonをはじめる

マイコン・ボードBBC micro:bit(以下 micro:bit)を 使えば、とても簡単にMicroPythonによるプログラミ ングをはじめることができます(写真7-1). micro:bit をラズベリー・パイ(またはWindows/macOS/Linuxを 搭載したPC)のUSB端子に接続すると、USBメモリ のようにドライブとして認識されます(図7-1①). ド ライブ名 MICROBITを確認し、フォルダを開くとフ ァームウェアの情報が書かれたDETAILS.TXTと、 Webサイトへのリンク MICROBIT.HTM(図7-1②)の 2つのファイルが入っています. リンクのほうを開き ます.

Webサイトmicro:bitにアクセスするので、言語メ ニューで[日本語]を選んでから(図7-1③),[はじめ よう]を選択すると、説明書が日本語で表示されます (図7-1④).

MicroPythonを始めるには、図7-2の[ステップ2: プログラムする]で[Pythonエディタ]を選択してから (図7-2⑤), [プログラムする]を選択してください(図 7-2⑥).

Python エディタ(図7-3)には、機能アイコンと Python プログラムのソースリストが表示されます. 機能アイコンはPython エディタのバージョンによっ て、表示が異なります. V1.1とV2.0のボタンの違い を p.98 のコラムに記します. V2.0 のほうが多機能で すが、基本的な使い方は同じです. プログラムを編集 し、機能アイコンの一番左にある[download]をクリッ



写真7-1 プログラミング言語 MicroPython に対応した micro:bit

micro:bit はイギリスの放送局BBCが教育用に開発したARM Cortex-MOを搭載するオープン・ソース・ハードウェア. イギ リスの7年生(中等教育の初年度)の子供たちに無償で配布され ている

クすると(図7-4⑦), BBC micro:bit用のファイル microbit.hexをダウンロードすることができます.一 度, ラズベリー・パイまたはPCにダウンロードして から, ドライブMICROBITにコピーします. ブラウ ザの[名前を付けて保存]機能(図7-4⑧)でドライブ MICROBITに直接ダウンロードするこ



STマイクロエレクトロニクス製STM32F767ZIは、LAN接続が可能なEthernetインターフェースを内蔵した32ビットARM Cortex-M7ベースのSTM32マイコンです.LAN経由でインターネットに接続できるので、IoT搭載マイコンと言っても良いでしょう.また、同マイコンを搭載したSTM32マイコン開発ボードNUCLEO-F767ZIには、写真8-1のようにLAN接続用Ethernetコネクタが標準装備されているので、手軽にMicroPythonを使ったIoT機器を製作することができます.

NUCLEO-F767ZIへ MicroPython のファームウェアを書き込む

STM32マイコンのEthernet対応ファームウェアは, 公式配布されている MicroPython のソースに含まれ ています.しかし、今のところビルドしたものは配布 されておらず、またビルド手順も複雑なので、筆者が 準備したスクリプトを使用してインストールしてくだ さい.

ファームウェアをSTM32マイコン開発ボード NUCLEO-F767ZIに書き込むには、CN11の7番ピン (BOOT0)とCN8の7番ピン(3V3)をジャンパ・ワイ ヤで接続した状態で、CN13のMicro USBをラズベリ ー・パイのUSB端子に接続し、CN1の電源用Micro USB端子へ5Vを供給します(**写真8-2**).

ダウンロードしたiotフォルダ内のmicropython/ nucleo-f767zi/install.shを実行すると必要なファイル



写真8-2 MicroPythonのファームウェアを書き込むため の構成例

STM32マイコン開発ボードNUCLEO-F767ZIのBOOTOをHレ ベルに設定し、CN1から電源を供給するとSTM32マイコンが ファームウェア書き換えモードで起動する



写真8-1 LAN 接続用 Ethernet インターフェース内蔵の STM32マイコンを搭載した NUCLEO-F767ZI LAN 接続用 Ethernet コネクタを装備しているので,手軽に Micro Python を使った IoT 機器が製作できる



写真8-3 ファームウェア書き換えモードに変更するため のBOOT0部の拡大図

STM32マイコンNUCLEO-F767ZIのCN11の7番ピン(BOOTO) のスルー・ホールから、CN8の7番ピン(3V3)をジャンパ・ア イヤで接続する



本章では、BLE通信でセンサ・データを送信するプログラムをMicroPythonで作成します。 ラズベリー・パイPicoに内蔵された温度センサや温湿度センサ・モジュールから取得したセンサ値をBLEモ ジュールRN4020で送信します。

プログラム開発やBLEの受信には従来のラズベリー・パイ4(または、400、3+、3)を使用します. ラズベリー・パイPicoでBLE送信した測定値をラズベリー・パイ4で受信して表示します.



RN4020 + ラズベリー・パイPico example03_rn4020.py

図 9-1 本章で製作する BLE ワイヤレス・センサの一例 ラズベリー・パイ Picoで測定した温度値を BLE で送信し、ラズ ベリー・パイで受信・表示する

ラズベリー・パイ Pico RP2040

2021年に発売されたラズベリー・パイ Pico は、従 来のシングルボード・コンピュータのラズベリー・パ イとは異なり、ユーザ・インターフェースやモニタ出 力を使った OS 機能に対応していない組み込み用のマ イコン・ボードです.より小型化、低価格化、省電力 化が求められる組み込み向けに、ラズベリー・パイ財 団が独自に開発したデュアルコア ARM Cortex M0+ マイコン RP2040を搭載しています.

同じラズベリー・パイのシリーズですが、使い方は 大きく異なります. プログラムを作成するには従来の 通常のラズベリー・パイ4(または,400,3+,3,2,1) またはパソコンが必要です.

BLE モジュール RN4020

本章で製作する BLE 通信部には、マイクロチップ・ テクノロジー社(Microchip Technology Inc.・米国) の RN4020を搭載した Bluetooth (BLE) モジュール AE-RN4020-XB(秋月電子通商製)を使用し、**図9-2**のよう にUARTと電源をラズベリー・パイPicoに接続します.

必要な機器

表9-1にワイヤレス・センサの実験に必要な機器を 示します. 細ピンヘッダは, ラズベリー・パイ Pico をブレッドボードに接続するのに使用します.

BLEモジュール用のピンヘッダは、AE-RN4020-XB に付属しています.マイクロUSBケーブル(microB タイプ)は、ラズベリー・パイPicoをプログラム作成 用のラズベリー・パイ(またはパソコン)に接続するの に使用します.ピンヘッダの取り付けにはハンダ付け 作業が必要です.

第10章 ラズベリー・パイとPythonで loTシステム開発入門

本章では、ラズベリー・パイがIoT機器を管理するIoTシステムを紹介します.これまでは、IoT機器単体を 説明してきましたが、本章ではIoT機器を組み合わせて、IoTのシステム化を図ります.ラズベリー・パイは、 IoTサーバとして各IoT機器を管理します.

IoT ボタンでチャイム音. 呼び鈴のシステム例

図10-1の例は、IoTボタンでチャイム音を鳴らす呼び鈴システムです。IoTサーバがIoT機器からの情報を取り扱う実験を行います。IoTサーバ(親機)は、複数のIoTボタン(子機)が送信した通知を、Wi-Fiアクセス・ポイント経由で受信してピンポン音を出力します。第4章で製作したexample20_iot_notifier.pyでも同様のことが行えます。

子機となる IoT ボタンは,GPIO実験ボード(第4章 で製作)を搭載したラズベリー・パイ上で example14_ iot_btn.pyを実行する,もしくは IoT Sensor Coreを 書き込んだ ESP32 マイコンにタクト・スイッチを追加 し,[センサ入力設定]の[押しボタン]で,[PingPong] を選択したものなどを使用することができます.

親機となるチャイム音を出力するIoTサーバは,第 4章のGPIO実験ボードを搭載したラズベリー・パイ 上でexample28_chime_btn.pyを実行します(あるい は、1台のラズベリー・パイ上で2つのLXTerminal を起動し、それぞれのLXTerminal上で子機と親機の プログラムを動かすことで、開発時のハードウェアの 台数を減らすこともできる). 各LXTerminalでの実 行例を図10-2に示します.

以下に、**リスト10-1**のIoT版チャイム呼び鈴のサ ンプル・プログラム example28_chime_btn.pyの主要 な処理内容について説明します.

- チャイム音を鳴らすための関数chimeを定義します. 定義だけなので,起動しただけでは実行されません. 実行するには後述の処理⑥などから呼び出す必要があります.
- 変数keyにPingが代入されていたときに554Hzの 音を1秒間,鳴らす処理です.
- 変数keyにPongが代入されていたときに440Hzの 音を0.3秒間,鳴らす処理です.
- ④ UDP 受信用のソケットへの接続に成功した場合に,



第4章で製作したGPIO実験ボード搭載のラズベリー・パイと親機となるチャイム呼び鈴サーバでシステムを構成する

^{第11章} ラズベリー・パイとPythonで loT音声認識入門

スマート・スピーカは,一般家庭に普及した IoT 機器のひとつです.「OK,Google」や「Alexa」などウェイ ク・ワードに続いて話した音声は、クラウド上のサーバに送信され、音声認識とAIで回答が生成され、スマート・ スピーカで返事が再生されます.

本章では、ラズベリー・パイ上で動作する音声認識ソフトと、クラウド上で動作する音声認識エンジンを使 う2種類の方法で、スマート・スピーカ・ライクなIoT音声ユーザ・インターフェースを製作します.

1 Google AIY Voice Kit と **Julius**の違い

音声認識にはクラウド・サービスを利用する方法と、 手元のPC上で認識する方法があります. ここでは. クラウド・サービス[Google Cloud Speech-to-Text] を使用する方法と、オープン・ソースの「大語彙連続 音声認識エンジン Julius」を使用する方法を説明しま す. それぞれの特徴の違いを表11-1で比較し、要件 に合った方法を選択すると良いでしょう.

Google Cloud Speech-to-Text

クラウド・サービスを使った音声認識の利点は、膨 大な音声辞書を利用し、また利用者から得た学習結果 から、確度を高めることができる点です. Google Cloud Speech-to-Textを始めるには、例えば、写真 **11-1**のようなGoogle AIY Voice Kitを使用します. SDカード用イメージ・ファイルも提供されているの で、インストールは簡単です。ただし、クラウドに接 続するには、Google Cloud Platformへの登録や認証 キーの設定が必要です. どちらかといえば, IoT 機器 側よりもクラウド・サービス側の開発に興味のある人 に向いています.

執筆時点では、ラズベリー・パイ本体やマイクロ

SDカードが付属しない Version 1と、それらが付属す る Version 2 が販売されています。組み立てや設定に 関する説明書(ウェブ)は英文ですが、写真を多用して いるので、英語があまり得意でなくても製作できるで しょう.

Google AIY Voice Kit V1

https://aiyprojects.withgoogle.com/voice-v1/ Google AIY Voice Kit V2 https://aivprojects.withgoogle.com/voice/

大語彙連続音声認識エンジン Julius

USBマイクロホンを接続したラズベリー・パイに Julius をインストールすることで、Google AIY Voice Kitよりも安価に音声認識を行うことができます. イ ンターネットに接続していない状態でも使用すること ができるので、クラウド・サービス料が不要なうえ、 プライバシーの確保も可能です。例えば、一般的なク ラウド・サービスを利用するスマート・スピーカでは、 ユーザはウェイク・ワードで呼びかけ、ウェイク・ワ ードを認識したときだけ音声をクラウドへ送信するこ とで、ユーザが意図しない音声をクラウドへ送信して しまう頻度を低減しています. Juliusであれば、ウェ イク・ワードなしで運用することもできるでしょう.



図11-1



loTの世界では,単にモノがネットワークに接続できれば良いというものではありません.さまざまなモノが 連携して動作することにより,新しいモノの価値が生まれます.

本章では、ラズベリー・パイが複数の機器やインターネットと連携して動作する実用的なIoTシステムを製作 します.ここで紹介したPython プログラムを少し修正すれば、さまざまなIoT応用システムへと展開すること ができるようになります.

インターネット照る照る坊主で LED を制御

インターネットと連携したシステムの第一歩として、 インターネットから得た天気情報に応じて、カラー LEDの発光色を制御してみましょう.プログラムを 実行すると、インターネットから天気情報を取得し、 予報が晴れのときは赤色に、曇りのときは緑色に、雨 や雪のときは青色にLEDを光らせます.また、晴れ のち曇りだと、赤色と緑色を混ぜた黄色に制御します.

ハードウェアは, 図12-1のようにインターネット 接続されたラズベリー・パイに, 第4章で製作した IoT実験用IOボードを接続して作成します.

ラズベリー・パイ上で実行するソフトウェアは、**リ** スト12-1の example34_led3_wea.py です. iot フォル ダの learning フォルダ内に収録しました.以下にプロ グラムの主要な動作を説明します.

- 変数city_idに天気情報向けの地域ID(1次細分区定 義表)を代入します.
- カラーLEDを接続したラズベリー・パイのGPIO を初期化し、出力に設定します。

- 天気情報をインターネットからJSON形式で取得し、辞書型変数res_dictに代入します。
- ④ 変数telopへ天気の内容(晴れ,曇りなど)を代入する処理部です。辞書型変数res_dictから項目forecastsを抽出し,forecasts内の配列の先頭に含まれる項目 telopを抽出し,変数telopに代入します。
- ⑤ 変数telop内の文字列内に[晴]が含まれるときに変数colorに赤色を混合します.演算子[]」は論理OR 演算を行います.赤の色番号は1なので、変数 colorの最下位ビットを1にします.同様に[曇]だ と緑、[雨]または[雪]だと青を混合し、光の3原色 の混合を行います.
- ⑥ 変数colorの値に応じたLEDの制御を行います.最下位ビットの赤色、次のビットの緑色、その次の青色に合わせてGPIOの出力を設定します.

ここでは、カラー LED を制御しましたが、プログ ラムを改造すれば、天気情報を利用した機器の制御も 可能です。例えば、変数 telop 内の文字列内に「雨」が 含まれていない状態から、「雨」が含まれるように変化 したときに[もうすぐ雨になります]とアナウンスを音 声出力する応用が考えられます。



図 12-1 インターネット照る照る坊主でLEDの発光

インターネットから天気情報を取得し,予報が 晴れのときは赤色に,曇りのときは緑色に,雨 や雪のときは青色にLEDを光らせる