すぐにPythonを体験できるサンプル・プログラムは
サポート・サイトからダウンロード!

トランジスタ技術 増刊
ボード・コンピュータ・シリーズ

ラズベリー・パイでI/O制御 &
Pico, micro:bit, STM32でクラウド通信

Pythonで作るIoTシステム プログラム・サンプル集

国野 亘 著

CQ出版社

第1章 Python I/O制御プログラミングの前に

IoT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．

1 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，非PCでの利用の割合が相対的に上回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

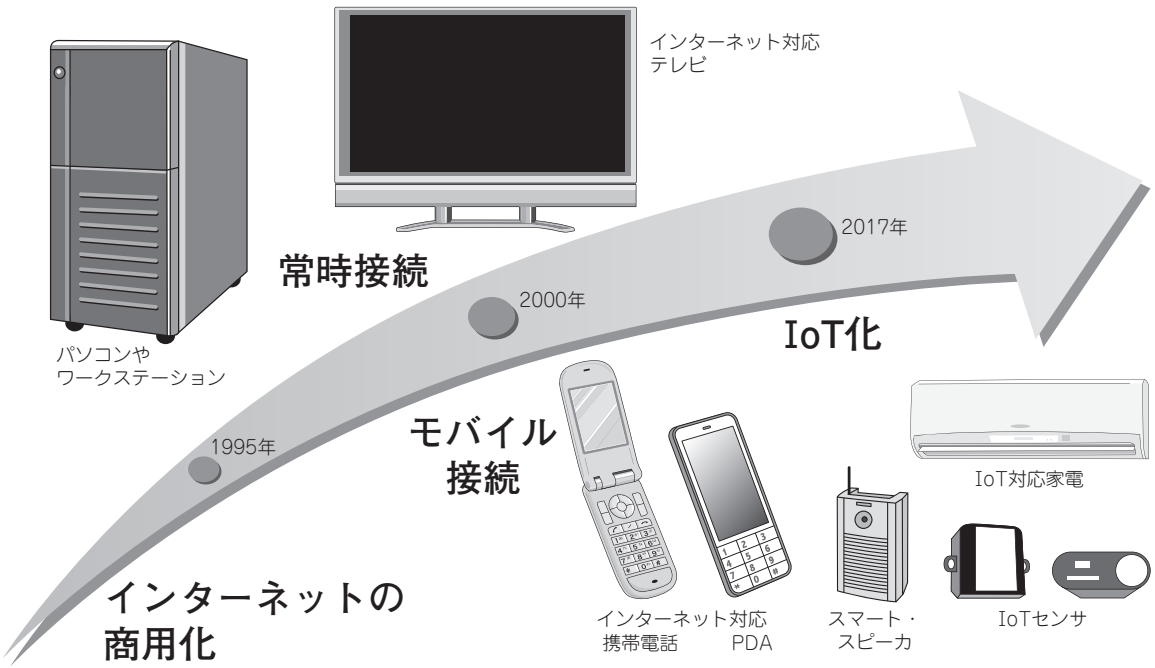


図1-1　インターネットへ接続される機器の変化
当初はPC(パソコン)やワークステーションが接続されていたが，技術進歩とともにさまざまな機器が接続されるようになってきた

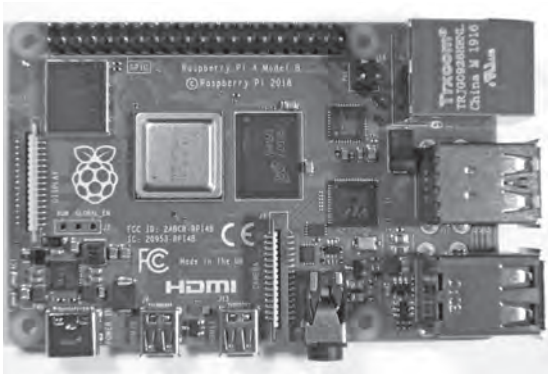
第2章 ラズベリー・パイの使い方

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．

1 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

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Zero W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．



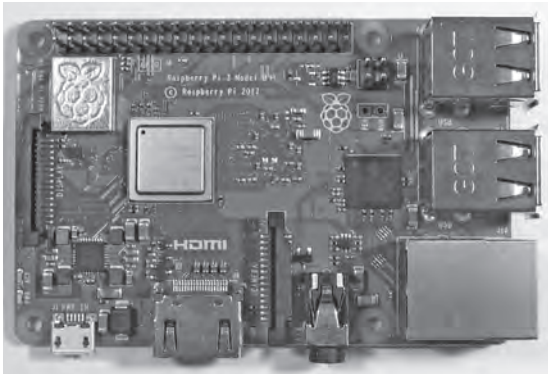


写真2-1　IoTのシステムに必要な機能が満載のラズベリー・パイ 4 Model B(左)**とラズベリー・パイ 3 Model B+**(右)
IoTサーバやIoT機器を簡単に製作することができ，汎用的なIoTシステムの実験が行える

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）
○	ラズベリー・パイ 4 Model B	2019年6月	$35～$75	○	○ Gigabit	64bit ARM Cortex-A72	7,080	1GB～8GB	4 ports	Micro 2 ports	570mA～3A OFF時：22mA［USB Type C］
○	ラズベリー・パイ 3 Model B+	2018年3月	$35	○	○ Gigabit	64bit ARM Cortex-A53	3,200	1GB	4 ports	標準	420mA～2.5A OFF時：100mA
○	ラズベリー・パイ 3 Model B	2016年2月	$35	○	○ 10/100	64bit ARM Cortex-A53	2,763	1GB	4 ports	標準	300mA～2.5A OFF時：80mA
△	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
△	ラズベリー・パイ Zero W	2017年2月	$10	○	×	32bit ARM 1176	1,250	512MB	Micro 1 port	Mini	130mA～0.3A OFF時：40mA
－	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)

第3章 ラズベリー・パイでPython入門 ～基礎編～

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↵
git clone https://github.com/bokunimowakaru/iot↵
```

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

```
pi@raspberrypi:~ $ 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

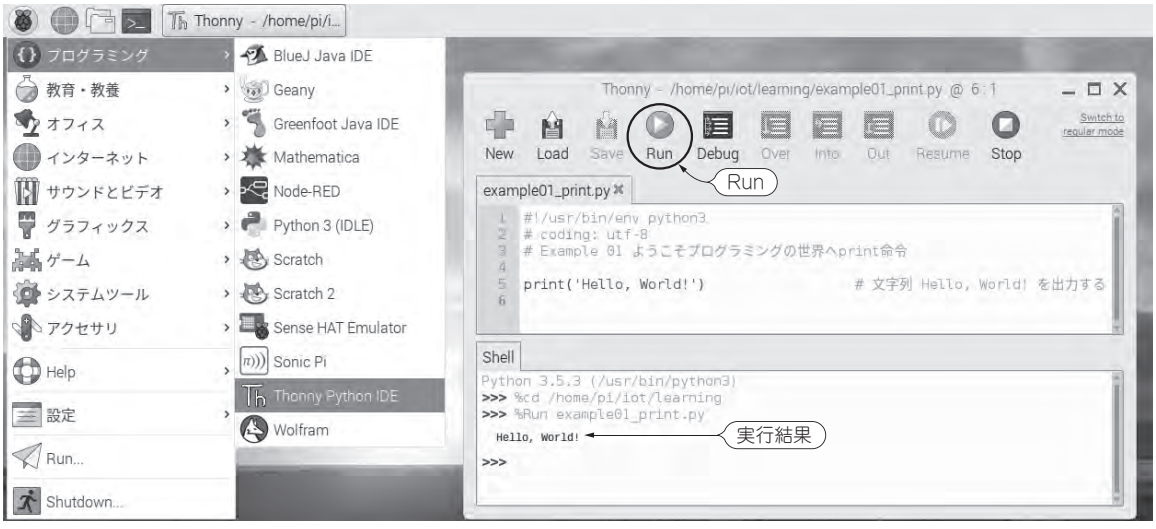


図3-2　ラズベリー・パイでThonny Python IDEを起動し，プログラムを実行したときのようす
プログラムを作成する際にIDE(統合開発環境)を使用すると便利

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．

ハードウェアの準備
ラズベリー・パイ用 GPIO 実験ボード

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以内にします．

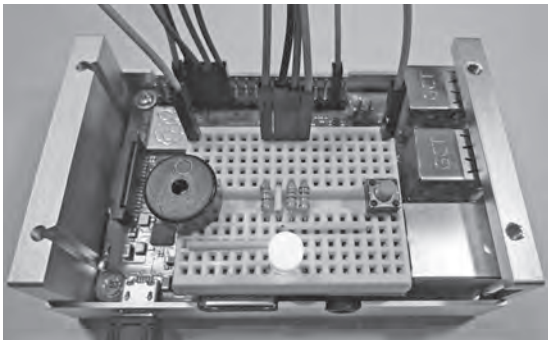


写真4-1　ラズベリー・パイに接続したGPIO実験ボード
ラズベリー・パイのGPIO端子にフルカラーLED，圧電スピーカ，タクト・スイッチを接続している

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

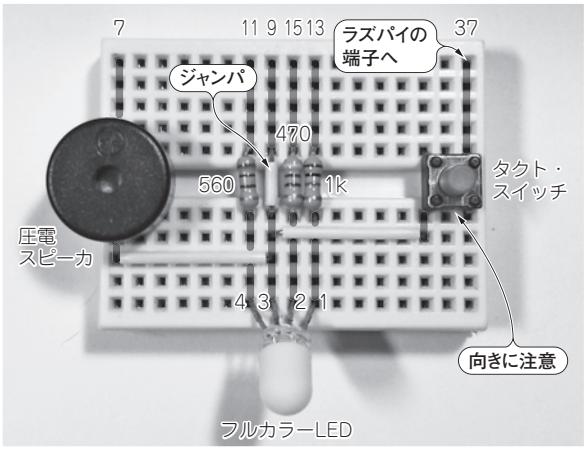


写真4-2
GPIO実験ボードの配線
ブレッドボードの縦のラインは内部で導通している．タクト・スイッチを挿す向きに注意

第5章 ラズベリー・パイ宅内サーバ用データ受信プログラム

IoT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．

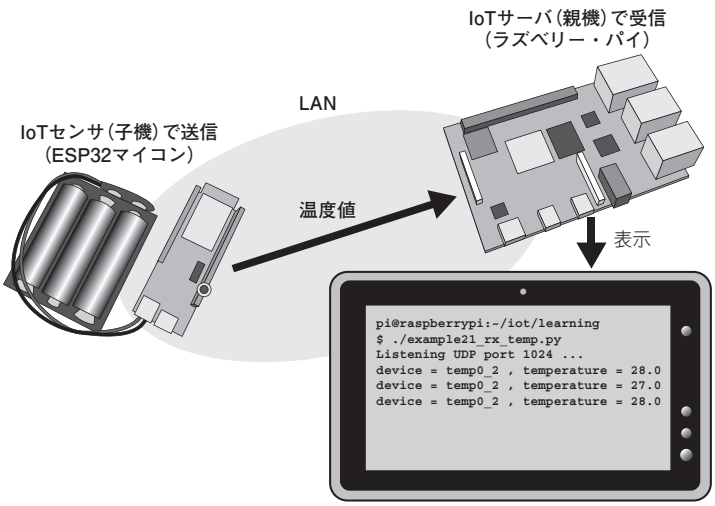


図5-1
IoT温度計が送信する温度値をラズベリー・パイで受信し表示する
ESP32マイコン内蔵の温度センサの値を送信するIoT温度計から温度値を，ラズベリー・パイで受信表示したり，ファイルを保存したりするプログラムを作る

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．

● 受信側ラズベリー・パイ用Pythonプログラム

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．

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温度計と合致したこと

第6章 ラズベリー・パイでBluetooth LEを受信するPythonプログラム

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．

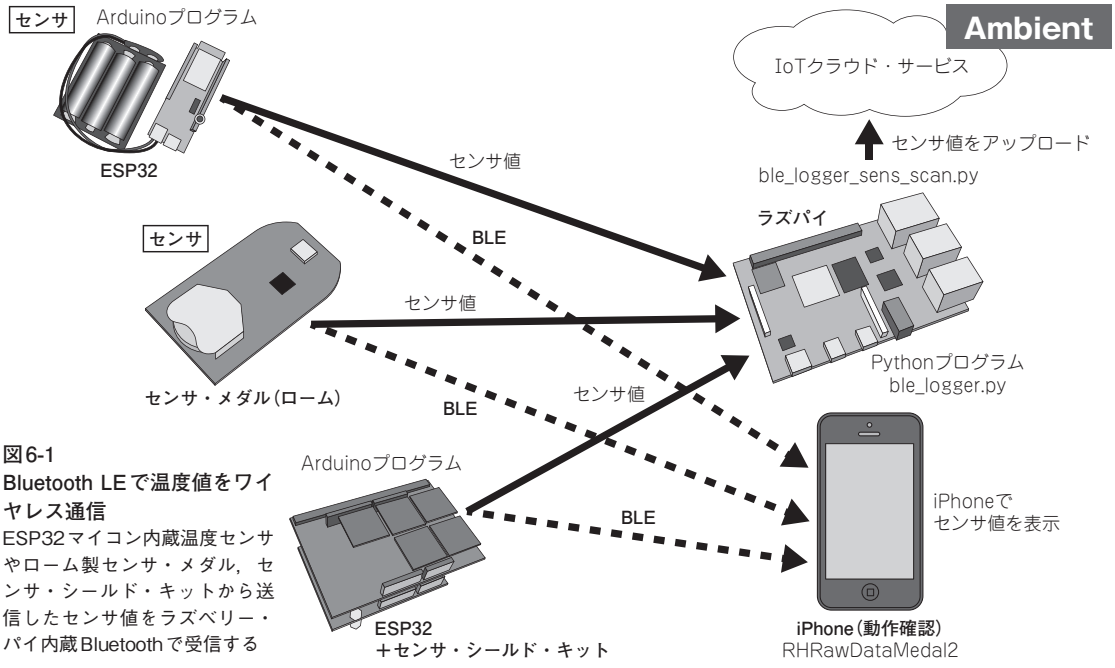


図6-1
Bluetooth LEで温度値をワイヤレス通信
ESP32マイコン内蔵温度センサやローム製センサ・メダル，センサ・シールド・キットから送信したセンサ値をラズベリー・パイ内蔵Bluetoothで受信する

ESP32

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(iot-temp-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，フォルダiot/iot-temp-ble-esp32/target/内のiot-temp-ble-esp32.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．

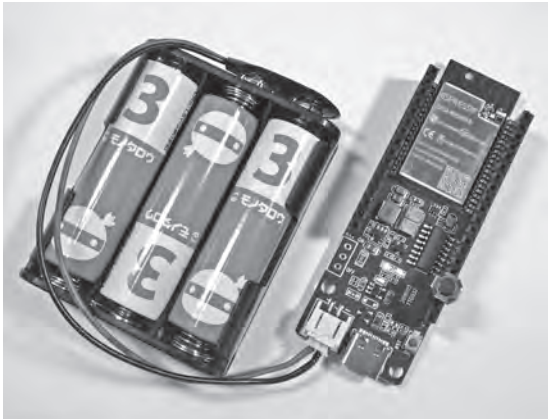
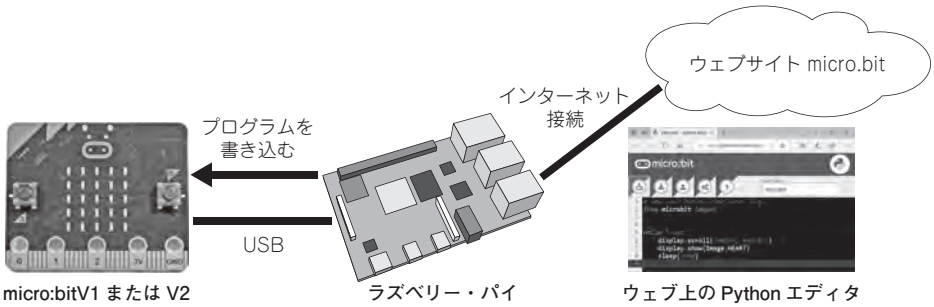


写真6-1　ESP32マイコン内蔵の温度センサを使いBluetooth LEで温度値を送信する
あらかじめ，プログラムiot_exp_temp_bleを書き込んだTTGO T-Koala

第7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7-1**①）．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7-1**②）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7-1**③），［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7-1**④）．

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7-2**⑤），［プログラムする］を選択してください（**図7-2**⑥）．

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］をクリックすると（**図7-4**⑦），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7-4**⑧）でドライブMICROBITに直接ダウンロードすることもできます

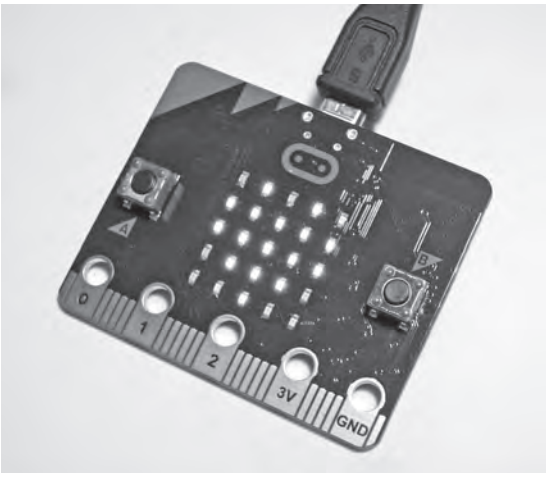


写真7-1　プログラミング言語MicroPythonに対応したmicro:bit

micro:bitはイギリスの放送局BBCが教育用に開発したARM Cortex-M0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

第8章 STM32マイコン用 MicroPythonプログラム

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．

1 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

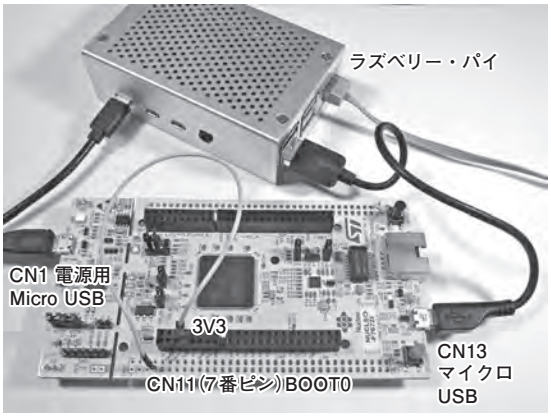


写真8-2　MicroPythonのファームウェアを書き込むための構成例
STM32マイコン開発ボードNUCLEO-F767ZIのBOOT0をHレベルに設定し，CN1から電源を供給するとSTM32マイコンがファームウェア書き換えモードで起動する

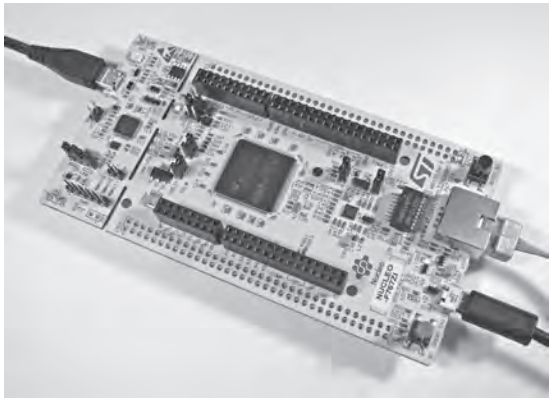


写真8-1　LAN接続用Ethernetインターフェース内蔵のSTM32マイコンを搭載したNUCLEO-F767ZI
LAN接続用Ethernetコネクタを装備しているので，手軽にMicroPythonを使ったIoT機器が製作できる

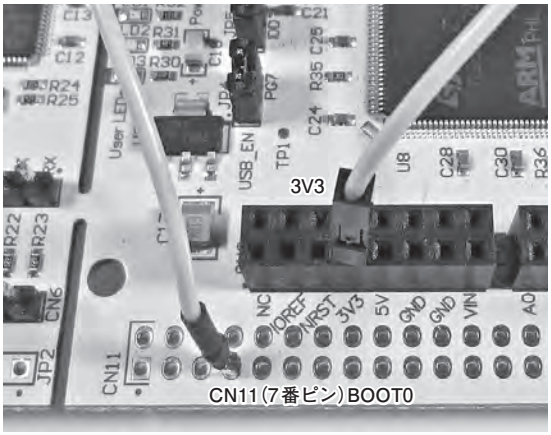


写真8-3　ファームウェア書き換えモードに変更するためのBOOT0部の拡大図
STM32マイコンNUCLEO-F767ZIのCN11の7番ピン(BOOT0)のスルー・ホールから，CN8の7番ピン(3V3)をジャンパ・ワイヤで接続する

第9章 ラズベリー・パイPicoで BLEワイヤレス・センサを作る

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．

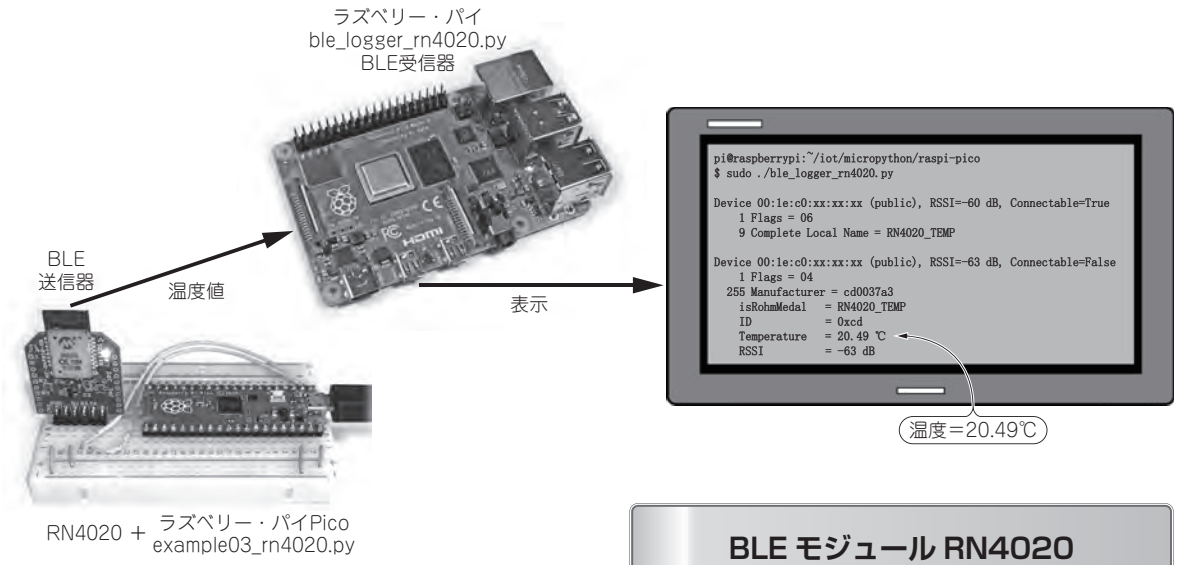


図9-1　本章で製作するBLEワイヤレス・センサの一例
ラズベリー・パイPicoで測定した温度値をBLEで送信し，ラズベリー・パイで受信・表示する

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IoT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．

1 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．

① 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．
② 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．
③ 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，

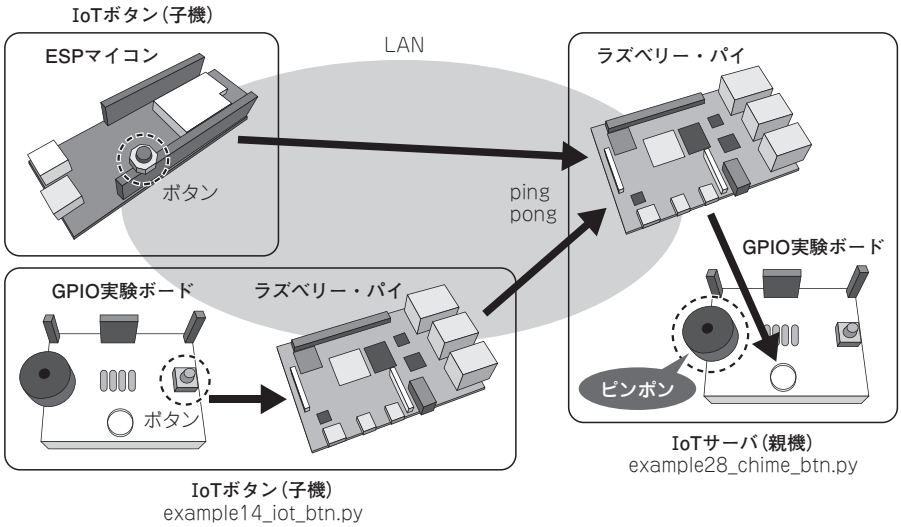


図10-1　IoTボタンでチャイム音を鳴らす呼び鈴のシステム例
第4章で製作したGPIO実験ボード搭載のラズベリー・パイと親機となるチャイム呼び鈴サーバでシステムを構成する

第11章 ラズベリー・パイとPythonでIoT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://aiyprojects.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．

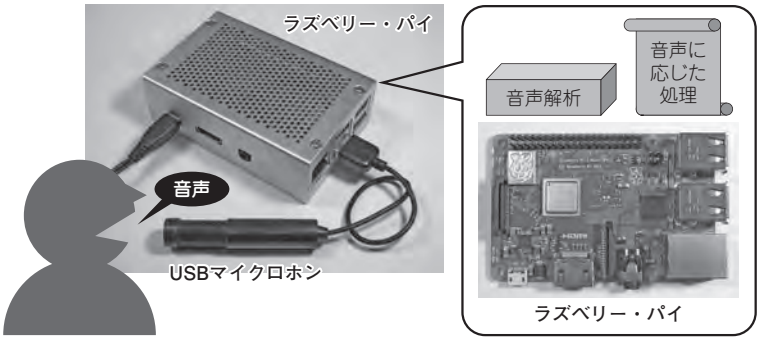


図11-1
ラズベリー・パイ上で動作する音声認識エンジンを使用してIoTユーザ・インターフェースを作成する
USBマイクロホンに入力した音声の認識結果から音声に応じた処理を行う

第12章 Pythonで広がるIoT応用システムの構築 ～IoTシステム応用編～

IoT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．

1 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．

① 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．
② 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．
③ 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．

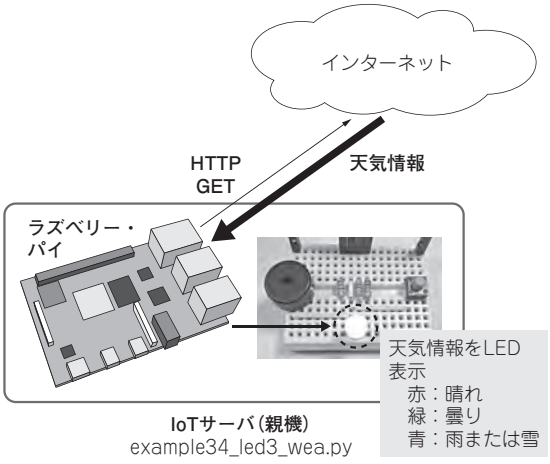


図12-1
インターネット照る照る坊主でLEDの発光色を制御
インターネットから天気情報を取得し，予報が晴れのときは赤色に，曇りのときは緑色に，雨や雪のときは青色にLEDを光らせる