

1人でできちゃう IoT

第2回…呼び鈴から LINE 通知 <前編>開発環境を整える

岩貞 智



写真1 呼び鈴が押されたらLINEに通知が来る装置

「インターホンの呼び鈴が押されたことを検知して 通知を送る」装置を作ります(**写真1**).

● 製作のきっかけ

最近は自宅にいる時間が増え,宅配サービスを頻繁 に利用するようになった方も多いと思います.この宅 配サービスですが,配達時に別の作業をしていると, 呼び鈴に気がつかないことがあります.そこで,呼び 鈴が鳴ったときに手元のスマートフォンへ通知するこ とで解決します.

● 学べること

この記事では、以下を学ぶことができます.

・光センサ・モジュールの仕組みと使い方

VSCodeを使用した MicroPythonの開発環境づくり

•WebREPLを使ったリモート・デバッグ方法

製作費用はセンサと IoT デバイスを含めて 2000 円 程度です.

全体構成

● 想定しているインターホンの仕様

今回想定しているインターホンは画面が付いている タイプで,呼び鈴が押されたら自動的にモニタに人が 表示されるものです.インターホン自体にスマート フォン連係機能があればよいのですが,ない機器を想 定しています.

● 光センサでインターホンの呼び出しを検知

全体構成を図1に示します.まず、インターホンから どのように呼び鈴が鳴ったことを知るかですが、今回 は光センサ・モジュール(以下,光センサ)を使います. インターホンの呼び鈴を鳴らされて、室内に設置して あるモニタが光ったことを検知して通知を行います.

● マイコンでセンサ情報をインターネットに送る

インターホンの画面端に光センサをつけた IoT デバ イスを取り付けます. その光センサと GPIO で接続し たマイコンでセンサ値を読み取り,スマートフォンへ 通知します.マイコンは外部のネットワークへの通知 が必要なため,Wi-FiやLAN などでインターネット に接続する必要があります.



28 Interface 2022年7月号 別冊付録

プログラムは本誌サポート・ページから入手できます. https://interface.cqpub.co.jp/2207py/

1人でできちゃう loT

● 通知にはLINEを使う

インターネットにはWi-Fiを経由して、LINE社が 提供するLINE Notifyを経由してスマートフォン上の LINEに通知します. LINE Notifyを使用することで HTTPのPOSTリクエストを送信するだけで簡単に LINEを使った通知ができます.

準備1…使用する部品をそろえる

ハードウェアの接続を図2に示します.以下,各部 品について説明します.

● 光センサ

使用する光センサは、秋月電子通商で販売されてい る Ambient Light Sensor^{注1}です. この光センサは、 基板に光センサのフォトトランジスタが取り付けられ ており、照度をアナログ電圧で出力するようにモ ジュール化したものです. これには、3つの色のリー ド線が出ており、それぞれのリード線は以下の配置と なります.

- •青色:光センサ出力
- •赤色:電源入力(DC 3.3~5V)
- •黒色:グラウンド

▶通知方法

青色の光センサ出力の値を読み取って通知を行いま す.光センサの出力値は照度に応じて変動し,暗いと 0に近く,明るいと数値が高くなります.事前にイン ターホンが光っていないときとインターホンが光った ときの照度を測っておき,しきい値を設けて,そのし きい値を超えることで通知を行うようにします.

● マイコン・ボード

Wi-Fi通信機能を搭載するマイコン・モジュール ESP32-DevKitC^{注2}を使用します. ESP32-DevKitCは, 低価格ながら各種センサ値を読み取ることができる GPIOを備え, 無線機能としてWi-Fi, Bluetoothが備 わっています.

また、電源の給電とPCからのソフトウェア書き込みができるように、USBがあり、ESP32のほとんどの GPIOにアクセスできるようにI/Oが外に引き出され ています. さらに入手性が高く情報量も多いです. マイコンには以下、2つの仕事があります.

注1:光センサ,秋月電子通商(税込450円). https://akizukidenshi.com/catalog/g/gM-07035/

注2:ESP32-DevKitC, 秋月電子通商(税込1230円). https://akizukidenshi.com/catalog/g/gM-15673/

Google Colaboratoryの使い方は本誌ウェブ・ページで. https://interface.cqpub.co.jp/python00/



図2 ハードウェアの接続

- ・光センサから電圧値(照度)を受け取る
- ・受けた情報を使って外部のネットワークへ(今回 はLINE Notify)へ通知する

● スマートフォン

通知先としてLINEを選択したため、インターネット につながりLINEを利用できるスマートフォンが必要に なります.

準備2…PCにマイコン・ボードを 認識させる

CP2102のドライバをインストール

皆さんのPCとマイコン・モジュールESP32-Dev KitCの間で,USB通信を行うための準備をします. 具体的にはPCに「CP210x USB-UARTブリッジVCP ドライバ」のインストールを行います(図3).以下の ダウンロード・サイトにアクセスし、ドライバをイン ストールしてください.

https://jp.silabs.com/developers/
usb-to-uart-bridge-vcp-drivers

なお、筆者は開発用PCとしてMacを使っています. 従って、ここで紹介するコマンドはMac用ですが、 Windowsでも同じ要領です.

準備3…マイコンにMicroPython のファームウェアを書き込む

ESP32用のMicroPythonのファームウェアが用意 されているため、これをESP32-DevKitCに書き込み



図3 PCに未知のマイコン・モジュールのドライバをインストー ルする

Interface 2022年7月号 別冊付録 29



図4 マイコンに実行プログラムを書き込むためには PC に書き込 みツールをインストールする必要がある

ます. MicroPythonのファームウェアのバイナリは, 以下から入手してください.

https://micropython.org/download/
esp32/

バージョンごとの安定版バイナリと、日々ビルドされている最新版バイナリがありますが、今回は特に最新のものを利用する必要がないため、安定版のv1.17のRelease (安定版バイナリ)を使用します (2022年1月9日時点).

● 書き込みツールをインストールする

この.binファイルをESP32に書き込むには、Espressif 社から提供されいている esptoolを使用する必要があり ます(図4). esptoolは、Pythonツールとして提供され ているため、Pythonのパッケージ管理ツールである pip3を用いて以下のコマンドでインストールを行います. \$ pip3 install esptool

pip3そのものがない場合は、各OSのパッケージ管 理ツールを利用してpip3をインストールしてください.

● ESP32既存のファームウェアを削除する

ESP32-DevKitCとPCをUSBでつないで、まずは既 存のファームウェアを削除します.以下コマンドを ターミナルから打ちます(Macの場合).

リスト2 binファイルの書き込みに成功すると表示されるメッ セージ

esptool.py v3.2
Serial port /dev/cu.SLAB_USBtoUART
Connecting
Chip is ESP32-DOWDQ6 (revision 1)
Features: WiFi, BT, Dual Core, Coding Scheme None
Crystal is 40MHz
1
Changed.
Configuring flash size
Flash will be erased from 0x00001000 to
0x00175fff
Compressed 1527504 bytes to 987584
Wrote 1527504 bytes (987584 compressed) at
0x00001000 in 25.3 seconds (effective
482.7 kbit/s)
Hash of data verified.

30 Interface 2022年7月号 別冊付録

リスト1 ファームウェアの削除が成功すると表示されるメッ セージ

esptool.py v3.2 Serial port /dev/cu.SLAB_USBtoUART Connecting
Chip is ESP32-DOWDQ6 (revision 1)
Stub running
Erasing flash (this may take a while) Chip erase completed successfully in 12.4s Hard resetting via RTS pin

\$ esptool.py --chip esp32 --port /
dev/cu.SLAB USBtoUART erase flash

なお、ポート先は環境に合わせて変更してください. Linuxの場合はポートが/dev/ttyUSB0となります. Windowsの場合は、ポートが「COM3」などとなります.成功すると**リスト1**の内容が出力されます.

● binファイルをESP32に書き込む

削除できたら先ほどダウンロードした.binファイル を以下のコマンドでESP32-DevKitCに書き込みます. \$ esptool.py --chip esp32 --port / dev/cu.SLAB_USBtoUART --baud 460800 write_flash -z 0x1000 esp32-20210902-v1.17.bin

なお、ポート先は環境に合わせて変更してください. Linuxの場合は、ポートが/dev/ttyUSB0となります.成功すると**リスト2**の内容が出力されます.

ここまで正常に書き込むことができれば, ESP32-DevKitCに MicroPython 用の開発環境を用意できたこ とになります.



ESP32のソフトウェア開発言語および実行環境は, 前回に引き続き MicroPython を使います.

● コード・エディタVSCode

MicroPythonの実行環境として、VSCode (Visual Studio Code)とPymakrを使ってみたいと思います. VSCodeも、それぞれのOSに合わせたバイナリが用 意されているため、以下からダウンロードしてインス トールしてください.

https://code.visualstudio.com/
download

Pythonを動かす指示を送るPymakr

Pymakrは、VSCodeの拡張機能として、ESP32な どのMicroPython実行環境があるデバイスに、REPL

1人でできちゃう loT

によるコンソールを追加します.また、単一ファイル の実行やMicroPythonのファイルをボード間で同期 することができます. なお, REPLはRead Eval Print Loopの略で、入力/評価/出力ループのことで す. Pythonはもともとコードを1行1行実行できるイ ンタープリタ言語であり、REPL機能を用意すること で対話的にコードを実行できます.

VSCodeの拡張機能からPymakrを検索してインス トールします. インストールが完了すると設定ファイ ルが自動的に開くため、この設定ファイルをリスト3 のように書き換えます. addressは、接続先ポートを 指定します. WindowsならCOM3, Linuxなら/dev/ ttyUSB0などになります.

open_on_startとauto_connectはデフォルトのまま trueにしていると、VSCode起動のたびに動くので、 falseへ設定するのがオススメです.

PCとマイコン間でシリアル通信を行うための node.js

node.jsはPCとマイコン・モジュールとのシリアル 通信を行うために利用されています. インストールし ておかないと、プログラムの書き込みやREPL機能を 利用できません.

PCのOSに合ったnode.jsを,以下のウェブ・サイ トからダウンロードしてインストールします. https://nodejs.org/ja/download/

2022年4月7日時点では、最新の安定版16.14.2 LTS をダウンロードしてインストールしました.

リスト3 Pymakrをインストールして設定ファイルを書き換える

"address": "/dev/cu.SLAB_USBtoUART" "open_on_start": false, "auto connect": false, :

動作確認

ESP32-DevKitCと開発用PCをUSBで接続します. その後, VSCode左下にある Pymakr Consoleをク リックすることで、隣にあるアイコンがX→?にな り, ESP32-DevKitCへconnectできます. デフォルト 起動では、端末上にあるmain.pyが実行されます.

ここで, VSCodeのTERMINAL上で「Ctrl + C」を 押すとmain.pyの実行を中断します. するとREPL が起動するため「print ("Hello World")」を入力しま す. その結果「Hello World」が表示されればよいです. * *

*

次回はESP32-DevKitCを使ってセンサからの値を 取得し、LINEアプリを利用してスマホへ通知する方 法を紹介します.

いわさだ・さとし

