

# ESP32 開発環境構築手順書 (Arduino IDE 利用)

2018/2/22

鈴木 森羅

## 1 はじめに

本資料では，Arduino IDE を利用した ESP32[1] の開発環境を構築する手順を示す．ESP32 とは，Wi-Fi と Bluetooth を内蔵する低コスト，低消費電力な SoC なマイクロコントローラである．また，Arduino IDE とは，Arduino と呼ばれるワンボードマイコンの統合開発環境であり，ソフトウェア開発に不慣れな人でも容易にプログラミングできるように設計されている．そして，ESP32 では，公式の提供する Arduino core for the ESP32[2](以降，Arduino core) というツールチェーンを用いることで Arduino IDE を利用して開発を行える．

ここで，本資料では，Mac 環境での構築手順を示す．OS のバージョンは，10.11.6 である．構築手順は Arduino core の GitHub リポジトリにあるドキュメントを基に記述する．また，サンプルプログラムは，秋月電子通商で購入した ESP32-DevKitC[3] 上で動作させた．

## 2 環境構築

手順の中で端末に入力するコマンドには，行頭に\$を付与する．

- (1) 公式サイトから最新バージョンの Arduino IDE をダウンロードし，インストールする．ダウンロードページの URL を以下に示す．

```
https://www.arduino.cc/en/main/software
```

- (2) 端末を開き，以下のコマンドで Arduino core を配置するディレクトリを作成する．また，以降の手順は端末から行う．

```
$ mkdir -p ~/Documents/Arduino/hardware/espressif
```

- (3) Arduino core をダウンロードする．以下に示すコマンドを順に実行し，  
~/Documents/Arduino/hardware/espressif に移動し，GitHub から Arduino core をダウンロードする．

```
$ cd ~/Documents/Arduino/hardware/espressif
```

```
$ git clone https://github.com/espressif/arduino-esp32.git esp32
```

コマンドを実行すると，~/Documents/Arduino/hardware/espressif/esp32 以下に Arduino core がダウンロードされる．

Windows の場合，Git GUI[4] を使用しダウンロードする．または，Arduino core の GitHub リポジトリのページから zip 形式でダウンロードする．

- (4) 以下のコマンドを実行し，Arduino core で利用するライブラリを最新の状態に更新する．

```
$ cd esp32
$ git submodule update --init --recursive
```

- (5) スクリプトを実行し，Arduino IDE にツールチェーンをインストールする．

```
$ cd tools
$ python get.py
```

このとき，Mac の場合，以下のようなエラーが表示される可能性がある．

```
xcrun: error: invalid active developer path (/Library/Developer/CommandLineTools)
, missing xcrun at: /Library/Developer/CommandLineTools/usr/bin/xcrun
```

この場合，以下のコマンドで xcode-select をインストールする．

```
$ xcode-select --install
```

Windows の場合，tools フォルダ内の get.exe を実行する．

- (6) Arduino IDE を再起動する．

### 3 動作確認

Arduino IDE 既存のサンプルプログラムを利用し，動作確認を行う．ここでは，LED を点滅させるサンプルプログラムを動作させる．

- (1) 配線する．

図 1 のようにブレッドボード上に配線する．このとき，適切な抵抗値を計算し，抵抗を選ぶ．

- (2) サンプルプログラムのソースコードを開き，編集する．

Arduino IDE を起動し，ウィンドウ上部の「開く」ボタンをクリックする．表示されたメニューから「01.Basics」の中の「Blink」を選択し，ソースコードを開く．ソースコード中の LED\_BUILTIN を 4 に置き換え，図 2 のように編集する．

- (3) ESP32 と計算機を接続する．

USB ケーブルで ESP32 と計算機を接続する．このとき，ESP32 開発ボードに搭載されている USB シリアル通信用 IC に応じたドライバをインストールする．本資料と同様に，秋月電子通商から開発ボードを購入した場合，CP2102 が搭載されている．以下の URL のページからドライバをダウンロードし，インストールする．ただし，以下のページは，CP2102[5] で検索すれば，1 件目に出る．

[https://jp.silabs.com/products/development-tools/softwareusb-to-uart-bridge-vcp-dr](https://jp.silabs.com/products/development-tools/softwareusb-to-uart-bridge-vcp-driver)

- (4) ソースコードをコンパイルし，ESP32 へ書き込む．

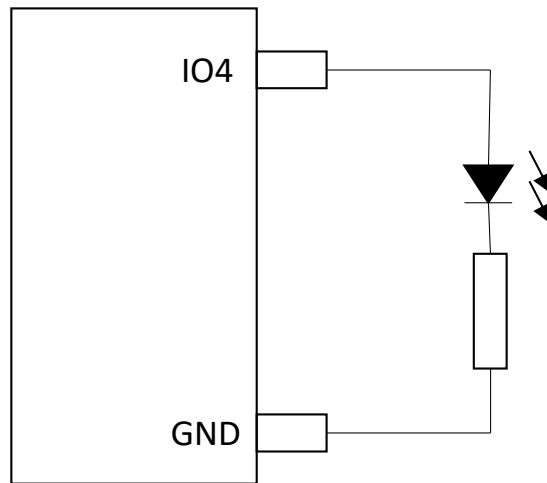


図 1 サンプルプログラム動作の回路図

```

1 void setup() {
2   pinMode(4, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(4, HIGH);
7   delay(250);
8   digitalWrite(4, LOW);
9   delay(250);
10 }

```

図 2 LED を点滅させるサンプルプログラム

メニューバーのツールから以下の設定を行う。

ボード： ESP32 Dev Module

シリアルポート： /dev/cu.SLAB\_USBtoUART

書込装置： AVRISP mkII

シリアルポートの設定は，USB シリアル通信用 IC により異なる。

設定後，Arduino IDE のウィンドウ上部の「マイコンボードに書き込む」ボタンをクリックし，書き込む。書き込みが終了すると，LED が点滅を始める。

## 4 おわりに

本資料では，Arduino IDE を利用した ESP32 の開発環境構築の手順，およびサンプルプログラムによる動作確認の手順を示した。

## 参考文献

- [1] ESP32 Overview, Espressif Systems (online), <https://www.espressif.com/en/products/hardware/esp32/overview> (accessed 2018-2-14).
- [2] espressif: Arduino core, GitHub, Inc. (online), <https://github.com/espressif/arduino-esp32> (accessed 2018-2-14).
- [3] ESP32-DevKitC ESP-WROOM-32 開発ボード, 秋月電子通商(オンライン), 入手先 (<http://akizukidenshi.com/catalog/g/gM-11819/>) (参照 2018-2-14).
- [4] Git GUI, GitHub, Inc. (online), <https://git-scm.com/> (accessed 2018-2-14).
- [5] CP2102, Silicon Labs (online), <https://jp.silabs.com/products/interface/usb-bridges/classic-usb-bridges/device.cp2102> (accessed 2018-2-14).