

ESP32の主な開発環境

ご購入はこちら

井田 健太

表1 ESP-WROOM-32の開発環境は大きく3つ…ベテラン/ビギナ/Pythonユーザー向け

名称	特徴	執筆時点のバージョン	開発環境の動作環境	開発言語	使用可能な無線機能	URL
ESP-IDF	Espressif Systems社の公式開発環境	v3.0	Windows, MacOS, Linux	C, C++	Wi-Fi, Bluetooth (Classic, Low Energy)	https://github.com/espressif/esp-idf
Arduino core for ESP32	Arduino IDEでESP32の開発を行うためのボード・パッケージ	6826508 (master ブランチの執筆時点でのHEAD)	Windows, MacOS, Linux	Arduino (C++)	Wi-Fi, Bluetooth (Classic, Low Energy)	https://github.com/espressif/arduino-esp32
MicroPython	組み込み向けに最適化されたPython実装	v1.9.3-521	Windows, MacOS, Linux	MicroPython (Python 3.4のサブセット)	Wi-Fi	https://micropython.org/

ESP32の開発環境として現時点で利用可能なものがいくつかあります。その中で筆者がおすすめする3つの開発環境を表1に示します。

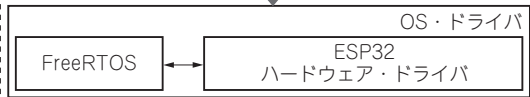
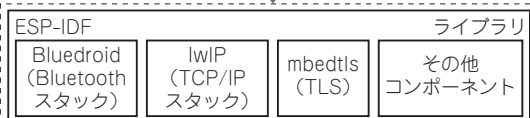
開発環境その①…純正ESP-IDF

● ESP-IDFとは

ESP-IDF (ESP-IoT Development Framework) は Espressif Systems 社によって開発・提供されている ESP32 向けの公式開発環境です。開発言語は C 言語 (C99+GNU 拡張), C++ 言語 (C++11 + GNU 拡張) を

アプリケーション

ユーザ・コード



書き込み↓

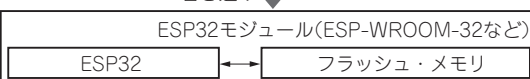


図1 ESP-IDFのソフトウェア構成

Wi-FiやBluetoothなどIoTに使える通信用スタックがすでに実装されている

使用します。他の開発環境は基本的にこのESP-IDFを用いて作られています。基本となるだけあって、ESP32の機能をほぼ全て使うことができます。デュアルコアであることを有効活用し高い処理能力を生かしたり、低消費電力化を極限まで行ったりしたい場合にはESP-IDFが候補となるでしょう。

● ソフトウェアの構成

ESP32はCPUコアが2つあるデュアルコア構成となっているので、これらのコアを有効活用するためにはタスクの管理や同期機構といったリアルタイムOSの機能が欠かせません。

ESP-IDFでは、OSとしてオープンソースのリアルタイムOSであるFreeRTOSを使用しています。

ESP-IDFは、幾つかのコンポーネントから構成されています(図1)。

FreeRTOS, ESP32のハードウェア・ドライバ, Wi-Fi, Bluetoothといった機能がコンポーネントとして実装されています。必須のコンポーネントを除けば、コンポーネントの有効・無効を切り替えることができます。

これにより、不必要な機能は除外して、プログラムのサイズを抑えるといったことが可能になっています。

● ESP-IDFのライセンス

コンポーネントとしては、ESP32のハードウェア・ドライバのようにESP32固有のもの以外に、Wi-FiやBluetoothの通信、フラッシュ・メモリ上のファイル・