

# オープンソースの本格デバッグ環境を作る

石原 和典

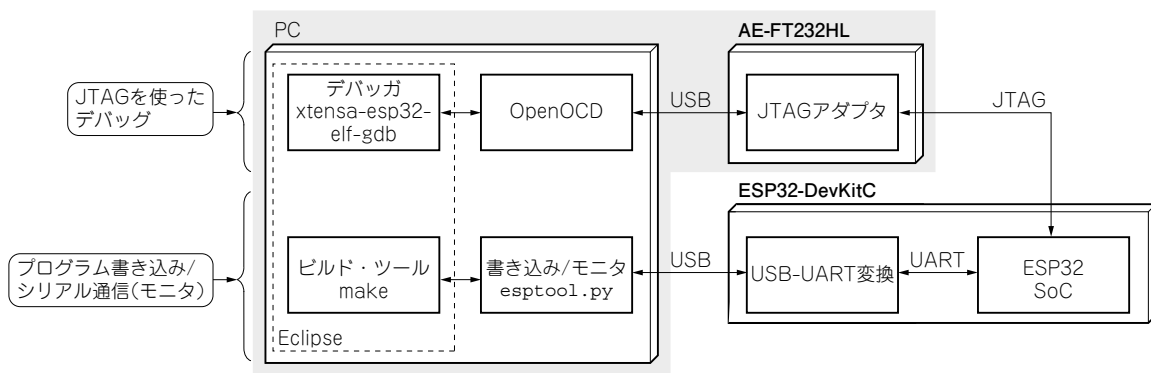


図1 複数のオープンソース・ツールを組み合わせることでデバッグ環境を構築する

ESP32でプログラミングする際、どのようなデバッグ手法を思い浮かべるでしょうか。プログラムの中にprintfなどを埋め込み、見たい情報をシリアルなどに出力して確認する、いわゆる「printfデバッグ」が多く使われているのではないでしょうか。

この方法はもちろん有用ですが、扱うプログラムによっては、ブレーク・ポイントを仕掛けて止め、ステップ実行したり、変数の中身を確認したりといった、もっと効率の良いデバッグを行いたい場合もあります。本稿では、ESP32でも高機能なデバッグができるよう、統合開発環境であるEclipseにOpenOCDとJTAGを組み合わせた、プログラム開発環境を構築してみます。

JTAGとはマイコンのデバッグで使われる標準的なインターフェースです。ESP32に限らずさまざまなマイコンの開発で使われています。そのJTAGを開発環境やデバッグ・ツールから使いやすいようにサービスを提供するのがOpenOCDです。JTAGとも開発環境とも独立しているため、開発するマイコン（CPU）を切り替えたり、開発環境を切り替えたりすることもできます。

## ● システム全体の構成

今回のシステムは統合開発環境のEclipse以外に、

Linux上で使われるデバッガgdb、OpenOCD、JTAGといった要素で構成します（図1）。

システムの前提条件は次の通りです。

- PCのOS  
Windows 10 64ビット版
- 開発環境  
ESP-IDF 3.3
- マイコン・ボード  
ESP32-DevKitC（秋月電子通商など）
- JTAGアダプタ  
AE-FT232HL（秋月電子通商など）

公式のESP-IDFを使った開発環境をお使いでも、OpenOCDと、JTAG adapterを未導入の方が多くと思います。OpenOCDはビルド済みバイナリが提供されているので、PCのディスクに展開するだけで使えます。JTAGアダプタはAE-FT232HL（秋月電子通商）が利用可能なため、1,200円程度のハードウェアだけで実機のマイコンでプログラムを動かしながらデバッグする（オンチップ・デバッグ）環境構築が可能です。

gdbやOpenOCDはLinuxなどUnix系のOS用のデバッグ・ツールですが、今回はWindows 10で使いますので、Windows上で動くLinuxのシェル環境が必要になります。

現在、ESP-IDFのWindowsへのインストール方法