

汎用デバッガ×VS Codeで構築するデバッグ環境

丸石 康

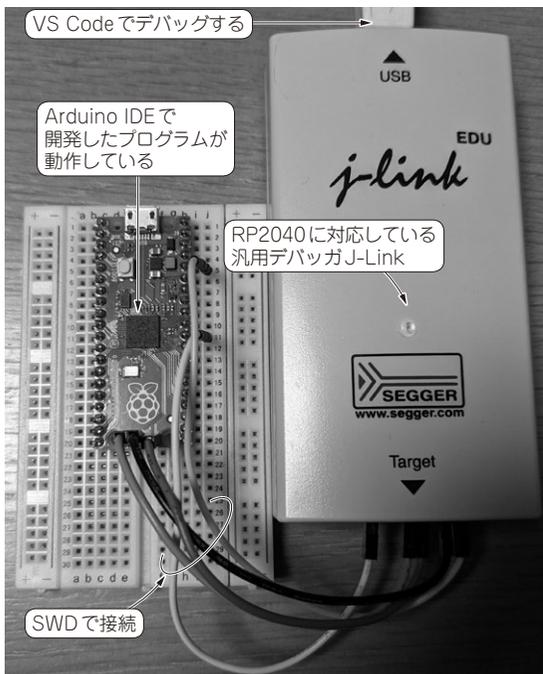


写真1 Arduino IDEで開発したPico用プログラムを汎用デバッガとVS Codeでデバッグしてみる
Picoと汎用デバッガJ-Link EDUを接続した様子

● おなじみ Arduino IDEでも Picoの開発ができる

Arduinoは、ラズベリー・パイ Pico (以降、Pico) に搭載されている RP2040 を公式にサポートすることを発表しました。Arduino が提供している統合開発環境 Arduino IDE は、ラズベリーパイ財団から提供されている公式 SDK (Software Development Kit) に比べると、環境構築の手順がシンプルなことや、豊富な Arduino 用ライブラリの資産が利用できることなどのメリットがあります。

● 汎用デバッガと VS Codeがあればデバッグも OK!

Arduino IDE は、スケッチ (C++ ベースの Arduino 専用プログラミング言語) のコンパイルと、シリアル・モニタを使ったデバッグができますが、ステップ実行や任意の変数の値確認など、いわゆるデバッガ機能はありません。2021年3月にベータ版がリリースされた Arduino IDE 2.0 には将来的にソースコード・デバッグ機能が追加されるようですが、本原稿執筆時点 (2021年5月) では対応が十分ではありません。

そこで本稿では、現時点での正式リリース版である Arduino IDE 1.8.15 でコンパイルした実行ファイルをデバッグする方法を紹介します。本稿で紹介するデバッグ方法では、写真1に示す汎用デバッガ J-Link (SEGGER) と VS Code を使います。

ステップ1: Arduino IDE に Pico の開発環境を追加する

● Picoの開発に Arduino IDE を使うメリット

▶ (1) 豊富な Arduino ライブラリが利用できる

Pico 用の C++ 開発環境に Arduino IDE を使えば、豊富な Arduino 用ライブラリの資産を Pico 用プログラムの開発に利用できます。

ライブラリ資産の中には、Pico の性能を生かせないものや Pico で使用できないものもありますが、コミュニティ・ベースの開発が進むにつれて問題は解消されていくと考えられます。

▶ (2) 少ない学習コストで利用できる

Arduino IDE と公式 SDK は、どちらもコンパイラに GCC (GNU Compiler Collection) を使っていますが、サポートしている API (Application Programming Interface) が異なります。Arduino IDE は、Arduino API をサポートしています。公式 SDK には独自の API が用意されています。

Arduino API は、通常だとマイコンによって異なるドライバなどのインターフェースの差分を吸収してくれるので、新規のマイコンやボードであっても少ない学習コストで開発できるようになります。