

# 2021年 お勧め開発環境 VS Code

第7回

Cortex-Debugを導入しノルディックの  
nRFマイコンをデバッグする

桐畑 鷹輔

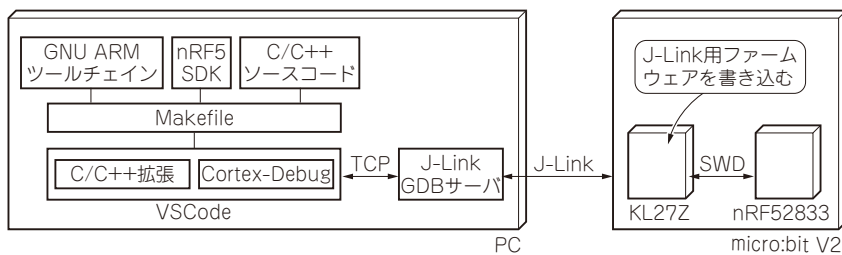


図1  
今回作るもの

Visual Studio Code (以下、VS Code) は、エディタとして利用する以外に、拡張機能の導入と設定ファイルを用意することによって、統合開発環境のように利用できます。今回、Arm Cortex-M4Fコアを採用したマイコンnRF52833(ノルディック・セミコンダクター)をターゲットに、VS Codeの拡張機能Cortex-Debugを使った開発方法を解説します。

## nRF52833マイコンが搭載された 入手しやすいmicro:bitでトライする

● 英国のBBCが作った初學者向けにだけ機能豊富なマイコン・ボードとしては、入手性の良いmicro:bit V2(BBC)を使います。このマイコンの特徴としてBLE機能があります。

今回はUARTで接続したCO<sub>2</sub>センサの情報をBLEでスマートフォンへ送ってみます(図1)。

micro:bitは、英国のBBCが作ったマイコン・ボードです。ブロック・プログラミング環境MakeCodeや、MicroPython、Arduino IDEでも手軽にプログラミングできます。2020年に発売されたmicro:bit V2には、BLE機能を内蔵したnRF52833マイコンが搭載されています。

### ● ボード上にデバッガも搭載

micro:bit V2の基板には、メインのマイコンnRF52833とは別に、KL27Z(NXPセミコンダクター)というマイコンが実装されています。KL27ZのCPUコアはArm Cortex-M0です。

KL27Zには、nRF52833のプログラム書き換え用に

DAP Linkファームウェアが書き込まれています。

micro:bit用に用意された開発環境を使う場合は、DAP Linkでよいのですが、今回はVS Codeで開発するために、KL27ZにJ-Link(SEGGER)のファームウェアを書き込み、J-Linkのデバッガとして利用します。

J-Linkは、SEGGERの開発したデバッガで、Armマイコンの開発に利用できます。他のデバッガと比べ、フラッシュ・メモリへの書き込みやブレイク実行が高速という特徴があり、実際に筆者もそのように感じることがあります。

ノルディック・セミコンダクター製のnRFマイコン評価ボードでは、J-Linkデバッガ・チップがオンボード搭載されており、J-LinkはnRFシリーズの開発ではよく使われます。デバッガ接続だけでログ出力を便利に行えるRTTという機能を持っていたり、さまざまなソフトウェアが提供されていたりするのが特徴です。

nRFシリーズ搭載で日本でも買えるようなマイコン・ボードは、デバッガが別途必要なものが多いですが、micro:bit V2はその必要がなく、安価で手軽に扱えるのでnRFマイコンの開発を試してみるのにお勧めです。

## 開発環境の構築

### ■ デバッガのファームウェアを書き換える

#### ● J-Link化：手順1…ダウンロード

以下のウェブ・ページからJ-Linkファームウェアを