第1章 実用的なデバイス開発をRustで

CMSIS-DAPの実装で実践するUSBデバイス開発

井田 健太

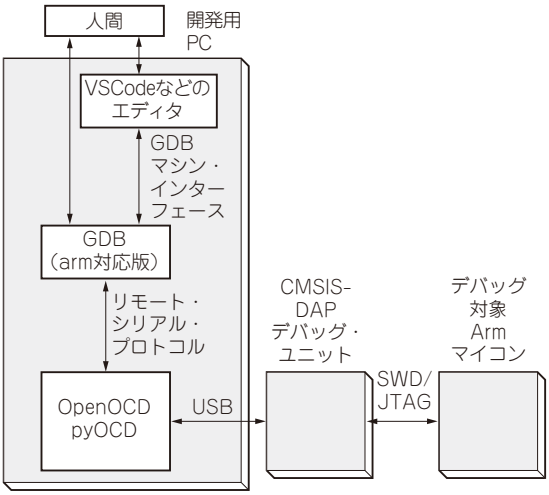

図1 デバッグ・ユニットを用いたデバッグ時の構成

Arm Cortexマイコンのデバッグのための仕組みはCMSIS-DAPとして仕様が決まっており，ラズベリー・パイPicoなどのArm Cortex-MのCPUコアを搭載したマイコン・ボードのデバッグなどに用いることができます．

このCMSIS-DAPの実装はArm公式のC言語で実装されたものがよく用いられますが，今回はRustにて実装しましたので，実装内容について紹介します．

また，その中でUSBによる通信制御を行いますので，RustでのUSB通信制御の方法についても解説します．

マイコンのデバッグ方法について

● マイコンのデバッグ環境

マイコンのデバッグにはJTAGやSWDといったプロトコルを用いて，マイコンに内蔵されたデバッグ回路とデバッガの間で通信を行う必要があります．

このとき，デバッガが動作しているPCでは直接JTAGやSWDといったプロトコルの信号を扱うのが難しいため，USB接続のデバッグ・ユニットを用います．

例として，**図1**に，デバッグ・ユニットを用いたArmマイコンのデバッグ環境の構成を示します．

● オープンソースのデバッガGDB

GDBはGNU Projectにより開発されているオープンソースのデバッガです．PC用のソフトウェアから組み込み向けのソフトウェアまで幅広いターゲットのソフトウェアをデバッグ可能です．

組み込み向けのターゲットのソフトウェアをデバッグする場合は，後述するOpenOCD，pyOCDやcargo-embedと言ったGDB Remote Serial Protocolを実装したソフトウェアと組み合わせて，ターゲットのデバッグ機能を制御します．

GDBはターゲットのアーキテクチャごとに専用のものの他に，複数ターゲット・アーキテクチャに対応したものもあります．

UbuntuなどのLinux環境ではgdb-multiarchという名称でArm CortexやRISC-Vなどの複数のアークテクチャに対応したGDBが用意されています．こういった環境では，**図1**中のArm対応GDBとしてgdb-multiarchを利用できます．

● CPUの内蔵デバッグ機能を使うOpenOCD/pyOCD

OpenOCDおよびpyOCDは，各種CPUの内蔵デバッグ機能を主にチップ外部から制御するデバッガです．OCDはOn Chip Debuggerの略です．

OpenOCDはC言語で記述されていますが，pyOCDはPythonで記述されています．

OpenOCD/pyOCDは単体でもブレークポイントの設定や実行状態の取得といったデバッガとしての基本的な機能を持っています．ソース・レベル・デバッグといった高度な機能については，前述のGDBと組み合わせてGDB Remote Serial Protocol経由で制御される機能（GDB Server機能）により実現しています．

OpenOCD/pyOCDが対象とするターゲット内蔵の