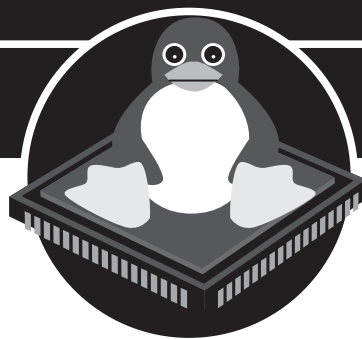


さまざまなシングル・ボード・コンピュータ向けに
自分専用カスタム!

ご購入はこちら



Yocto Projectではじめる 組み込みLinux開発入門

第21回

Yocto Project 5.0編④…実機でリモート・デバッグ

三ツ木 祐介

第18回からは、2024年4月にリリースされた最新のLTS (Long Term Support) であるYocto Project 5.0 (コードネーム: Scarthgap) を紹介しています。ターゲット・ボードにはラズベリー・パイ4モデルB (以降、ラズベリー・パイ4、写真1) です。

今回は、Yocto Project 5.0で新たに追加された、リモート・デバッグにおけるVisual Studio Code (VS Code) との連携機能について紹介します。(編集部)

デバッグの基礎知識

● バグを取り除く作業のこと

デバッグは、プログラムの動作上の問題(バグ)を取り除く作業のことです。通常デバッガと呼ばれるプログラムを使用して、あらかじめデバッグ情報が埋め込まれた監視対象のプログラム(デバッギ)を動作させながら行います。

デバッガは、デバッグ情報を参照して、実行中のプログラムの位置に対応するソースコードを表示したり、メモリや変数の状態を参照できるようになっています。

● 大きく分けて2通りの実施方法がある

組み込み機器で動作するソフトウェアのデバッグには、大きく分けて2種類あります。1つ目は、デバッガを直接ターゲット(デバッグ対象のマイコン・ボードなど)で動作させるローカル・デバッグです。2つ目は、ホストPC上でデバッガを動作させ、ターゲット上のスタブと呼ばれる軽量のプログラムと通信して行うリモート・デバッグです。

▶その①…ローカル・デバッグ

ローカル・デバッグでは、デバッグに関する全ての処理をターゲットで行うため、デバッギのソースコードなどが全てターゲット上に配置されている必要があります。デバッガ自身も比較的重い処理を行うため、CPUやストレージなどのリソースを多く使用することになります。

逆に言えば、デバッガを動作させる条件を満たしていれば、LANやシリアル・ポートなどでホストPCに接続する必要がなく、ターゲットだけで完結すること

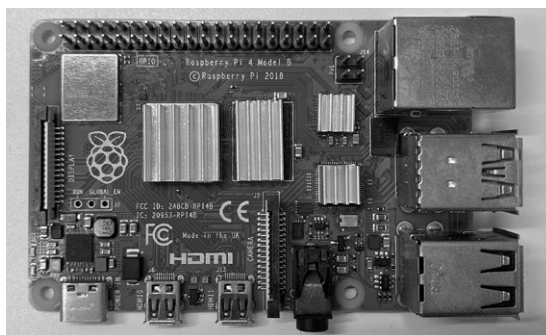


写真1 最新LTSであるYocto Project 5.0 (Scarthgap) でラズベリー・パイ4向けLinuxを作ってみる

今回はラズベリー・パイを動作させてリモート・デバッグを試してみる

ができます。

▶その②…リモート・デバッグ

リモート・デバッグでは、デバッガをホストPCで実行します。そのため、デバッギのソースコードなどはホスト上に存在すればよいので、ターゲット上に配置する必要はありません。デバッガの処理の大半をホストPCで行うので、ターゲット上のCPUやストレージはそこまで使用しませんが、ホストPCとターゲットをLANやシリアル・インターフェースなどで接続する必要があります。

*

ラズベリー・パイ4は、比較的ターゲットのリソースが豊富ですが、有線/無線LANに接続できるので、どちらの方法でもデバッグ可能です。

Yocto Project 5.0から追加! VS Codeとの連携機能

● 組み込みLinux環境向けIDEが手軽に使える

Yocto Project 5.0には、リモート・デバッグを行うときにVS Codeと連携する機能が提供されています。そのため、統合開発環境(IDE)のように、ホストPCでソースコードを修正しながらターゲット上でデバッグを行うことができるようになっています。

組み込みLinux環境向けの統合開発環境の構築は、

第1回 組み込み開発とLinuxの関係(2022年9月号)

第2回 ラズパイ編①…最小構成のLinuxイメージを作る(2022年10月号)

第3回 ラズパイ編②…Dockerを使ってWindowsにビルド環境を構築する(2022年11月号)