

# Yocto Projectではじめる 組み込みLinux開発入門

第20回

Yocto Project 5.0編③…ラズパイのブート・シーケンスのレシピ実装 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。(編集部)

前回(2025年8月号)、ラズベリー・パイ・シリーズのブート・シーケンスを調べました。ブート・パーティションに存在するファイルの役割についても触れ、meta-raspberrypiでは次のレシピでブート・パーティションにファイルを配置していることがわかりました。

- recipes-bsp/bootfiles/rpi-bootfiles.bb
- recipes-bsp/bootfiles/rpi-cmdline.bb
- recipes-bsp/bootfiles/rpi-config\_git.bb

今回はこれらのレシピ・ファイルがどのような実装になっているかを見ていきます。

## レシピ①…ブート・パーティションに必要な ファイルを配置(rpi-bootfiles.bb)

リスト1にrecipes-bsp/bootfiles/rpi-bootfiles.bbの抜粋を示します。このレシピのポイントは、次に示す3つです。

- (1) deployクラス
- (2) nopackagesクラス
- (3) do\_deployタスク

6行目のinherit deploy nopackagesによって、deployクラスとnopackageクラスを読み込んでいます。29~49行目のdo\_deployタスクの定義がこのレシピの本体と言える処理です。

### ● (1) deployクラス…ファイルのコピー処理

deployクラスは、OE-Core(meta)のclasses-recipe/deploy.bbclassで定義されています。deploy.bbclassの内容をリスト2に示します。

deployクラスで行っている処理は、次の通りです。

1. コピー元のディレクトリをDEPLOYDIRとして設定

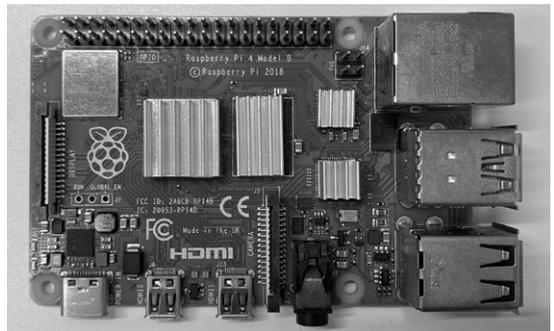


写真1 最新LTSであるYocto Project 5.0(Scarthgap)でラズベリー・パイ4向けLinuxを作ってみる

今回はラズベリー・パイのブート・シーケンスのレシピ実装を見てみる

2. inherit(継承)したレシピのdo\_deployタスクに必要なファイルをDEPLOYDIRにコピー
3. 継承したレシピの全てのDEPLOYDIRからDEPLOY\_DIR\_IMAGEにコピー

このクラスでは、do\_deployタスク自体は定義していません。これは、inherit(継承)したレシピがそれぞれdo\_deployタスクを定義することを期待しているためです。

### ▶ ファイルをコピーする処理の内容

このクラスの最も重要な部分は、9~10行目の定義です。この記述は、ビルド・ツールBitBakeの「変数フラグ」と呼ばれる機能です。BitBake内部では、変数フラグをキーに、複数の変数やタスクが横断的に処理されます。ここではdeploy.bbclassを継承した全てのレシピのdo\_deployタスクについて、sstate-inputdirsとsstate-outputdirsが設定されるようになっています。

sstate-inputdirsで指定されたDEPLOYDIRは7行目で定義されており、各レシピのWORKDIR配下のdeploy-\${PN}というディレクトリが設定されています。一方、sstate-outputdirsで指定されたDEPLOY\_DIR\_IMAGEは、OE-Coreのconf/bitbake.confで定義された変数となり、基本的には全てのレシピで同じ場所を指すようになっています。