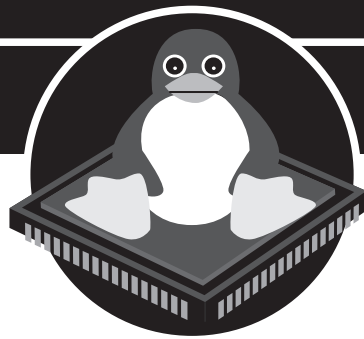


さまざまなシングル・ボード・コンピュータ向けに  
自分専用カスタム!

# Yocto Projectではじめる 組み込みLinux開発入門



第12回

ROCK 4C+ 編③…ターゲット向けにカーネルをビルドし直す

三ツ木 祐介

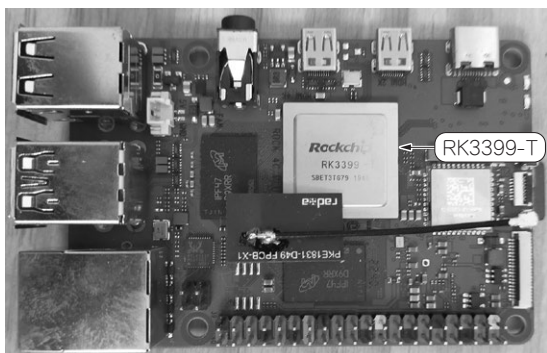


写真1 ターゲット・ボード…RK3399-Tを搭載するROCK4 C+ (Radxa)

ラズベリー・パイ4 との互換性を重視していて、十分代替品になり得る性能を持っている

第10回(2023年10月号)からSoCとしてRK3399-T (Rockchip)を搭載する写真1のROCK4 C+ (Radxa)を新たなターゲット・ボードとして、Yocto ProjectでLinuxを作成しています。

前は、1世代前の製品であるROCK 4C (Radxa)のブート・シーケンスを調査し、ブートローダがどのようにLinuxカーネルを起動するかを確認しました。その結果、単にmicroSDカードに書き込まれたLinuxを起動するだけであれば、ROCK4 C+向けにブートローダを改造する必要はなさそうであることが分かりました。

一方で、第10回で作成したLinuxイメージでは、ディスプレイやキーボードは全然反応しませんでした。今回は、ROCK4 C+向けのカーネルを作成して、ターゲットに接続したディスプレイやキーボードを使用できるようにします。

## Linuxカーネルに用意されている 環境の差異を吸収する仕組み

### ● その①…カーネル・コンフィグ

Linuxカーネルは、さまざまなCPUのアーキテクチャで動作し、さまざまなデバイスを制御できます。カーネルがどのアーキテクチャで動作するのか、どのようなデバイスを制御するのかは、使用する環境に

よって異なります。そのような差異を吸収するために、どの機能を有効化/無効化するかを設定する仕組みとして、カーネル・コンフィグレーション(カーネル・コンフィグ)があります。

Linuxカーネルをビルドするとき、カーネル・コンフィグを設定し、今回作成するカーネルがどのようなアーキテクチャのどのようなハードウェアで動作するのかを決めます。その中でもアーキテクチャやハードウェアのデフォルト値となるコンフィグを定義したものを「defconfig」と呼びます。SoCメーカーやボード・メーカーなどのBSP(Board Support Package)提供者は、自身が提供するハードウェアで動作するためのカーネルのdefconfigも提供します。

### ● その②…デバイス・ツリー

カーネル・コンフィグではさまざまな機能の有効化/無効化を設定できますが、ターゲット・ボードが搭載しているペリフェラルとSoCが、どのように接続されているかまでは定義できません。組み込みの環境では、同じSoCを搭載していても周辺のペリフェラルなどのハードウェアがどのように接続されているかはターゲットによって異なります。Armなどの組み込み向けアーキテクチャではその差異を吸収するためにデバイス・ツリーという仕組みを採用しています。

dtcというリスト1のようなテキスト・ファイルでdtcというコマンドでバイナリ形式であるdtbというファイルに変換します。U-Bootなどのブートローダでカーネルとdtbを組み合わせで使います。このような構成では、ロードするdtbを差し替えることでカーネルを再ビルドせず、新たなハードウェア構成に対応させることが可能となっています。

一方でカーネルとdtbをあらかじめ結合して使用することもできるようになっています。

リスト1は、[linux.org](http://linux.org)のDevice Tree Usage<sup>(1)</sup>にある例を一部抜粋したものです。