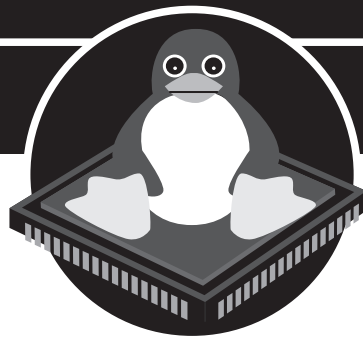


さまざまなシングル・ボード・コンピュータ向けに
自分専用カスタム!

Yocto Projectではじめる 組み込みLinux開発入門



第11回

ROCK 4C+ 編②…起動の仕組みを調べる

三ツ木 祐介

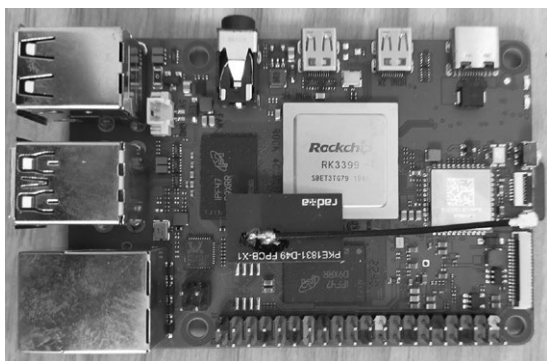


写真1 ターゲット・ボード…RK3399-Tを搭載するROCK4 C+ (Radxa)
ラズベリー・パイ4との互換性を重視していて、十分代替品になり得る性能を持っている

前回(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。

前回は、RK3399 (Rockchip)を搭載するROCK 4C (Radxa)用のマシン定義(meta-rockchipのrock-pi-4c)を使ってOSイメージを作成し、起動を試みました。一部のペリフェラルは動作しなかったものの、シリアル・コンソールで接続するとLinuxが起動していることが確認できました。同じシリーズのボードとはいえ、別のハードウェア設定を使って起動することができました。

今回は、別のハードウェア設定でどのように起動したのか、また完全とはいかないものの一般的に十分なペリフェラルを動作させるには、どのようにすればよいのか考えます。

① RK3399のブート・シーケンスを調べる

ボードに電源が入ってからLinuxが起動するまでの流れは、SoCにより異なります。ここでは、ROCK 4Cシリーズに搭載されているRK3399のブート・シーケンスを調べます。

● 一般的なLinuxブート・シーケンス

OSの一般的なブート・シーケンスを次に示します。

- (1) IPL (Initial Program Loader) の実行
- (2) SPL (Secondary Program Loader) の実行
- (3) フル機能のブートローダの実行
- (4) カーネルの実行

OSのカーネルを実行するには、このように段階的にブートローダ(Program Loader)が実行されます。これはSoC起動時にメモリやストレージを含めたほとんどのペリフェラルが初期化されていない状態であり、実行可能なプログラムや使用可能なメモリのサイズに制限があるためです。

U-Bootなどのブートローダは多機能なので、プログラムのサイズも大きく、メモリを多く使用します。そのため、SoCの起動直後にいきなり実行することができません。多機能で大きなプログラムを実行するには、段階を踏んでペリフェラルを初期化していく必要があります。

▶最初に実行されるプログラム「IPL」

IPLは、SoCのリセット解除後、最初に実行されるプログラムです。多くの場合、SoCのROMに書き込まれています。RK3399ではBoot ROM (BROM) と呼ばれるROMに書き込まれています。

▶ストレージやメモリを初期化する「SPL」

IPLの次に実行されるプログラムです。主にストレージやメモリの初期化を行います。U-Bootでも、SPLに相当するプログラムをビルドできます。

▶フル機能のブートローダを実行すると…

この状態では、既にストレージやメモリが完全に使用可能です。また、ネットワークやファイル・システムなどにもアクセス可能で、さまざまなデバイスからカーネルをロード/実行できるようになっています。

● RK3399のブートに必要な2つの要素

▶(1) U-Bootを起動するためのローダ

U-Bootは多機能なブートローダです。本体を実行するためには、段階的に幾つかの小さなローダを実行する必要があります。U-Bootのソースコードからは、それらのローダをビルドできるようになっています。