

# 公式SDKを使わずに LチカとUART通信

中森 章

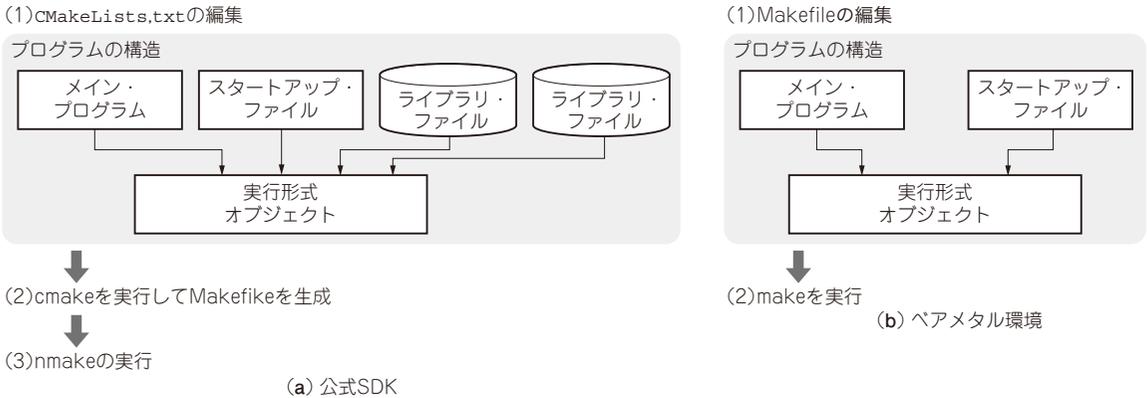


図1 公式SDKとベアメタル環境のプログラム作成の違い

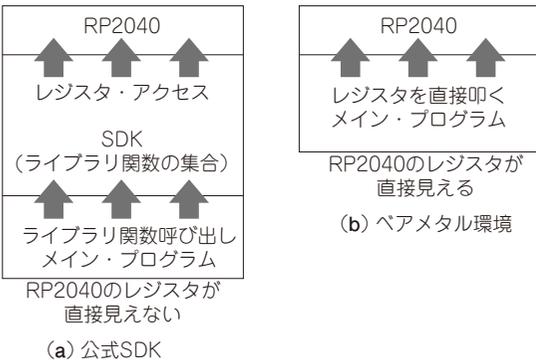


図2 公式SDKとベアメタル環境の実行形式プログラムの違い

## 公式SDKを使いたくない

筆者は、開発環境が単純なことが好みます。その点で、Windowsでベアメタル開発を行う環境にこだわっています。確かに、ラズベリー・パイ Pico (以降、Pico) を Windows で公式SDKを使って行う開発も OS を使わないという点ではベアメタルなのですが、公式SDKがOSのごとく作成したプログラムの便宜を図ってくれます。多分、多くの人は、これで満足すると思っています。

### ● 公式SDKはcmakeを使う前提だけど…

しかし、公式SDKを使った開発には、大きなハードルがあると思っています。それは、cmakeという構築ツールで一括管理されているため、cmakeになじみのない人(筆者も含む)には、自前のプログラムをどうやって作ってよいか分かりません。第7部第1章では、筆者の思いつく使い方を示しましたが、それはまっとうなものでしょうか。それに、何だかやぼったい手順です。

### ● make一発でプログラムがビルドされることを目指す

筆者が目指すのは、もっと単純な、WindowsのPowerShell内で、make一発でプログラムがビルドできる環境です。そこで、注目したのがDavid Welch氏のサンプル・プログラム<sup>(3)</sup>です。これらのプログラムは、少し変則的で、RAM上で実行するものですが、「単純でmakeを使う」という点で筆者の期待に合致しています。図1に公式SDKとベアメタル環境のプログラム作成の違いと図2に実行形式プログラムの違いを示します。

ともかく、David Welch氏のおかげで公式SDKを使わずにPicoの開発環境を構築するという勇気が湧きました。徒然草ではありませんが、「少しのことに、先達はあらまほしき事なり」です。