

Eclipseからpico-sdkやTinyUSBを使って ビルド&デバッグ! FreeRTOSプログラムの作成方法も

Pico用USBプログラミング 環境の構築

関本 健太郎

この章では、第2部 第2章以降の活用例として紹介するプログラムを作成するための環境の構築方法を説明していきます。開発のフレームワークとしてEclipseを、OSはWindows 10をメインで使用します。

ステップ1： Pico関連のライブラリの準備

Raspberry Pi財団が提供するラズベリー・パイPico（以降、Pico）関連のソフトウェアには、主なところでpico-sdk、pico-examples、pico-extras、pico-playgroundがあります。

● pico-sdk…アプリケーション作成

pico-sdkは、RP2040向けのアプリケーション作成のためのヘッダ・ファイル、ライブラリ、CMakeを利用したビルド・システムのフレームワークを提供しています。この中でTinyUSBライブラリがサブモジュールとして含まれています。

● pico-examples…各種周辺機能のサンプル

pico-examplesには、pico-sdkを利用したRP2040の各種周辺機能に関連するサンプル・プログラムが多数含まれています。

● pico-extras…追加ライブラリ

pico-extrasには、pico-sdkに含める前の段階の追加ライブラリ[オーディオ(I²S)サンプル、VGA/DPIなどのサンプル]が含まれています。

● pico-playground…pico-examples同様のサンプル

pico-playgroundは、pico-examplesと同様のサンプル・プログラム集ですが、追加ライブラリとしてpico-extrasを利用するプログラムを含んでいます。

*

これらはCMakeによるビルドが前提になっています。本記事では、pico-sdkのみ利用するので、適当なフォルダを作成し、pico-sdkリポジトリをクローンしておきます(リスト1)。

ステップ2： TinyUSBライブラリの設定

pico-sdkには、gitサブモジュールとして、libフォルダ下にTinyUSBライブラリがインストールされます。この設定は前節のPico関連のライブラリを中心に開発していく場合には便利ですが、基本的にTinyUSBライブラリのgitリポジトリの特定のコミットを前提としているため、pico-sdkで使われるTinyUSBのコミットよりも新しいコミットを利用したり、TinyUSBのバグを修正したりする場合には管理が複雑になります。そこで、独立してTinyUSBのgitリポジトリをクローンし、TinyUSBからpico-sdkのライブラリを参照するように設定します。

ここでは、pico_projectというgitリポジトリを作成し、そのサブモジュールとしてTinyUSBライブラリを設定します(リスト2)。

TinyUSBライブラリには多くのマイコンのライブラリがgitサブモジュールとして登録されています。これらをまとめてgitサブモジュールとして初期化すると、余分な時間とディスク領域を確保することになるので、後述のビルド・スクリプトでは必要なサブモジュールのみ初期化しています。

リスト1 pico-sdkリポジトリをクローンするコマンド

```
mkdir c:\pico
cd pico
git clone https://github.com/raspberrypi/pico-sdk.git
git clone https://github.com/raspberrypi/pico-examples.git
git clone https://github.com/raspberrypi/pico-extras.git
cd pico-sdk
git submodule update --init
// tinyusbサブモジュールを初期化する
```

リスト2 プロジェクトのgitリポジトリを作成するコマンド

```
cd c:\pico
git init pico_projects
cd pico_projects
git submodule add https://github.com/hathach/tinyusb.git lib/tinyusb
```