

ラズパイ Pico の USB 活用

第2回 USB CDCクラスを利用したFirmataライブラリ

関本 健太郎

2022年10月号特集は「USBホスト&デバイス ラズパイPico 虎の巻」でした。特集では、TinyUSBのサンプル・プログラムを詳しく解説したり、USBホスト、USBデバイスの製作事例を紹介したりしました。その中で、USB-I²Cブリッジについて、次のように解説しました。I²Cブリッジ製品には以下があります。

1. USB HIDクラスを利用したもの(2022年10月号特集第3部第1章)
2. USBベンダ・クラスを利用したもの(2022年11月号, pp.184-187)
3. USB CDCクラスを利用したもの(今回)

なお、上記1と2について、サポート・ページで公開しています。

<https://interface.cqpub.co.jp/2311usb/>

● 様々なプラットフォームで利用できる

シリアル通信を利用したホストとデバイスの通信プロトコルとして、Firmataがあります。USBデバイスの場合には、USB CDCクラスを利用したUSBブリッジ機能を介して利用できます。

Firmataライブラリは、シリアル通信(USB CDCクラス)を利用して、ホストPC上のソフトウェアと通信するためのFirmataプロトコルを実装しています(図1)。Windows, Linux, macOSで動作します。これにより、使っているプログラミング環境用に独自のプロトコルやオブジェクトを作成しなくても、カスタム・ファームウェアを作成できます。Arduino IDEに特化したFirmataライブラリが公開されており、広く利用されています。そこでRP2040向けにビルドして利用したいと思います。

USBディスクリプタ

Arduino IDEのボード・マネージャで「Arduino Mbed OS RP2040 Boards」→「Raspberry Pi Pico」を選択したとき、USBディスクリプタ(図2)は、典型的なCDCクラスのディスクリプタ構成になっています。

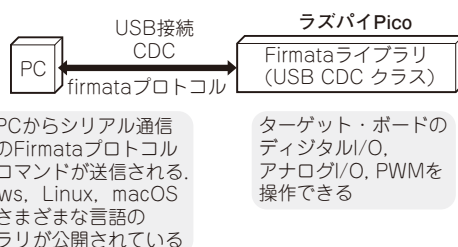


図1 Firmataライブラリの概要

Firmataプロトコル

Firmataは、コンピュータ(またはスマートフォン/タブレットなど)上のソフトウェアからマイコンと通信するためのプロトコルです。このプロトコルは任意のマイコンのファームウェアに実装できます。

Firmataはコマンド・バイトが8ビット、データ・バイトが7ビットというMIDIメッセージ・フォーマットに基づいています。Firmataのプロトコルを表1に示します。

Firmataメッセージ・フォーマットには、データ・メッセージ拡張(Data Messages Expansion, 0xE0, 0x90, 0xC0, 0xD0)とコントロール・メッセージ拡張(Control Messages Expansion)があります。

● データ・メッセージ拡張のフォーマットの指定方法

データ・メッセージ拡張のフォーマットは、第1バイトの上位4ビットがコマンド、下位4ビットがピン番号またはポート番号を指定します。データ・バイトが7ビットなので、8ビットのデータは、下位7ビットのデータをデータ部の第1バイトで、8ビット目のデータをデータ部の第2バイトの最下位ビットに割り当てます。例えば、

▶ (1) ホストからデバイスのポート2の値を 0x11001111 に設定するコマンドを送信する場合

Firmataのプロトコルでは、0x92(下位4ビットが