

USB-シリアル変換と USBメモリ…cdc_msc

関本 健太郎

動作

cdc_mscは、USBメモリ・デバイスとUSB-シリアル変換デバイスのコンボ・デバイスです(図1)。USBメモリ・デバイスは、USB MSC (マス・ストレージ・クラス)のディスクリプタを持ち、8KバイトのFAT12ファイル・システムとして構成されています。ただ、データはRAMに保存するので、ホストから切り離すと消えてしまいます。

USB-シリアル変換デバイスは、USB CDCクラスのディスクリプタを持ち、デフォルトの設定では、ホストからのシリアル入力データをホストへエコー・バックする設定になっています。

ディスクリプタ

USBディスクリプタ(図2)として、インターフェース・ディスクリプタを3つ持ちます。

ディスクリプタ0とディスクリプタ1は、USB CDCクラスのコントロールとデータ通信の機能を定義します。ディスクリプタ2はUSB MSCクラスの機能を定義しています。

その他に、インターフェース・アソシエーション・ディスクリプタが存在し、USB CDCクラスとして、2つのインターフェースを持つことを定義しています。ちなみに、2022年5月時点のRP2040用のMicroPythonもこの構成になっています。

プログラム

● main関数

サンプル・プログラムは、main関数の先頭で、TinyUSBのUSBデバイス初期化処理を行い、メイン・ループ処理を実施しています。メイン・ループ処理では、USBデバイスのタスク処理をtud_taskで実行し、USB CDCクラスに入力されたデータをcdc_taskで処理する流れになっています(リスト1)。

注1: USBメモリの内部ではSCSIコマンドがやり取りされている。

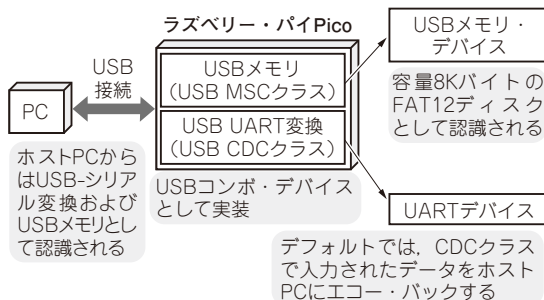


図1 cdc_mscのシステム構成

● cdc_task関数

cdc_task関数(リスト2)では、CDCクラスの入力をチェックし、入力データがあればデータを読み込み、tud_cdc_writeとtud_cdc_write_flashを呼び出し、データをホストへエコー・バックします。tud_cdc_writeは、USB CDCバッファへの書き込み処理だけを行い、この時点ではデータはUSBを介してホストには書き込まれません。tud_cdc_write_flashを呼び出すことで、ホストへの書き込みが実施されます。

CDCは、その他にライン・ステートが変更されたときに呼び出されるtud_cdc_line_state_cbとホストからCDCインターフェースがデータを受け取ったときに呼び出されるtud_cdc_rx_cbというコールバックが提供されています。

MSCクラスの処理は、msc_disk.cで実装されています。この中で、FAT12のブート・セクタのデータと、さまざまなSCSIコマンドを受け取ったときに呼び出されるコールバック関数群が定義されています^{注1}。このサンプルでは、ディスクのデータ領域は8Kバイトしかなく、データ領域はRAM領域に配置されています。msc_disk.cにRP2040のフラッシュ・メモリ領域への読み書き処理を追加すれば、フラッシュ・メモリとして実装できそうです。

◆引用文献◆

- (1) TinyUSBのサンプル。
<https://github.com/hathach/tinyusb/tree/master/examples/device>

せきもと・けんたろう