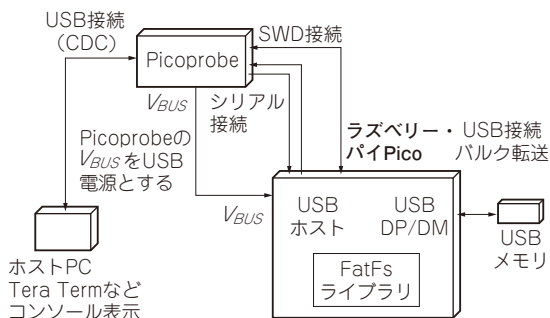


ドライバのバグを修正して、 FatFsによるファイル読み書きに挑む

MSCクラスを使ったUSBメモリへの アクセスにトライ

関本 健太郎



USBホストとして、接続されたUSBメモリをFatFsライブラリでマウントし、ファイルを書き込む

図1 USBホストとしてUSBメモリにデータを読み書きしたい

2022年5月時点のTinyUSBでは、USBホストはUSBクラスとしてはHID (Human Interface Device) クラスと、MSC (Mass Storage Class) しかサポートしていません。GitHubのIssue欄を参照すると、RP2040マイコンに至っては、残念ながら、MSCクラスはバグのため(仕様のため)動作しないようです。

ただし、プル・リクエストがされており、そのコードを利用すると、動作するようです。本章では、そのプル・リクエストでの修正点を踏まえ、MSCのサンプルを利用し、USBメモリへのアクセスにトライします(図1)。

TinyUSBのUSBホストの問題点

第1部第4章でUSBホスト/デバイスの初期化とイベント処理について説明しました。ここでUSBパケットの転送処理について簡単に説明し、現在どの部分に問題があるのかを確認します。

● USBホスト・モードでのUSBパケット転送

RP2040マイコンのUSBの処理は、ホスト/デバイスの共通処理を実装した`rp2040_usb.c`、デバイス処理を実装した`dcd_rp2040.c`、およびホスト処理を実装した`hcd_rp2040.c`の3つのファイルに分かれています(図2)。

▶ ホスト処理

USBホスト・モードの場合、USBデバイスが接続

されると、TinyUSBのホスト・ライブラリは、USBデバイスにディスクリプタをリクエストし、USBデバイスのクラス情報やそれにひも付くエンドポイント情報を取得するとともに、USBデバイスのデバイス・アドレスを設定します。

ホストのクラス・ライブラリは、USBデバイスに合わせて、デバイス・アドレスとエンドポイント・アドレスをキーにデータ転送を開始します。TinyUSBの場合には、デバイス・アドレスとエンドポイント・アドレスにひも付く情報を`hw_endpoint`構造体(第1部第4章参照)に格納します。

ホストのデータ転送は、デバイス・アドレス、エンドポイント・アドレス、データのポインタおよびサイズを指定して`hcd_edpt_xfer`関数を呼び出すことで開始します。

▶ ホスト/デバイス共通

`hcd_edpt_xfer`関数中で、デバイス・アドレス、エンドポイント・アドレスから、`hw_endpoint`構造体を確保します。`hw_endpoint`構造体とデータのポインタおよびサイズを指定して、`hw_endpoint_xfer_start`関数を呼び出します。

`hw_endpoint_xfer_start`関数では、`hw_endpoint`構造体にデータのポインタ、転送サイズを設定して、`_hw_endpoint_start_next_buffer`関数を呼び出します。

`_hw_endpoint_start_next_buffer`関数では、エンドポイント制御レジスタを設定し、1パケット分のデータ転送を開始します。1パケット分の転送が完了すると、割り込みがかかり、`hcd_rp2040_irq`割り込み関数が呼び出されます。割り込み関数の中でデータの転送情報が更新されます。

その後、`hw_endpoint_xfer_continue`が呼び出され、最初に指定したデータ長が転送されている場合には、`_hw_endpoint_xfer_sync`関数、`sync_ep_buffer`関数を呼び出した後、`hw_xfer_complete`関数、`hw_trans_complete`関数を呼び出し、転送を完了します。

一方、まだ転送データが残っている場合には`hw_endpoint_xfer_continue`関数から、`_hw_endpoint_xfer_sync`関数、`sync_ep_buffer`関数を呼び出した後、繰り返し`hw`

◆参考文献◆

(1) cjhilder/tinyusb (TinyUSBの修正情報)。

<https://github.com/cjhilder/tinyusb.git>