



FRマイコン基板を使った USBホスト・システムの 設計事例

前編



関連データ

根岸 智明

USBキーボードをつなげてみよう！

本誌2008年5月号付属FRマイコン基板の特徴として、USBホスト&ターゲット機能を内蔵している点が挙げられる。FRマイコン基板のUSBホスト応用事例として、ここではUSBキーボードの入力をASCIIコードなどに変換するサンプル・プログラムを作成する。USBホスト・スタックにはmini-USBHostSTRを使用している。(編集部)

● MB91FV310A のUSB ホスト機能を活用

本誌 2008 年 5 月号付属 FR マイコン基板に実装されている MB91FV310A には、フラッシュ ROM と RAM が内蔵されており、シングル・チップ・モードで動作させることができます。さらに、USB ホスト機能も内蔵しているため、さまざまな USB 周辺機器を接続することが可能です。

ただし、CPU 内蔵の資源だけでシステムを実現しようとした場合、RAM 容量が 16K バイトしかないために、高度なソフトウェアの実装は難しいケースがあります。特に、USB ホスト機能を活用したシステムの場合はより難しくなります。

そこで、FR マイコン基板を代表的な USB 周辺機器の機能を変換するアダプタ、またはモジュールとして使うことを考えました。今回は USB キーボードに対応し、キーボードで押されたキーを ASCII コードなどに変換するサンプル・プログラムを作成してみました。

なお、USB ホストを制御するミドルウェアとして mini-USBHostSTR (インターフェイス製)を使用しています。本

ミドルウェアの詳細は、先月号の特集 第 2 章 (2008 年 6 月号, pp.56-65) を参照してください。

● HID キーボード対応プログラムの動作仕様

HID キーボード・クラス対応のプログラムの動作の様子を、実際のソース・ファイルを使って説明します。

キーボード対応サンプル・プログラムのソース・ファイルをリスト 1 に示します。リストを見ると分かるように、main() 関数そのものは非常にシンプルな作りであり、実際に USB キーボードを動作 (ログ出力) させるために使用している API は表 1 に示す二つのみです。また、表 2 に USB キーボード・クラス・ドライバへの設定データ構造体を示します。

この二つの API には、それぞれコールバック関数を指定することができ、Usb_Host_Init() に指定したコールバック関数 (リスト 1 では Usb_Host_Callback() がこれに相当) は、デバイスの状態変化 (接続、切断など) に合わせて呼び出されます。また、Usb_Hid_Keyboard_Setting() に指定したコールバック関数 (リスト 1 では

表 1
USB ホスト・スタックの API

	型	内 容
引き数	なし	コールバック関数
戻り値	ISTATUS (long)	0 : 正常終了, 負数 : 異常終了

(a) USB Host スタック初期化関数 Usb_Host_Init()

	型	内 容
引き数	USBH_KEYBOARD_SETTING_T	設定データ (コールバック関数)
戻り値	ISTATUS (long)	0 : 正常終了, 負数 : 異常終了

(b) USB キーボード・クラス・ドライバへの設定 Usb_Hid_Keyboard_Setting()

表 2
USB キーボード・クラス・ドライバ
への設定データ構造体 (USBH_KEYBOARD_SETTING_T)

フィールド名	内 容
void(*eventf)(IHANDLE, int32_t, void *)	コールバック関数
void *prm	将来拡張用 (NULL 指定)

表3
USB キーボードの
イベント種別

シンボル名	イベント種別
HID_EVENT_KEY_DOWN	キー・ダウン
HID_EVENT_KEY_UP	キー・アップ
HID_EVENT_CTL_KEY_DOWN	制御キー (Shift, Ctrl, Alt, Windows) ダウン
HID_EVENT_CTL_KEY_UP	制御キー (Shift, Ctrl, Alt, Windows) アップ

Hid_Keyboard_Event() がこれに相当]は、USB キーボードの状態の変化に合わせて呼び出されます。

このように、状態の変化に合わせてUSB ホスト・スタックから呼び出されるコールバック関数を使用し、ログ出力 (CN5) を行うことでプログラムを実現しています。

Usb_Host_Callback() 関数を見ると、デバイスの状態変化に合わせて、ログ (“CN:USB Keyboard”, “CN:Unknown”, “DS:”) を出力しているのが分かります。

また、Usb_Hid_Keyboard_Event() 関数では、USB キーボードの状態変化に合わせて、文字コードを出力するようになっています。

なお、このコールバック関数の引き数 prm (第3パラメータ) 値は、キー・コードがそのまま与えられます。そのため、今回のように文字コードが必要な場合は、アプリケーション (USB ホスト・スタックの上位) 側で、キーボードの種別に応じて、文字コードに変換する必要があります。

さらに、コールバック関数の引き数 event (第2パラメータ) 値は、USB ホスト・スタックが独自に割り振ったイベントが与えられます。詳細は表3を参照してください。

● サンプル・プログラムの動作例

先月号で解説したUSB マウス対応サンプル・プログラムのひな型プロジェクトをベースに、リスト1のソース・ファイルを組み込んでプログラムをコンパイルし、生成された mhx ファイルをCPU に書き込みます。

次にFR マイコン基板のCN5 とパソコンをUSB ケーブルで接続し、認識された仮想COM ポートをターミナルで開いてください。各種パラメータは、通信速度 19200bps, データ長8ビット, ストップ・ビット1ビット, パリティなしとしてください。そしてFR マイコン基板のCN3 にUSB キーボードを接続します。

このプログラムを動作させたときのターミナルの画面表示の様子を図1に示します。

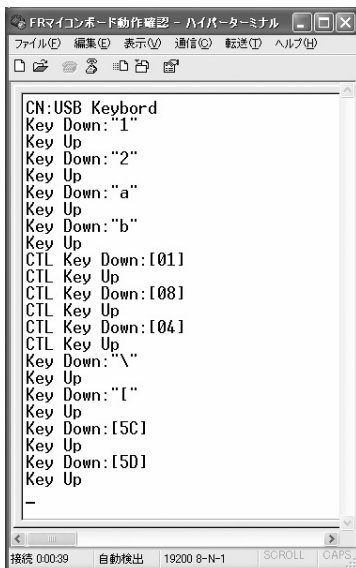


図1 USB キーボード接続時の動作例

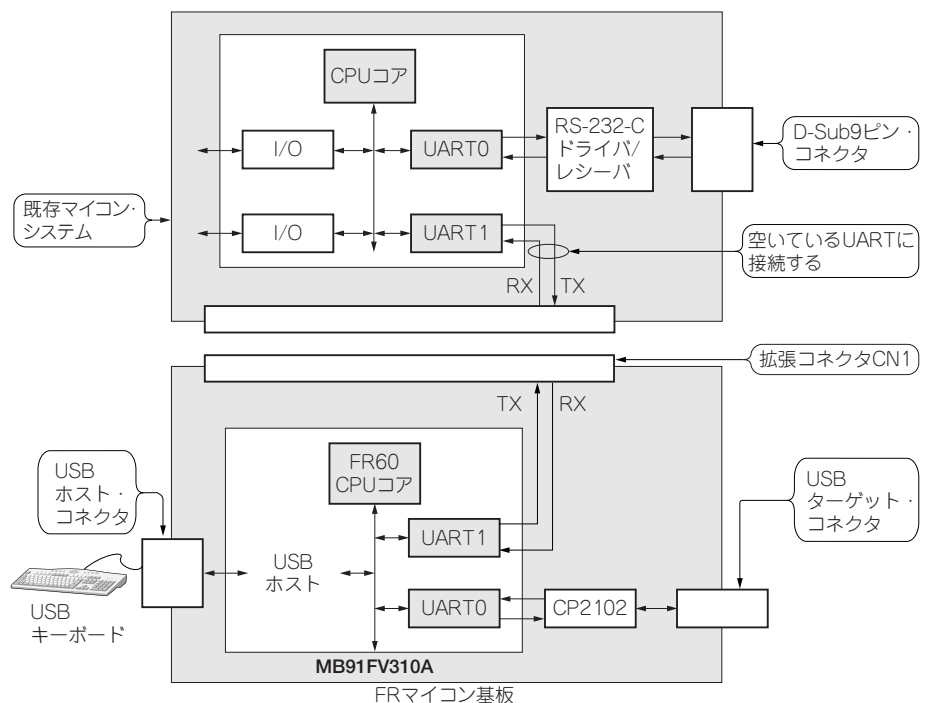


図2 FR マイコン基板をUSB キーボード変換アダプタとして使う